Delft University of Technology

Metaheuristic optimization for scheduling mixed-fleet electric buses in a practical urban network

Bosi, Tommaso; Rinaldi, Marco; D'Ariano, Andrea; Viti, Francesco

DOI
10.1016/j.cie.2025.111782

Licence
CC BY

Publication date
2025

Document Version
Final published version

Published in
Computers and Industrial Engineering

Citation (APA)
Bosi, T., Rinaldi, M., D'Ariano, A., & Viti, F. (2025). Metaheuristic optimization for scheduling mixed-fleet electric buses in a practical urban network. *Computers and Industrial Engineering*, *213*, Article 111782. https://doi.org/10.1016/j.cie.2025.111782

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Computers & Industrial Engineering 213 (2026) 111782

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Metaheuristic optimization for scheduling mixed-fleet electric buses in a practical urban network

Tommaso Bosi, Marco Rinaldi, Andrea D'Ariano *, Francesco Viti

Department of Civil, Computer Science and Aeronautical Technologies Engineering, Roma Tre University, Rome 00154, Italy
Department of Transport & Planning, Delft University of Technology, 2628 CD Delft, the Netherlands,
Department of Civil, Computer Science and Aeronautical Technologies Engineering, Roma Tre University, Rome 00154, Italy
Faculty of Science, Technology and Medicine (FSTM), University of Luxembourg, Esch-Sur-Alzette L-4364, Luxembourg

ARTICLE INFO

Keywords:
Vehicle Scheduling
Electric bus
Mixed-Fleet
Urban mobility
Fleet management
Simulated annealing
Genetic algorithm

ABSTRACT

This study addresses the scalability challenges of the Mixed-Fleet Multi-Terminal Electric Bus Scheduling Problem by exploring various heuristic and metaheuristic approaches applied to large urban networks. A novel Repeated Local Search (RLS) algorithm is developed to optimize full-day scheduling, incorporating key factors such as fleet assignment, charging constraints, and deadheading costs, while accounting for limited charging infrastructure. The RLS method generates initial greedy yet feasible schedules for a mixed fleet of electric and hybrid buses, serving as the foundation for two metaheuristic strategies: Simulated Annealing and a Genetic Algorithm. The Simulated Annealing approach is implemented in two variants: one integrating a Mixed-Integer Linear Programming (MILP)-based move, and the other using an RLS-based move to reschedule trip chains while maintaining feasibility. Meanwhile, the Genetic Algorithm employs repair mechanisms to correct infeasible solutions arising during the crossover process. To evaluate these methodologies, a three-phase experimental framework is employed: (1) stress-testing a MILP model under various fleet and infrastructure conditions, (2) benchmarking MILP performance against metaheuristic methods on small-scale instances, and (3) conducting a comparative analysis of metaheuristics across small, medium, and real-size urban scenarios. The urban-scale instances are derived from real-world public transit timetables in Luxembourg City, encompassing 1,084 trips, 12 terminals, 10 bus lines, and full-day operations. Results indicate that the proposed metaheuristic approaches achieve solutions comparable to exact MILP formulations in small-scale cases while offering substantial scalability improvements for larger networks. Each algorithm exhibits distinct advantages and trade-offs, highlighting the importance of selecting an appropriate method based on the specific scenario and computational constraints. These findings extend prior research on smaller instances and suggest that as urban transit systems transition to electric fleets, the marginal operational benefits for transit agencies may diminish with increasing network size.

1. Introduction

Transportation plays a fundamental role in carbon emissions, with its impact recently estimated at 1/4th of all energy-related CO_2 emissions worldwide (European Environment Agency, 2022). As a result, the swift focus of policy, planning, and research has converged on the electrification of this sector, with a shared goal of propelling a transition toward sustainability. Unlike private transport, Public Transport (PT) stands as a promising catalyst for the rapid adoption of Electric Vehicles (EVs), driving sustainability and decarbonization within the urban mobility landscape.

Nevertheless, in many countries, particularly in Europe, the transition to fully electrified bus fleets is expected to be a gradual process, bringing forth challenges in effectively managing mixed- or full-electric fleet scheduling (Pelletier et al., 2019). These challenges stem primarily from factors encompassing battery technologies and materials, as well as the policy requirements and investments associated with charging infrastructure (Fig. 1). Over the past decade, battery prices have consistently declined, accompanied by improvements in specific energy and lifespan (International Energy Agency, 2023). This trend might

* Corresponding author.
E-mail addresses: tommaso.bosi@uniroma3.it (T. Bosi), m.rinaldi@tudelft.nl (M. Rinaldi), andrea.dariano@uniroma3.it (A. D'Ariano), francesco.viti@uni.lu (F. Viti).

https://doi.org/10.1016/j.cie.2025.111782
Received 4 March 2025; Received in revised form 16 December 2025; Accepted 22 December 2025
Available online 26 December 2025
0360-8352/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

encourage a strategy of delaying significant electric bus purchases to capitalize on lower acquisition costs, while also avoiding both sunk costs and missing out on further technological advancements. Furthermore, achieving full electrification will demand substantial economic investments in establishing multiple charging terminals to meet the growing charging needs during daily operations. Numerous technologies and charging strategies warrant consideration, including options like pantograph and plug-in charging, opportunity, or depot charging (Jefferies and Göhlich, 2020). However, it is crucial to highlight that not all terminal or depot locations align with building requirements or feasibility standards for these charging solutions, mainly due to constraints like space availability and logistical compatibility (Hall and Lutsey, 2020; Amiri et al., 2023; Feng, 2024). In cases where there is an excessive demand for recharging relative to the low number of available charging stations, the potential for queues at terminals or depots arises, which could subsequently lead to delays in adhering to the timetable. Another aspect of concern is the possibility of grid instability and blackouts. While Bus-to-Grid (B2G) has the potential to enhance grid flexibility and provide demand response capabilities, the sudden withdrawal of a large number of connected vehicles during peak demand could strain the grid and lead to voltage fluctuations or even localized blackouts (Fei et al., 2023; Noel and Mccormack, 2014). Opting for a mixed-fleet strategy provides a buffer against the risk of overloading the grid during high-demand periods. From an operational standpoint, the incorporation of EVs into the bus operator's fleet introduces additional complexities. The limited range of EVs represents a constraint on multiple aspects of PT planning (Ceder, 2016): line planning might require alteration in order to ensure that (some) line terminals are strategically located near charging stations able to partly recharge EVs in a matter of minutes or even a few seconds; timetabling might be adapted to allow for additional slack times between consecutive trips, so to allow quick charging at terminals, causing a trade-off with respect to the current level of service; vehicle scheduling is especially impacted by range constraints: while fleets composed by internal combustion vehicles can be scheduled so as to maximise the distance travelled and minimise downtime, and to best utilise the available fleet, the limitations related to battery discharging and recharging require careful consideration both in terms of how feasibly dense a given EV's schedule can be, as well as how access to limited capacity charging stations is handled.

For operators, electrification might therefore lead to increased overheads as redundancies in the fleet become necessary in order to meet all constraints. Hence, a well-considered strategy for selecting the most suitable vehicle types and assigning them to specific trips holds the key to upholding the schedule's integrity and ensuring the desired service level. In literature, these problems are categorized as Multiple Vehicle Type Scheduling Problems (MVTSP), a category known for its NP-hard complexity, as established by (Costa et al., 1995). While seeking exact solutions to said problems becomes quickly intractable with problem size and complexity, the potential advantages for operators are considerable: effective management of mixed-fleet scenarios can simultaneously leverage the advantages of reduced operational costs for electric buses and maintain adherence to energy-related constraints by minimizing the number of trips performed by internal combustion or hybrid buses within the schedule. Nevertheless, as the scale of the studied problem increases from a few selected lines to urban-sized instances, the intricacies of multi-terminal interactions quickly become impractical from a computational perspective. Vehicle scheduling must consider interactions between bus routes (such as utilizing common terminals for route switches), including deadheading trips to efficiently distribute resources across lines, and access to a limited number of charging stations, all while striving to avoid station queues. Consequently, the development of effective solution heuristics and metaheuristics becomes imperative for addressing real-world instances while still delivering solutions of acceptable quality.

Following previous studies (Picarelli et al., 2020; Rinaldi et al., 2020), in this work we focus on designing a solution framework for the Mixed-Fleet Multi-Terminal Electric Bus Scheduling Problem (MFMTEBSP), with a specific emphasis on ensuring scalability that enables the application to urban-scale scenarios. The scenario under consideration involves: a mixed fleet of electric and hybrid buses, a subset of terminals equipped with a limited number of charging stations, and the implementation of opportunity charging at terminals (OC-T) via pantographs. The objective is to minimize both operational costs, encompassing fleet assignment, charging, and deadheading costs, as well as fixed costs associated with the fleet size. As a starting point, we propose a modified version of the MILP model from (Picarelli et al., 2020), aiming to tackle full micro instances and establish an optimal benchmark. Then we introduce a Repeated Local Search (RLS) heuristic, named Chain-Trip Builder (CTB). This heuristic is implemented both as a standalone method and as a feasible single-solution and population

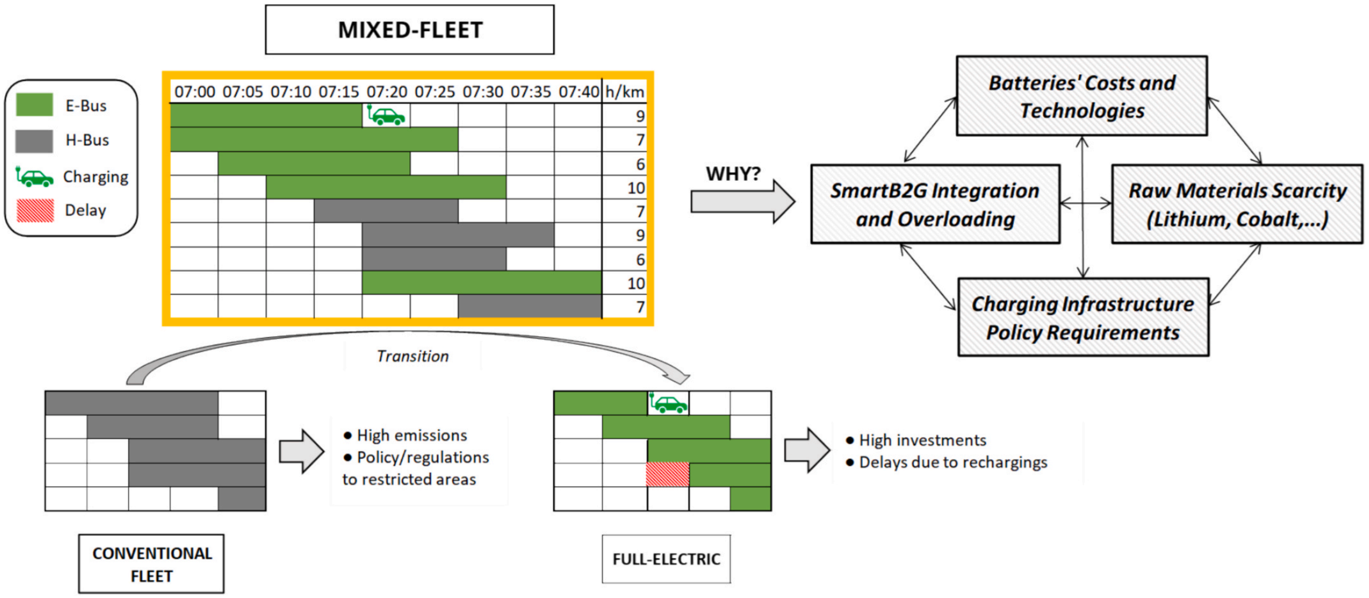

MIXED-FLEET
E-Bus
H-Bus
Charging
Delay
07:00 07:05 07:10 07:15 07:20 07:25 07:30 07:35 07:40 h/km
9
7
6
10
7
9
6
10
7
WHY?
Batteries' Costs and Technologies
SmartB2G Integration and Overloading
Raw Materials Scarcity (Lithium, Cobalt,...)
Charging Infrastructure Policy Requirements
Transition
High emissions
Policy/regulations to restricted areas
High investments
Delays due to rechargings
CONVENTIONAL FLEET
FULL-ELECTRIC

Fig. 1. Comparison of motivations for mixed-fleet adoption versus conventional or fully-electrified fleets, considering their strengths and the limitations of the ecological transition era.

generator for subsequent metaheuristics. The CTB exploits a stochastic strategy to assign electric or hybrid buses to feasible chains of trips, which are generated using a window-based approach. We conduct an in-depth analysis and hyperparameter tuning stage (using grid search) for CTB to ensure proper configuration of exogenous parameters. We further introduce two nature-based metaheuristics: a Simulated Annealing (SA) and a Genetic Algorithm (GA). While the GA leverages crossover operations to generate new offspring solutions, the SA employs two distinct moves: the first is a double-chain switch move, utilizing a modified version of the MILP model presented; the second move, a *n-chain* switch move, selects *n* chains of trips for rescheduling and employs a modified version of the CTB to find feasible solutions.

To validate our proposed approach, we engage in three distinct testing phases. Initially, we subject the MILP model to a stress test, identifying crucial instance attributes for utilization in the subsequent phase. In the second stage, we assess the quality gap among the approximate approaches by comparing them with the MILP model, employing artificial micro instances. For the third stage, we evaluate the performance of the proposed metaheuristic across three case studies, encompassing both real-world and synthetic scenarios of increasing sizes. During this phase, we handle urban-sized instances, involving a full day's schedule that spans 12 terminals, 10 bus lines, and a total of 1084 trips. Additionally, we explore multiple fleet size compositions and infrastructure configurations through various tests. This allows us to assess the metaheuristics' effectiveness, behaviors, and their ability to tackle problems that are infeasible for MILP-based approaches due to their prohibitive scale. While our methodology is developed with scalability in mind, its application and evaluation in this study focus on a case study representing a mid-sized European city. Given the specific characteristics of Luxembourg City, including its modest size and relatively short bus routes, future research may explore the applicability of the proposed methods in denser and more complex urban environments.

This paper is organised as follows. Section 2 presents a literature review centred on the Electric-Vehicle Scheduling Problem (E-VSP) within public transportation. Special emphasis is placed on recent advancements in both fully electrified and mixed-fleet strategies, along with the utilization of heuristics and metaheuristics in addressing urban network scenarios. In addition, we discuss the contributions of our study to the Mixed-Fleet Bus Scheduling Problem. In Section 3 we first present a MILP formulation of the Mixed-Fleet Multi-Terminal Electric Bus Scheduling Problem and our assumptions. We then detail the design and development of the CTB, the customized SA, and its corresponding moves, as well as the GA, integrated with its repair rules. Afterwards, in Section 4 we detail the experimental results obtained on three distinct testing phases, accompanied by a comprehensive outline of the experimental setup and the KPIs definition. Finally, concluding remarks are drawn in Section 5 along with directions for further extending the proposed methodology.

2. Literature review

In this section, we delve into the primary studies relevant to the Multi-Vehicle Scheduling (MVSP) and Electrified Vehicle Scheduling (E-VSP) Problems, which have been categorized in Table 1 and Table 2. The key aspects guiding the categorization of these studies include:

- Objective Functions: We examine the specific objectives pursued in each study;

Table 1
Objective functions, Constraints, and Specifications related to state-of-the-art.

	Objective Function	**Constraints & Specs**				
Study	**Minimizing**	**Fleet**	**Charging Location**	**Vehicle Technology**	**Number of Chargers**	**Charging Technology**
Kliewer et al. (2006) (Kliewer et al., 2006)	ST, DH, FS	Multi	–	T	–	–
Pepin et al. (2009) (Pepin et al., 2009)	ST, TR, FS	Full	–	T	–	–
Marin Moreno et al. (2019) (Marín Moreno et al., 2019)	DH, FS	Full	–	T	–	–
Chao & Xiaohong (2013) (Chao and Xiaohong, 2013)	CR, FS	Full	–	E	–	BS
Li (2014) (Reuer et al., 2015)	ST, TR, FS	Full	DC	E, NC	–	F, BS
Reuer et al. (2015) (Reuer et al., 2015)	TR, CR, DH, FS	Full	OC-T	E	SST	F
Van Kooten et al. (2017) (Van Kooten Niekerk et al., 2017)	DH, BD	Full	OC-T	E	SST	F
Rogge et al. (2018) (Rogge et al., 2018)	DH, CR, CI, FS	Multi	DC	E	–	DPI
Liu et al. (2019) (Liu and Ceder, 2020)	FS, CI	Full	OC-T	E	AT	F
Olsen et al. (2020) (Olsen et al., 2022)	TR, DH, FS	Mixed	OC-T	E,T	SST	F
Jefferies et al. (2020) (Jefferies and Göhlich, 2020)	FS, CI, CR, CW	Full	DC, OC-T	E	AT	F
Rinaldi et al. (2020) (Rinaldi et al., 2020)	TR, CR, DH	Mixed	OC-T	E, NC	ST	F
Picarelli et al. (2020) (Picarelli et al., 2020)	TR, CR, DH	Mixed	OC-T	E, NC	SST	F
Zhou et al. (2021) (Zhou et al., 2020)	TR, CR, DH, EM	Mixed	OC-T	E, T	–	F
Wang et al. (2021) (Wang et al., 2021)	TR, DH, CR, ST, FS	Full	OC-T	E	–	F
Perumal et al. (2021) (Perumal et al., Aug. 2021)	TR, DH, CW	Full	DC	E	–	DPI
Sistig et al. (2023) (Sistig and Sauer, Jun. 2023)	TR, DH, CW	Full	DC,OC-T	E	SST	F, DPI
Parmentier et al. (2023) (Parmentier et al., May 2023)	TR, DH, FS	Full	DC, OC-T	E	AT	F, DPI
Zhou et al. (2024) (Zhou et al., Apr. 2024)	TR, FS, CR, BD	Full	DC	E	SST	DPI
Our Study	TR, CR, DH, FS	Mixed	OC-T	E, NC	SST	F

ST = Stand Time
DH = Deadheading
FS = Fleet Size
TR = Travel Costs
CR = Charging Costs
CW = Crew Scheduling
CI = Charging Infrastructures
BD = Battery Deprecation
EM = Emissions

Fleet: Full, Multi – Multi-vehicles, Mixed – Mixed-fleet
Charging Location: DC – Depot Charging, OC-T – Opportunity Charging at Terminal
Vehicle Technology: E – Electric, T – Traditional, NC – Non-conventional
Number of Chargers: AT – All Terminals, SST – Sub-Set of Terminals, ST – Single Terminal
Charging Technology: F – Fast Charging, DPI – Depot Plug-in BS – Battery Swapping

Table 2
Solution Methods and Datasets related to state-of-the-art.

	Solution Method					Dataset				
Study	**MIP**	**CG**	**S**	**MH**	**H**	**|T|**	**|D|**	**|L|**	**|K|**	**Test**
Kliewer et al. (2006) (Kliewer et al., 2006)	●					7068			124	Germany
Pepin et al. (2009) (Pepin et al., 2009)		●		TS, LNS	●	1500	8			Random
Marin Moreno et al. (2019) (Marín Moreno et al., 2019)	●			GA		719	5			Colombia
Chao & Xiaohong (2013) (Chao and Xiaohong, 2013)				GA		119		3		China
Li (2013) (Li, 2014)	●	●				947				USA
Reuer et al. (2015) (Reuer et al., 2015)					●	10,702				Real-world
Van Kooten et al. (2017) (Van Kooten Niekerk et al., 2017)	●	●				543		3		Belgium
Rogge et al. (2018) (Rogge et al., 2018)				GA		200		3		Germany
Liu et al. (2019) (Liu and Ceder, 2020)	●				●	544			3	Singapore
Olsen et al. (2020) (Olsen et al., 2022)					●	10,710				Germany
Jefferies et al. (2020) (Jefferies and Göhlich, 2020)			●			4748	1	39	93	Real-world
Rinaldi et al. (2020) (Rinaldi et al., 2020)	●				●	1008		21		Luxembourg
Picarelli et al. (2020) (Picarelli et al., 2020)	●				●	536			5	Luxembourg
Zhou et al. (2021) (Zhou et al., 2020)				SA		575			4	China
Wang et al. (2021) (Wang et al., 2021)		●		GA-CG		145	3	3		China
Perumal et al. (2021) (Perumal et al., 2021)					●	1109		13		Denmark
Sistig et al. (2023) (Sistig and Sauer, 2023)					●	282		3	4	Germany
Parmentier et al. (2023) (Parmentier et al., 2023)		●			●	800	3	6	7	Real-world
Zhou et al. (2024) (Zhou et al., 2024)	●	●		ALNS	●	100	4			Singapore
Our Study	●			SA, GA	●	1084		10	12	Luxembourg

MIP = Mixed-Integer Programming
CG = Column Generation
S = Simulator
MH = Metaheuristic
H = Heuristic

|T| = Trips
|K| = Terminals
|D| = Depots
|L| = Lines

- Problem specifications: This encompasses crucial elements such as fleet composition and technology, charging infrastructure specifics including location and technology, and the number of charging stations;
- Methodology proposed: We detail the approaches utilized, including Mixed-Integer Programming, Column Generation, and Simulator, as well as heuristic and metaheuristic methodologies;
- Dataset considerations: We account for factors such as the number of trips, depots, bus lines, and terminals in the datasets employed.

For comprehensive references, we conducted meticulous analyses of up-to-date surveys such as those by (Pelletier et al., 2019) and (Perumal et al., 2022). In public transportation literature, vehicle scheduling problems have been extensively researched over the last fifty years. For a comprehensive overview, readers can turn to (Ceder, 2016) and (Ibarra-Rojas et al., 2015). In its most generic form, involving multi-terminals and/or multiple vehicle types, this problem showcases NP-hard complexity (Costa et al., 1995; Shang et al., 2023), hence motivating considerable research efforts in developing efficient exact and approximate solution algorithms. As shown by (Gintner et al., 2005; Kliewer et al., 2006) and (Pepin et al., 2009), the scalability of methods is a critical issue as well as the development of tools that run on PCs with common specifications, particularly when the field reaches a certain level of maturity. Notably, (Pepin et al., 2009) can be viewed as a counterpart, for traditional fleets, to our study. In this case, the focus was on tackling dense urban network instances of increasing magnitudes, incorporating a comparison of metaheuristics such as tabu search and large neighborhood search.

Nevertheless, it is important to note that neither a quality gap assessment nor the inclusion of real-world instances were part of the study's framework. The impact of PT electrification has gained traction in research in the last decade, with studies evaluating the potential benefits and risks arising from fleet replacement both from economic and operational perspectives (Fusco et al., 2013; Xylia, 2019; Lajunen, 2014). This trend has most recently led to an increased number of contributions and developments in the E-VSP, both concerning full fleet electrification scenarios and mixed-fleet integration. Considering the numerous specifications that could potentially be taken into account for the electrified version of this problem, our review will focus on analyzing studies that distinctly outline research gaps. This analysis will encompass factors such as fleet composition, charging infrastructure, comparisons involving MIP comparison, as well as heuristic and metaheuristic evaluations specifically targeted at urban-scale instances.

Mixed-Fleet. This attribute holds significance from both practical and theoretical perspectives, since most companies do not operate pure EV fleets but are gradually introducing EVs into their existing hybrid fleet (Goeke and Schneider, 2015). Notably, economic savings have been meticulously examined and quantified in prior studies (Picarelli et al., 2020; Rogge et al., 2018). The limitations of this full-electrified fleet scenario are underscored by analogous challenges found in related problems such as Fleet Replacement (Feng and Figliozzi, 2013; Bektaş et al., 2019) and Vehicle-to-Grid (Steward, 2017; Manzolli et al., 2022; Thingvad et al., 2021). As far as our knowledge extends, only four earlier studies concentrate specifically on mixed-fleets of electric and traditional buses: (Rinaldi et al., 2020) tackle the single-terminal version of our problem by introducing a MILP approach for real-size instances; (Picarelli et al., 2020) enhance this MILP framework, considering multiple terminals and employing a Time-Decomposed approach for small-scale instances; A further heuristic approach has been proposed by (Olsen et al., 2022) with feasibility and scalability as their primary objectives. This study employs real-world timetables encompassing thousands of trips; (Zhou et al., 2020) offer a solution closest to ours. They suggest a single Simulated Annealing approach for managing medium-sized instances with up to 575 trips and 4 terminals. Notably, all terminals in their scenario are equipped with charging stations. In all the aforementioned studies, no comparisons involving MIP or metaheuristic approaches have been presented for evaluating quality gaps. However, the assessment of quality gaps, even when applied to micro instances, remains a pivotal aspect for evaluating the performance of proposed approximate approaches.

Charging infrastructure. Within the vehicle scheduling problem, the focus typically centers around two fundamental timetable elements: routes and trips. Often, for routes, Depot Charging (DC) is considered at the end of the route (Rogge et al., 2018; Li, 2014), and Opportunity Charging at terminal stops (OC-T) is linked to trips covering and leverages fast charging technologies. OC-T is widely used by transport companies and has proven to be preferable in terms of Total Costs of Ownership (Jefferies and Göhlich, 2020). Nevertheless, when considering OC-T, it is noteworthy that not all prior studies account for a subset of terminals equipped with charging stations (Liu and Ceder, May 2020; Zhou et al., 2020). This constraint reflects real-world limitations related to grid access, capital investment, and deadheading implications. (Zhang et al., 2024) further expand this modeling dimension by incorporating station capacity and peak power constraints into a metaheuristic framework. Their results show operational cost savings between 7 % and 37 % when such constraints are enforced, underscoring the need to integrate infrastructure-aware scheduling strategies. Moreover, our experiments encompass various charging infrastructure configurations to comprehensively examine the behavior of the proposed metaheuristics in relation to varying percentages of terminals equipped with chargers (similar experiments are shown in (Olsen et al., 2022).

MIP Comparison and Quality Gap Assessment. Quality gap assessment is an essential stage to evaluate the performance of an approximate algorithm. However, the intricacies of electrification introduce variables and constraints to the VSP that escalate the computational effort required for exact approaches. This exponential escalation often renders MIP models impractical even for small-scale instances. Only a handful of prior studies have ventured into MIP comparisons while proposing approximate methodologies: (Li, 2014) propose a vehicle-scheduling model for electric transit buses employing battery swapping or fast charging. This model is subsequently compared with column-generation-based algorithms; (Van Kooten Niekerk et al., 2017) introduce dual models for E-VSP. The first model integrates a linear charging process, while the second considers various charging processes and integrates battery depreciation costs. Similar to the previous case, these model outcomes are contrasted with a column generation approach. Our study extends to the incorporation of a non-novel MILP model tailored to address full-scale instances. This model not only undergoes a rigorous scalability stress test but also engages in a comprehensive comparison with the heuristic and metaheuristic proposals applied to micro instances.

Exploring Multiple Heuristic/Metaheuristic Approaches and Comparisons. The evolution from exact approaches used in formalizing problems and controlled scenarios to approximate solution algorithms designed for scalability in real-world scenarios is a natural progression. (Chao and Xiaohong, 2013) directed their attention to the Full-electrified Vehicle Scheduling Problem coupled with battery swapping. Their aim was to minimize the demand for charging and the size of the bus fleet. To tackle this, they introduced a Genetic Algorithm, tailoring it to small-scale instances comprising 119 trips from three routes within the Chinese bus network. The same approach is integrated by (Wang et al., 2021) with a Column Generation algorithm (GA-CG). To comprehensively evaluate the efficacy, they pitted this against a Branch-and-Price approach as well as an unoriginal Adaptive Large Neighbourhood Search Algorithm. The methodology is then used to solve small instances of up to 145 trips and three lines from the China bus network. (Parmentier et al., May 2023) introduced a bidirectional pricing strategy combined with diving heuristics, solving large instances of up to 800 trips based on data from Montréal. Their results underline the competitiveness of CG-based decompositions for large-scale applications. (Zhou et al., 2024) addressed the Electric Bus Charging Scheduling Problem in a complex bus network, featuring multiple depots, routes, and heterogeneous buses, as a set-covering model. For small instances, they apply a branch-and-price algorithm to obtain optimal solutions; for larger, real-world scenarios, they develop an optimization-based adaptive large neighborhood search (opt-ALNS). Integrating metaheuristic techniques with MIP approaches presents an intriguing synergy. (Jovanovic et al., 2023) applied a matheuristic variant of Fixed Set Search (FSS) to the E-VSP, combining a reduced search space strategy with MILP-based intensification. (Vendé et al., 2023) address a multi-day EB assignment with overnight charging under depot charger constraints and battery degradation. After proposing an MILP formulation, they design two multi-phase matheuristics: one utilizing the full MILP in rolling horizon, and another industrial-style day-by-day approach. On 264 real-world instances, these matheuristics significantly outperform sequential strategies in both cost and solution quality. The complexities arising from electrification-related constraints often hinder the seamless application of pure metaheuristic moves without leading to infeasibility outcomes (Marín Moreno et al., 2019; Rogge et al., 2018). With the exception of (Zhou et al., 2020) and our study, prior research predominantly centered around Genetic Algorithms due to the integration of crossover operations, effectively circumventing the aforementioned limitations. To the best of our knowledge, our study represents a pioneering effort introducing a diverse set of heuristic and metaheuristic strategies for the MFEBSP. Specifically, we introduce a Repeated Local Search heuristic, a Simulated Annealing algorithm featuring two distinct moves – one of which integrates a MIP model – and a Genetic Algorithm bolstered by repair rules. This multi-pronged approach rises to tackle larger-scale scenarios, beyond preceding contributions (Yao et al., 2020).

Large-Scale Scenarios, Real-World and Synthetic. The complexity of E-VSP instances is primarily derived from the scale of trips planned into the timetable. However, this attribute should not be the sole consideration. For instance, when we account for trip coverage, we demonstrate in Section I.2 of the Supplementary Materials that the number of terminals emerges as another critical factor. Indeed, beyond trip count, the number of depots, bus lines, and terminals exerts substantial influence, as evident from Table 15. (Reuer et al., 2015) address Full-Electrified VSP including a subset of terminals equipped with charging stations. Their approach introduces two local search algorithms, enabling them to navigate thousands of trips. Notably, there is a lack of insight into quality gaps or the tally of terminals. (Olsen et al., 2022) confront instances of equivalent magnitude, employing decomposition strategies to attain feasible vehicle rotations. Their work extends to encompass experimentation with diverse charging infrastructure compositions. In the landscape of these studies, heuristic methodologies reign supreme, adeptly handling practical scenarios at a regional level. An intriguing endeavor by (Jefferies and Göhlich, 2020) aims to minimize fleet size, charging station count, workforce, and energy requisites through the implementation of a discrete-event simulation tool. Their scope spans scenarios featuring up to 4748 trips, 93 terminals, 39 bus lines, and a singular depot. However, delving into the realm of metaheuristic approaches sees a reduction in instance size, as the computational demand escalates to surmount local minima (Rogge et al., 2018; Liu and Ceder, 2020; Zhou et al., 2020). Despite the intricacies in comparing complexities due to varied settings, our case study seizes upon urban-sized instances, encompassing up to 1084 trips, 12 terminals, and 10 bus lines. In doing so, we surpass previous benchmarks (Yao et al., 2020).

Holistic Framework. Special considerations come into play when investigating studies that interweave E-VSP with additional challenges, such as crew scheduling and optimizing charging station locations. (Rogge et al., 2018) present a Genetic Algorithm (GGA) designed for the TCO-optimized planning of depot charging for battery-powered bus fleets and the associated charging infrastructure. In another innovative approach, (Perumal et al., 2021) converge two challenges that are often addressed sequentially: E-VSP and Crew Scheduling, forming what is known as E-VCSP. To tackle this amalgamated problem, they propose an Adaptive Large Neighborhood Search that leverages branch-and-price heuristics. Echoing this approach, (Sistig and Sauer, 2023) also delve into the same problem and methodology, elucidating the pronounced influence of crew scheduling constraints on the total cost of ownership and the requisite vehicle count across varying electrification paradigms. Complementing these integrated approaches, (Kost et al., 2025) address the temporal variability and operational uncertainty inherent in electric vehicle scheduling. Kost et al. propose a MILP framework with stochastic travel times evaluated via Monte Carlo sampling, highlighting how variability in deadheading and charge reserves affects feasibility and robustness. While holistic models that deal with the integration of strategic and tactical bus planning phases can be useful in specific conditions (i.e. crew costs higher than capital costs and national labour laws), they require constraints to be relaxed and simplified as well as instances size reduction due to computational limits. This often involves narrowing the focus to a single or few routes, as demonstrated by (Sistig and Sauer, 2023).

3. Research contributions

Building upon the identified research gaps, our research makes significant contributions in the following areas:

- We address the scalability issues in the Mixed-fleet Multi-Terminal Electric Bus Scheduling Problem, encompassing both electric and hybrid buses. Previous mathematical programming approaches (e.g., (Picarelli et al., 2020; Rinaldi et al., 2020)) provided sub-optimal solutions for reduced-scale instances with partial daily timetables. In this context, we consider a subset of terminals equipped with limited charging stations, incorporating the concept of opportunity charging at terminals. The formulation of our objective function includes fleet assignment, charging, and deadheading costs. Particularly, the scarcity of charging stations directly influences deadheading costs;
- We introduce the Chain-Trip Builder (CTB), a novel Repeated Local Search algorithm. This heuristic algorithm generates feasible solutions using stochastic and window-based strategies. CTB serves as the foundation for generating initial solutions for subsequent metaheuristic approaches. We perform comprehensive hyperparameter tuning to identify optimal settings for parameters such as window size, iteration count, and probability variations. The definition of "window size" is provided in Section 3.2;
- We propose two tailored nature-inspired metaheuristics: Simulated Annealing (SA) and Genetic Algorithm (GA). While most studies, including (Reuer et al., 2015; Olsen et al., 2022) and (Wang et al., 2021), primarily address challenges in large scenarios using heuristics or metaheuristics tailored for small instances, our research introduces metaheuristics that successfully yield high-quality solutions for urban-sized scenarios. SA leverages the best solutions derived from CTB and employs two distinct moves. The SA-2CM variant incorporates the revised MILP model to reschedule two chains of trips. A pre-processing phase allows to reduce the instance size by considering only the time-slots and terminals involved by the two chains. On the other hand, the SA-nCH utilizes an adapted CTB to reschedule a number n of chains selected randomly from a trip-proportional range;
- To facilitate testing and evaluation, we develop two synthetic-data generators for micro and real-world instances. The first generator introduces variability by modulating power and travel time parameters for each trip, resulting in a range of timetable shapes. The second generator mirrors the daily behavior of real-world timetables;
- Our research unfolds through three distinct experimental phases. Initially, we subject a modified version of the MILP model sourced from (Picarelli et al., 2020) to a stress test, identifying critical attributes from a computational efficiency standpoint. This examination is crucial for uncovering the primary contributors to complexity, as the prevailing literature often emphasizes the number of scheduled trips as the pivotal factor. Subsequently, we conduct a rigorous comparison between the MILP and our proposed approximate algorithms, focusing on assessing the quality gap between solutions. Indeed, prior similar works proposing heuristic and metaheuristic algorithms, such as (Olsen et al., 2022; Zhou et al., 2020), lack assessments against an optimal benchmark. Finally, we evaluate the performance of our metaheuristics using real-world timetables from the City of Luxembourg's bus network (up to 1084 trips, 12 terminals, and 10 bus lines). We explore scenarios involving different proportions of electric buses and varied charging infrastructure compositions, enabling a thorough understanding of algorithm behavior and efficiency.

4. Methodological approach

In this section, we first present the formulation of the Mixed-Fleet Multi-Terminal Electric Bus Scheduling Problem, which builds on the model previously introduced in (Picarelli et al., 2020). Moving forward to Section 3.2, we delve into the introduction of the CTB algorithm, describing its flowchart and the hyperparameters optimization outcomes. The two nature-based metaheuristics are then detailed in Sections 3.3 and 3.4. Here, we elaborate on both the variants of the Simulated Annealing technique – SA-2CM and SA-nCH. Then, we will break down the Genetic Algorithm and talk about its repair rules.

4.1. The Mixed-Fleet Multi-Terminal electric bus Scheduling problem

In this section, we formulate the problem of dispatching a mixed-fleet of $I = \{1,\dots,i\}$ electric buses and $H = \{1,\dots,h\}$ hybrid or conventional combustion buses to serve a set of scheduled trips $J = \{1,\dots,j\}$ characterised each by a scheduled departure time $D_j = \{1,\dots,d_j\}$ and duration $T_j = \{1,\dots,t_j\}$ with $d_j, t_j \in \tau \forall j$ and $\tau = [0,1,\dots,N]$ the discretisation of time in consecutive time steps, each with duration T_s. Each trip is further characterised by an energy requirement $U_j = \{1,\dots,u_j\}$ [kWh], a departure and arrival terminal $\alpha_j, \beta_j \in B$ with $B = \{1,\dots,b\}$ the set of bus terminals defined in the given transportation network, including any depot wherein buses can be stored when not in service. We define a subset $\overline{B} \subseteq B$ of bus terminals equipped with m charging stations each. Two matrices, representing all possible deadheading trip combinations over the set of terminals B, are defined as $\widehat{U} = \{\widehat{u}_{b_o,b_d}\}\forall(b_o,b_d) \in B : o \neq d$ and $\widehat{T} = \{\widehat{t}_{b_o,b_d}\}\forall(b_o,b_d) \in B : o \neq d$, respectively capturing the given deadheading trip's required energy [kWh] and duration [time steps]. Binary decision variables $y^t_{i,j}$ and $z^t_{h,j}$ capture whether trip j has been initiated by either electric bus i or hybrid bus h at time t. Similarly, binary variables ω^t_{i,b_o,b_d} and k^t_{h,b_o,b_d} serve to indicate whether electric bus i or hybrid bus h has initiated a deadheading trip from terminal b_o to terminal b_d at time t. Diverging from the original formulation, a key distinction is that deadheading trips performed by conventional buses were not explicitly captured to limit the model's complexity. These trips were instead computed in a post-processing phase. Charging events find expression through the variable $x^t_{i,b,m}$, representing whether a charging event has taken place for bus i at charger m of terminal b at time t. As with our previous works, we assume that opportunity charging stations are capable of completing a full charge of the given bus fleet within a single time step T_s. Finally, the locations of electric and hybrid buses are tracked respectively by integer variables $g^t_{i,b}$ and $p^t_{h,b}$. Table 3 and Table 4 summarise the problem's variables and parameters.

Table 3
Problem variables.

Description	Variable	Domain
1 if e-bus i is initiating trip j at time t, 0 otherwise	$y^t_{i,j}$	$\{0,1\}$
1 if h-bus h is initiating trip j at time t, 0 otherwise	$z^t_{h,j}$	$\{0,1\}$
1 if e-bus i is being recharged at charging station m of bus terminal b at time t, 0 otherwise	$x^t_{i,b,m}$	$\{0,1\}$
1 if e-bus i is initiating a deadheading trip ω from terminal b_o to terminal b_d at time t, 0 otherwise	ω^t_{i,b_o,b_d}	$\{0,1\}$
1 if h-bus h is initiating a deadheading trip k from terminal b_o to terminal b_d at time t, 0 otherwise	k^t_{h,b_o,b_d}	$\{0,1\}$
Total residual energy in kWh at time t for bus i	ε^t_i	$[0,E]$
Slack variable ensuring consistency in energy consumption	s^t_i	$[0,E]$
1 if e-bus i is either located at or heading towards terminal b at time t, 0 otherwise	$g^t_{i,b}$	$\{0,1\}$
1 if h-bus h is either located at or heading towards terminal b at time t, 0 otherwise	$p^t_{h,b}$	$\{0,1\}$

Table 4
Problem parameters.

Description	Parameter	Domain
Total energy in kWh required to perform trip j	u_j	$\mathbb{R}^+$
Preferred departure time step for trip j	d_j	$\mathbb{Z}^+$
Duration in time steps of trip j	t_j	$\mathbb{Z}^+$
Initial energy in kWh available for e-bus I at time 0	$\overline{\varepsilon_i}$	$[0,E]$
Energy cost for a full charge at time t	q^t	$\mathbb{R}^+$
Total battery capacity in kWh for all e-buses	E	$\mathbb{R}^+$
Minimum acceptable battery charge threshold in kWh	μ	$\mathbb{Z}^+$
Departure bus terminal of trip j	α_j	$\mathbb{Z}^+$
Arrival bus terminal of trip j	β_j	$\mathbb{Z}^+$
Total energy in kWh required to perform a deadheading trip from terminal b_o to terminal b_d	$\widehat{u}_{b_o,b_d}$	$\mathbb{R}^+$
Duration in time steps of a deadheading trip from terminal b_o to terminal b_d	$\widehat{t}_{b_o,b_d}$	$\mathbb{Z}^+$
Initial location of e-bus i	G_i	$\mathbb{Z}^+$
Initial location of h-bus h	P_h	$\mathbb{Z}^+$

Building upon the above-described components and variables, the resulting MILP formulation begins with the following objective function, which minimises the total operational cost of the schedule:

$$\sum\nolimits_t\sum\nolimits_i\sum\nolimits_j c\bullet y^t_{i,j}+\sum\nolimits_t\sum\nolimits_h\sum\nolimits_j \widehat{c}\bullet z^t_{h,j}+\sum\nolimits_t\sum\nolimits_i\sum\nolimits_{b_o}\sum\nolimits_{b_d}\overline{c}\bullet \omega^t_{i,b_o,b_d}+\sum\nolimits_t\sum\nolimits_i\sum\nolimits_b\sum\nolimits_m q^t\bullet x^t_{i,b,m}\#(1)$$

where $c=\eta_1\cdot U_j, \widehat{c}=\eta_2\cdot U_j, \overline{c}=\eta_1\cdot\widehat{U}$ are weighted costs related to, respectively, executing trips J via electric buses, hybrid buses, and deadheading trips. Energy components η_1,η_2 convert the trip's energy requirements into monetary values, with η_2 including an adaptation coefficient to represent the different consumption rate of h-buses. The penalty cost term r [EUR], which in the original formulation was applied to quantify the delays from the planned departure time, has been removed such that a Just-In-Time approach (JIT) is considered ($r = 0$). The JIT approach is further considered by the approximate approaches. System dynamics are captured by the following constraints:

$$\sum\nolimits_j y^t_{i,j}+\sum\nolimits_{b_o,b_d\in B}\omega^t_{i,b_o,b_d}+\sum\nolimits_{b\in\overline{B}}\sum\nolimits_m x^t_{i,b,m}\le 1,\forall i,t\#(2)$$

$$y^{d_j}_{i,j}+\frac{1}{t_j-1}\sum\nolimits_{\overline{t}=d_j+1}^{d_j+t_j-1}\left(\sum\nolimits_{\overline{j}}y^{\overline{t}}_{i,\overline{j}}+\sum\nolimits_{b_o,b_d\in B}\omega^{\overline{t}}_{i,b_o,b_d}+\sum\nolimits_{b\in\overline{B}}\sum\nolimits_m x^{\overline{t}}_{i,b,m}\right)\le 1,\forall i,\forall j:t_j>1\#(3)$$

$$\omega^t_{i,b_o,b_d}+\frac{1}{\widehat{t}_{b_o,b_d}-1}\sum\nolimits_{\overline{t}=t+1}^{t+\hat{t}_{b_o,b_d}-1}\left(\sum\nolimits_{\overline{j}}y^{\overline{t}}_{i,\overline{j}}+\sum\nolimits_{\overline{b_o},\overline{b_d}\in B}\omega_{i,\overline{b_o},\overline{b_d}}^{\overline{t}}+\sum\nolimits_{b\in\overline{B}}\sum\nolimits_m x^{\overline{t}}_{i,b,m}\right)\le 1,\forall i,\forall(b_o,b_d)\in B:\widehat{t}_{b_o,b_d}>1,\forall t\#(4)$$

$$\sum\nolimits_j z^t_{h,j}+\sum\nolimits_{b_o,b_d\in B}k^t_{h,b_o,b_d}\le 1,\forall h,t\#(5)$$

$$z^t_{h,j}+\frac{1}{t_j-1}\sum\nolimits_{\overline{t}=d_j+1}^{d_j+t_j-1}\left(\sum\nolimits_{\overline{j}}z^{\overline{t}}_{h,\overline{j}}+\sum\nolimits_{b_o,b_d\in B}k^{\overline{t}}_{h,b_o,b_d}\right)\le 1,\forall h,\forall j:t_j>1\#(6)$$

$$k^t_{h,b_o,b_d}+\frac{1}{\widehat{t}_{b_o,b_d}-1}\sum\nolimits_{\overline{t}=t+1}^{t+\hat{t}_{b_o,b_d}-1}\left(\sum\nolimits_{\overline{j}}z^{\overline{t}}_{h,\overline{j}}+\sum\nolimits_{\overline{b_o},\overline{b_d}\in B}k_{h,\overline{b_o},\overline{b_d}}^{\overline{t}}\right)_{\le 1,\forall h,\forall(}b_o,b_d)_{\in B:}\ \widehat{t}_{b_o,b_d}>1,\forall t\#(7)$$

$$\sum\nolimits_t\left(\sum\nolimits_i y^t_{i,j}+\sum\nolimits_h z^t_{h,j}\right)=1,\forall j\#(8)$$

$$\sum\nolimits_{t\ne d_j}\left(\sum\nolimits_i y^t_{i,j}+\sum\nolimits_h z^t_{h,j}\right)=0,\forall j\#(9)$$

$$\sum\nolimits_i x^t_{i,b,m}\le 1,\forall m,t,b\in\overline{B}\#(10)$$

$$y^{d_j}_{i,j}-\frac{\varepsilon^{d_j}_i}{u_j+min_{\beta_j\notin\overline{B},b_d\in\overline{B}}\left(\widehat{u}_{\beta_j,b_d}\right)+\mu\cdot E}\le 0,\forall i\forall j\#(11)$$

$$\omega^t_{i,b_o,b_d}-\frac{\varepsilon^t_i}{\widehat{u}_{b_o,b_d}+\mu\cdot E}\le 0,\forall i,t,(b_o,b_d)\in B\#(12)$$

$$\varepsilon^t_0=\overline{\varepsilon}_i,\forall i\#(13)$$

$$E\cdot\sum\nolimits_{b\in\overline{B}}\sum\nolimits_m x^t_{i,b,m}-\sum\nolimits_j y^t_{i,j}\cdot u_j-\sum\nolimits_{b_o,b_d\in B}\omega^t_{i,b_o,b_d}\cdot\widehat{u}_{b_o,b_d}+\varepsilon^t_i-s^t_i=\varepsilon^{t+1}_i,\forall i,t\#(14)$$

$$\sum\nolimits_{b\in\overline{B}}\sum\nolimits_m x^t_{i,b,m}-\frac{s^t_i}{E}\ge 0,\forall i,t\#(15)$$

$$\frac{1}{E}\cdot s^t_i-\frac{1}{E}\cdot\varepsilon^t_i\le 0,\forall i,t\#(16)$$

$$\sum\nolimits_m x^t_{i,b,m}-g^t_{i,b}\le 0,\forall t,i,b\in\overline{B}\#(17)$$

$$\sum\nolimits_j y^t_{i,j}+\sum\nolimits_{b_d\in B}\omega^t_{i,b_o,b_d}-g^t_{i,b_o}\le 0,\forall i,t,b_o\in B,j:\alpha_j=b_o\#(18)$$

$$\sum\nolimits_j y^t_{i,j}+\sum\nolimits_{b_o\in B}\omega^t_{i,b_o,b_d}-g^{t+1}_{i,b_d}\le 0,\forall i,t,b_d\in B,j:\beta_j=b_d\#(19)$$

$$\sum\nolimits_j y^t_{i,j}+\sum\nolimits_{b_o\in B}\omega^t_{i,b_o,b_d}-\left(g^{t+1}_{i,b_d}-g^t_{i,b_d}\right)\ge 0,\forall i,t,b_d\in B,j:\beta_j=b_d\#(20)$$

$$\sum\nolimits_b g^t_{i,b}=1,\forall i,t\#(21)$$

$$g^0_{i,b}=\begin{cases}1,b=G_i\\0,otherwise\end{cases},\forall i,b\#(22)$$

$$\sum\nolimits_h z^t_{h,j}+\sum\nolimits_{b_d\in B}k^t_{h,b_o,b_d}-p^t_{h,b_o}\le 0,\forall h,t,b_o\in B,j:\alpha_j=b_o\#(23)$$

$$\sum\nolimits_h z^t_{h,j}+\sum\nolimits_{b_o\in B}k^t_{h,b_o,b_d}-p^{t+1}_{h,b_d}\le 0,\forall h,t,b_d\in B,j:\beta_j=b_d\#(24)$$

$$\sum\nolimits_h z^t_{h,j}+\sum\nolimits_{b_o\in B}k^t_{h,b_o,b_d}-\left(p^{t+1}_{h,b_d}-p^t_{h,b_d}\right)\ge 0,\forall h,t,b_d\in B,j:\beta_j=b_d\#(25)$$

$$\sum\nolimits_b p_{h,b}^t = 1, \forall h, t\#(26)$$

$$p_{h,b}^0 = \begin{cases} 1, b = P_h \\ 0, otherwise \end{cases}, \forall h, b\#(27)$$

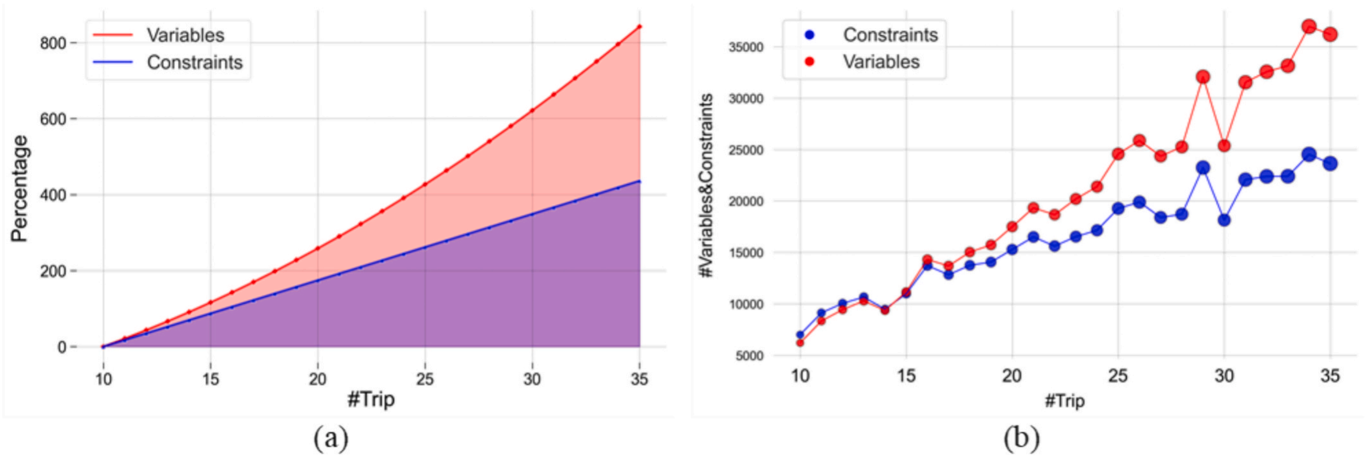
The modified formulation differs from the original version in three key ways. Firstly, we made modifications to the constraints associated with the time-decomposition, specifically equations (2) and (5), to address full-size instances. Secondly, we expanded the set of constraints related to hybrid coverage, encompassing constraints (5–7) and (23–25), to account for possible deadheading activities. This expansion overlaps with the binary variable k_{h,b_o,b_d}^t. Additionally, we consider the JIT approach by eliminating delay terms from constraints (3), (6), (9), and (11).

Equations (2–4) govern the trip execution of electric buses. Specifically, (2) ensures that an e-bus can only initiate either a regular trip j, a deadheading trip ω, or perform a charging action at any given time, as long as it is not otherwise occupied. Constraints (3–4) ensure that e-buses remain unavailable to perform any action for the entire duration of, respectively, service trips and deadheading trips. Constraints (5–7) concern trip and deadhead execution for hybrid buses, similarly, ensuring that no activity can be performed concurrently as long as a previous service trip or deadheading is not complete. Equations (8–9) ensure that *all* trips be eventually executed by either e-bus or h-bus, and that no trip is executed hat no trip is executed before or after its scheduled departure time d_j. Constraints (10–16) capture the energy dynamics of the e-buses, in terms of evolution of the state of charge and recharging events. Constraint (10) ensures that charging actions, taking place solely at terminals equipped with charging facilities, involve an individual e-bus at any given time. Constraint (11) ensures that the execution of a trip by an e-bus can only be performed if it has sufficient energy to do so, such that after performing a service trip the leftover energy is at least both above the minimum threshold value and bears sufficient spare energy to perform a single deadheading trip to the closest charging facility. Similarly, Equation (12) imposes that deadheading trips can only be performed if enough energy is available and on condition that the minimum energy threshold is respected. Constraint (13) initialises the energy of each bus to the given parametric input. Constraint (14) captures the state of charge dynamics: the evolution of the available energy obeys one of four potential scenarios: if a service trip is executed by bus i at time t, its remaining energy will be reduced by the amount required for the trip performance at time $t + 1$. The same dynamic is established for deadheading trips. If a charging action is performed at time t, the state of charge at time $t+1$ will be a full charge, following the assumption that opportunity chargers are indeed capable of completely recharging a battery pack during a single time step. Finally, if no trip or action is performed, the energy state is maintained. Constraints (15–16) ensure the correct functionality of (14) by regulating the slack variable s_i^t.

Equations (17–22) govern the location dynamics of e-buses and Equations (23–27), similarly, those of h-buses. Equation (17) ensures that charging actions can only take place at a given charging station provided that bus i is indeed located at the appropriate terminal b. Constraint (18) ensures that either service or deadheading trips departing from a given bus terminal can only be performed by buses located in said terminals. Constraint (19) ensures that an e-bus location is updated to the arrival terminal if either service or deadheading trips are performed. Equation (20) guarantees that locations remain unchanged unless a trip of either kind is performed, while Equation (21) that e-buses must occupy a single given location at any given time. Constraint (22) relays the parametric initial location choice to each e-bus. Constraints (23–27) mimic exactly the dynamics of (17–22) for hybrid buses, including deadheading as a potential source of changes in h-bus location.

In Section I.2 of the Supplementary Materials, we demonstrate that the limitations of the MILP model are primarily linked to its scalability.

Specifically, while the number of constraints increases linearly with the number of trips, the number of variables exhibits non-linear growth, as illustrated in Fig. 2. This exponential growth is attributed to two main sets: the time slots denoted as τ and the terminals represented by set B. The computational complexity is particularly sensitive to the set τ, which discrete size is determined by the trip j with the $max(d_j + t_j)$. For each time slot, the model construction phase generates variables and constraints associated with fleet assignment, charging, and deadheading activities. Moreover, the number of terminals $|B|$ significantly impacts the aforementioned constraints, as it multiplies the relation arcs for each terminal. This is why we explore multiple dimensions for the dataset, as discussed in Section 2. Another factor contributing to the computational burden is the percentage of e-buses within the fleet. The formulation associates a higher number of constraints and variables with these vehicles to establish feasible charging schedules, taking into account the limited autonomy of electric batteries. Due to these limitations, the MILP approach can effectively handle only micro instances comprising up to 60 trips and 4 terminals within a reasonable processing time. Therefore, the proposal of approximate approaches becomes imperative to manage urban network scenarios that extend across multiple dimensions, encompassing trips, terminals, and bus lines.



(a)

(b)

Fig. 2. (a) Depicts the percentage trends in the number of constraints and variables as instance size increases, with a fixed proportional number of time slots. Notably, the number of variables exhibits exponential growth, while the number of constraints demonstrates a linear increase; (b) illustrates these trends in number but under the influence of randomly generated instances.

4.2. Chain-Trip Builder: A Repeated local search

The limitations discussed in the linear programming model above and in Section I.2 of the Supplementary Materials emphasize the critical need for the development of faster heuristic algorithms capable of preserving high solution quality. Additionally, both of the proposed metaheuristic approaches outlined in Sections 3.3 and 3.4 necessitate a robust initial foundation. An optimal starting point can significantly accelerate the convergence of both Simulated Annealing and Genetic Algorithm. For instance, when the initial population consists of individuals already in close proximity to optimal solutions, GAs can yield positive results with fewer generations. Similarly, a well-selected initial solution streamlines the tuning process for SA, reducing the number of iterations required and enabling more precise parameter adjustments. Furthermore, GAs face the challenge of striking a balance between exploration (wide-ranging search across the solution space) and exploitation (focused attention on promising regions). An ideal initial population can bias the algorithm towards early-stage exploitation, thereby enhancing the likelihood of rapidly identifying exceptional solutions while still permitting exploration later in the optimization process. Lastly, it is worth noting that many metaheuristic methods can get trapped in local minima or settle for suboptimal solutions. A strong initial solution can aid both Simulated Annealing and Genetic Algorithms in commencing their search closer to the global optimum, significantly increasing the probability of escaping local minima and converging towards a global solution.

First, we introduce the feasible solution generated by the heuristic, referred to as a *Block*. The block object encapsulates a set of activated buses, along with their corresponding sequences of trips (chains). A graph visualization of a block is presented in Fig. 3. Each row in the visualization illustrates a sequence of feasible events associated with the corresponding bus code on the left. As indicated in the legend, bus codes can be linked to e-buses (depicted by green squares) or h-buses (represented by grey squares). These events can encompass trips (blue circles), deadheads (orange circles), or full charges (red inverted triangles). Ad In terms of solution structure, there are two possible perspectives: the trip-based array and the bus-based block. In this study, the choice of structure depends on the algorithmic approach. Both the heuristic and Simulated Annealing utilize the block object, whereas the Genetic Algorithm employs an array structure. The length of the array corresponds to the number of trips to cover, with each entry populated by the bus code assigned to that specific trip. ditionally, the x-axis represents the chronological distribution of these events throughout the day. The constructive heuristic introduced here is a Repeated Local Search termed the “Chain-Trip Builder” (CTB). In contrast to an Iterated Local Search (ILS), RLS does not leverage information from solutions generated in previous iterations.

The primary objective of CTB is to construct blocks with extended chains of events. The idea is that longer chains can reduce the number of buses activated. The CTB pseudocode is presented in Algorithm 1. The algorithm’s workflow is outlined as follows:

- **Data Pre-processing.** At the initial stage, the algorithm ingests either synthetic or real-world timetables. These timetables provide comprehensive information for each scheduled trip, including departure time, origin, destination, time required, and energy consumption. Additionally, essential supplementary data is provided, such as unitary costs, fleet availability, the number of terminals (with or without charging stations), and the quantity of charging stations. Some hyperparameters, such as the *window size w*, have been fine-tuned through a grid search process, and the optimal parameter configuration is also presented;
- **Bus type selection.** The heuristic iteratively scrolls the timetable to determine if the current trip is covered or not. This process continues until all scheduled trips are covered. The heuristic employs an e-bus or an h-bus initiation strategy based on a first probability parameter, denoted as p_1. Three distinct bus type selection probabilities have been defined:

1 **Uniform probability.** The probability of initializing an e-bus for an uncovered trip is uniformly distributed over the half-open interval $[low, high)$

$$p_1 = \frac{1}{high - low}\#(28)$$

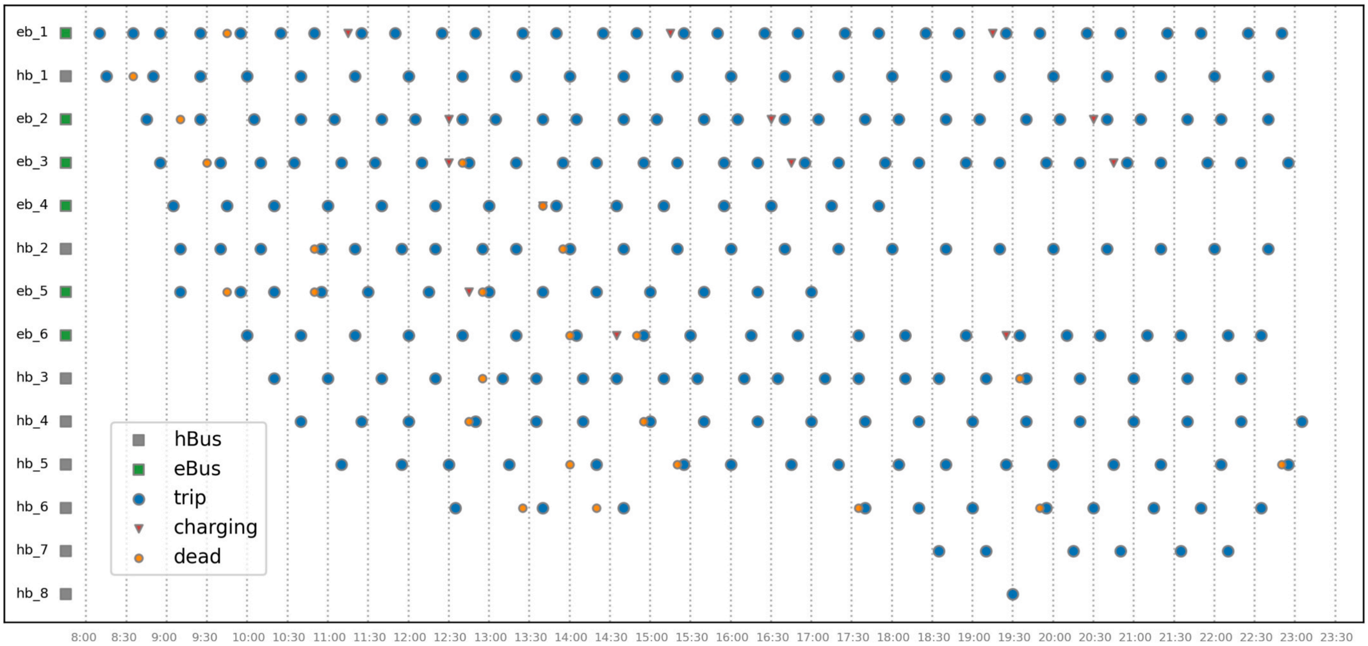


Fig. 3. Graph visualization of a block object, with bus codes on the left and the corresponding chain of events.

```
Input: Set of trips $J$, electric buses $I$, hybrid buses $H$;
Trip data: departure times $D_j$, durations $T_j$, energy needs $U_j$,
terminals $\alpha_j, \beta_j$;
Deadhead matrices $\hat{T}$, $\hat{U}$;
Charging station info, hyperparameters: window size $w$, iterations $k$, $p_1$
strategy, $p_2$ strategy
Output: Best block with minimum operational cost
Initialize $bestBlock \leftarrow \emptyset$;
for $i \leftarrow 1$ to $k$ do
    Set all trips as uncovered;
    Initialize $currentBlock \leftarrow \emptyset$;
    while uncovered trips exist do
        Select uncovered trip $j_0$;
        Choose bus type using $p_1$ strategy;
        Assign new bus ID, initialize new chain from $j_0$;
        while true do
            Generate feasibility window $W$ of size $w$ for next trips;
            if $W = \emptyset$ then
                break;
            foreach $j \in W$ do
                Evaluate all event combinations (direct, deadhead,
                  charge);
                Check feasibility (time, location, SoC, charger
                  availability);
                Compute rank via Eq. (33) and $p_2$ probability via
                  Eq. (34);
            Select next trip from $W$ using $p_2$ distribution;
            Add selected trip and events to current chain;
        Append completed chain to $currentBlock$;
    if cost(currentBlock) < cost(bestBlock) then
        $bestBlock \leftarrow currentBlock$;
return $bestBlock$
```

2 **V1 probability.** This probability p_1 is computed based on various factors, including the Euler number, the ratio of trips currently covered θ, and γ, which considers the trips covered per single e- or h-bus activated. As the rate of e-bus usage r_e increases, the likelihood of the total e-bus operational cost surpassing that of h-buses (owing to factors like charging and deadheading) also rises, consequently reducing the probability of selecting an e-bus

$$\theta = \frac{trip_{covered}}{Tot_{trip}} \#(29)$$

$$\gamma = \frac{r_h}{r_h + r_e} where r_e = \frac{trip_{covered,e}}{ebus_{activated}} \bullet 100\#(30)$$

$$p_1 = 1 - \theta \bullet e^{-\gamma} \#(31)$$

3 **Uniform + V1 probability.** The probability for bus type selection is set to follow the *uniform* probability if the number of trips covered is below a specific threshold δ. However, if the number of covered trips exceeds this threshold, the probability switches to *V1*. This approach is designed to mitigate the initial e-bus selection probability by covering a certain number of trips before transitioning to *V1*

$$p_1 = \begin{cases} \frac{1}{high - low}, trip_{covered} \leq \delta \\ 1 - \theta \bullet e^{-\gamma}, otherwise \end{cases} \#(32)$$

- **Chain building.** After determining the bus type, the algorithm assigns the bus code to the currently uncovered trip, initiating the construction of the chain (as depicted in Fig. 4). For each current trip covered by the bus, the heuristic defines a certain number of potential feasible next trips, referred to as *window*. One of them is then selected based on a second probability, denoted as p_2. This feasibility is defined not only in terms of temporal and spatial compatibility, that is, whether the arrival time and terminal of the current trip align with the departure time and location of a candidate next trip, but also by the battery energy levels of electric buses. To assess feasibility, the heuristic generates and evaluates a structured set of event combinations, as detailed in Fig. 24 of Section II of the Supplementary Materials. These combinations represent possible sequences of operational events, such as deadheading, charging, or direct continuation, that may be required for a vehicle to transition from the current trip to a candidate one. Each path in the event combination tree corresponds to a distinct feasibility scenario, which is evaluated in terms of time compatibility, charging infrastructure availability, and battery state-of-charge. Only those event combinations that result in a valid energy trajectory (i.e., where the vehicle retains sufficient energy to reach and execute the next trip, potentially after recharging) are retained. If at least one valid combination exists for a given candidate trip, it is included in the feasibility window. If the window is not empty, p_2 a probability weight for each trip within the window. This weight is based on three key parameters: the position pos_i of the trip within the window and the count of deadheading dd_i and charging ch_i events required to cover that

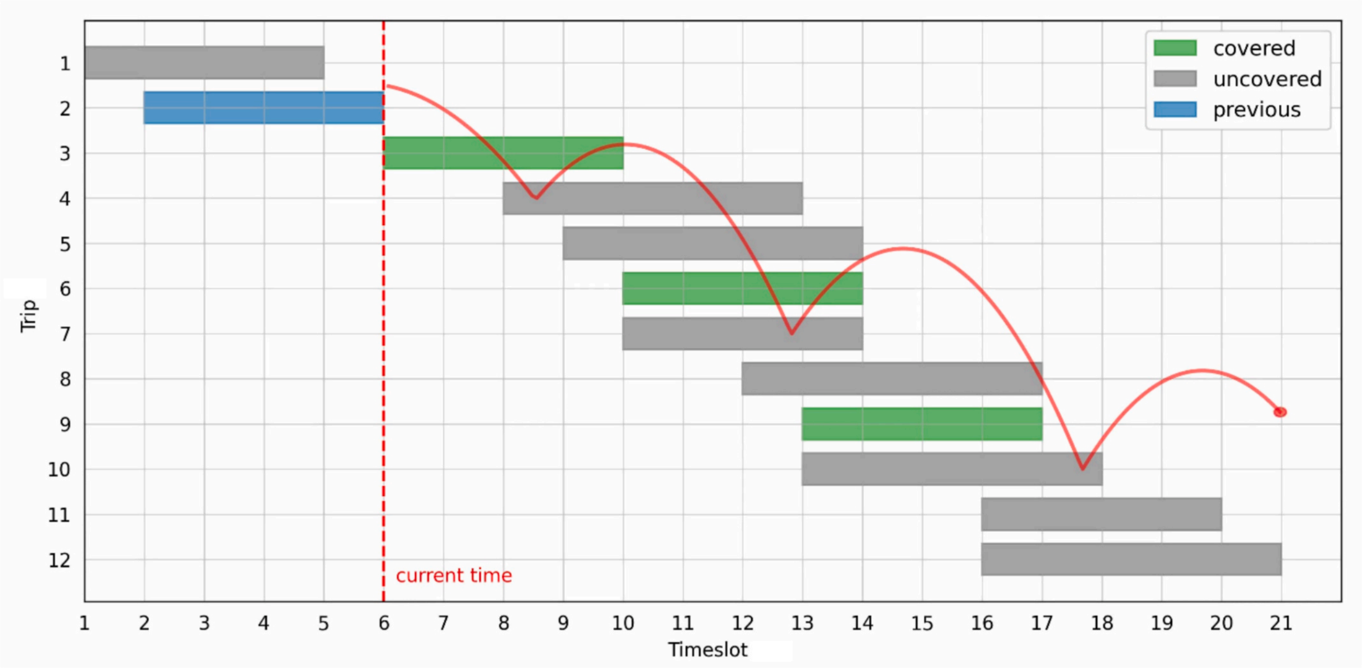


Fig. 4. The principle behind the chain of trips generation. Considering the arrival time of the current trip (blue rectangle), the heuristic selects one of the next feasible trips that have not yet been covered (grey rectangle).

particular trip (Eq. (33)). Notably, a higher position within the window and an increased number of deadheads or full charges result in a lower probability weight. Initially, a ranking process assigns a weight to each cost-efficient event combination for covering a feasible trip (Eq. (34)).

Subsequently, the contribution of each combination to the total weight determines its ranking position, thereby influencing the associated probability.

$$rank_i = pos_i + dd_i + ch_i\#(33)$$

$$p_{2,i} = \frac{rank_i}{\sum_{i\in W} rank_i}\#(34)$$

- **Chain and Block generation.** When no other feasible trips exist within the created time window, the heuristic proceeds to generate both the chain of trips and the corresponding detailed chain of events for the bus. Moreover, when all the trips scheduled in the timetable have been assigned, the heuristic returns the block object.

As it has been described so far, the CTB algorithm exhibits two primary limitations displayed in Fig. 5.a, which were identified during preliminary testing. Firstly, blocks generated may contain chain breaks (indicated by the blue halo), associated with e-buses. Secondly, it may exhibit an upper triangular shape (indicated by the red halo). Both of these limitations contribute to an increased requirement for fleet size to cover the timetable. Chain breaks occur when electric buses are unable to reach the nearest charging stations after completing a previous trip. This situation arises because the original function designed to create feasible event combinations does not account for potential stalls at terminals lacking charging stations when selecting the next feasible trip within the window.

To address and rectify this behavior, an attribute for forecasting disruptions has been integrated into the event combination generation function. This upgrade enables the system to anticipate potential disruptions, bypass the selection of the next trip, and redirect the e-bus to the nearest charging station for a full charge. Fig. 5.b provides an example of identified disruptions. Furthermore, to mitigate the aforementioned limitation related to block shape, a repeated approach to block generation has been introduced, with the aim of transforming the block shape from its upper triangular form into a more efficient trapezoidal configuration. The incorporation of these features is evident in Fig. 5.b, where longer chains are observed, resulting in a reduction in the number of h-buses required.

The computational complexity of a single CTB iteration is $O(|J|\cdot w\cdot e)$, where $|J|$ is the number of trips to schedule, w is the predefined window size (i.e., number of candidate next trips evaluated at each step), and e is the average number of event combinations considered for feasibility (e.g., deadheading, charging, continuation). Since the algorithm performs k independent iterations and retains the best block, the total time complexity becomes $O(k\cdot|J|\cdot w\cdot e)$. In practice, the algorithm remains efficient due to relatively small window sizes and bounded event evaluations. In our experimental setup (Section 4.3.1, see also Fig. 18), we adopt realistic parameter ranges that ensure computational tractability even for instances involving over 1000 trips.

The CTB algorithm is designed to generate an optimal block based on the provided timetable data and hyperparameter values. It allows for the exploration of various combinations of three key hyperparameters: the window size, the probability type for p_1 and the number of blocks generated before selecting the best one. To fine-tune these hyperparameters, we conducted experiments using small real-world instances comprising 257 trips and 5 terminals, two of which are equipped with charging stations. For a more comprehensive understanding of additional parameters, refer to Section 4 of this paper. We visualized the results of our hyperparameter tuning using a 3D representation in Fig. 6, while a tabulated summary of the Grid Search data can be found in Table 11 of the Supplementary Materials. The grid search systematically explores all possible combinations of the three parameters while adhering to predefined ranges: the window size varies between 2 and 10 next feasible trips, and the number of iterations considers values from the array $[50, 100, 150, 200]$. Each node in the grid represents a specific combination and records the best solution score obtained during the search process. The results show several noteworthy behaviors:

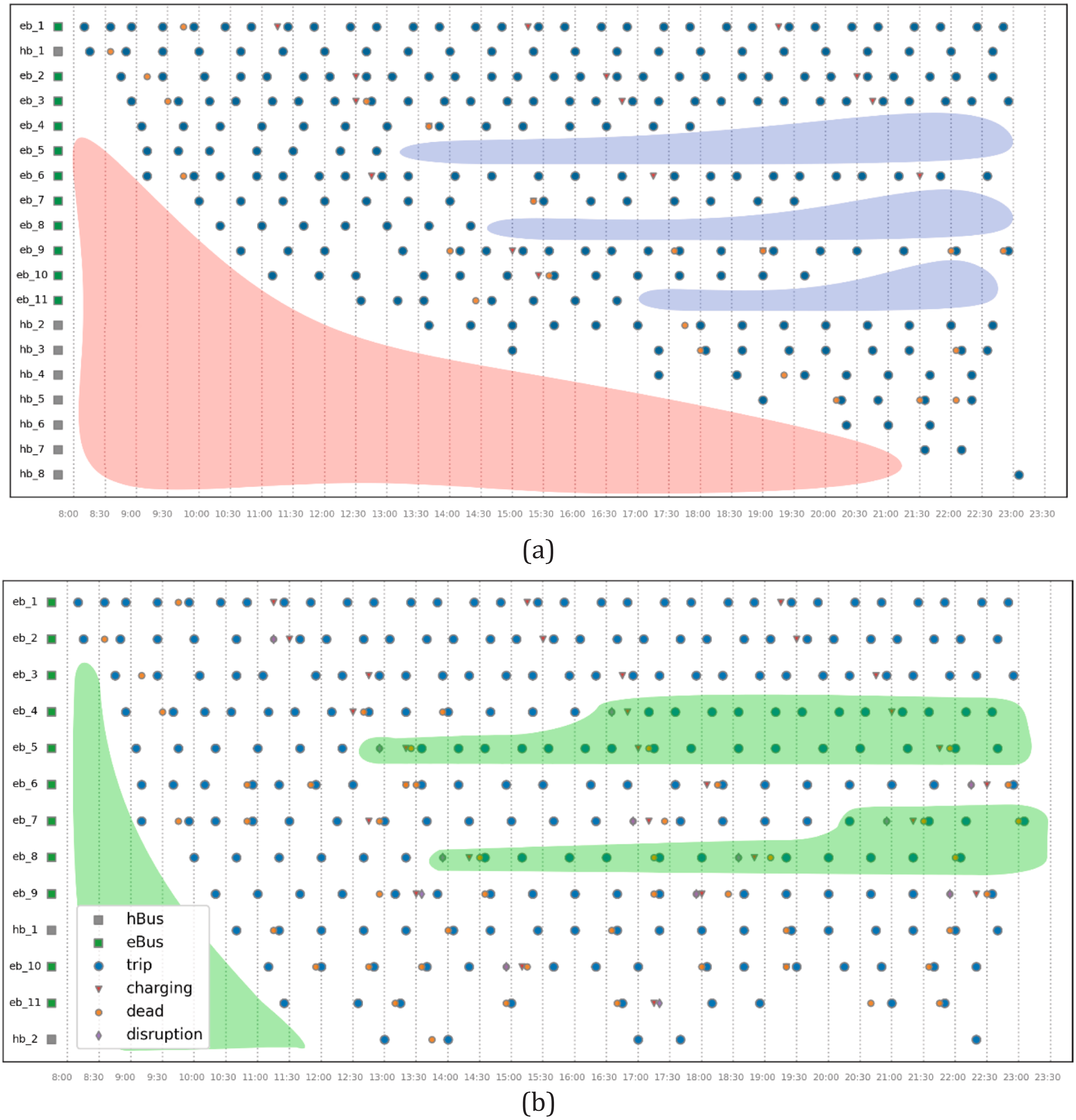


Fig. 5. Illustrations highlighting identified limitations of the original CTB (a) and subsequent enhancements addressing disruptions post-upgrade (b).

- Among the hyperparameters studied, the window size emerges as the most critical. The wider the window, the higher the variety of solutions explored and the better the best solution found, in terms of operational costs and fleet size. However, caution is required when selecting the window size, as excessively wide windows may lead to computational failures and prolonged processing times. Indeed, excessively large jumps in creating the chains can quickly deplete the available bus resources. It is observed that for the small instances considered, a window size of 7 proves to be a reasonable choice;
- The Uniform and V1 probability types consistently yield the best results. This indicates that initializing more e-buses at the outset tends to produce superior solutions;
- While the CTB algorithm significantly reduces processing times compared to the MILP approach, the number of blocks generated can influence computational efficiency. In this regard, the best combination involves 100 iterations. It is worth noting that exceeding this iteration count does not yield any noticeable improvements.

While the CTB can function effectively as a standalone tool, as demonstrated in the tests discussed in Section 4, it does exhibit certain limitations owing to its short-sightedness and the sequential block construction approach. Two primary limitations are noteworthy. The CTB employs reactive charging, meaning that the decision to perform a full charge is made only when the remaining energy is insufficient for

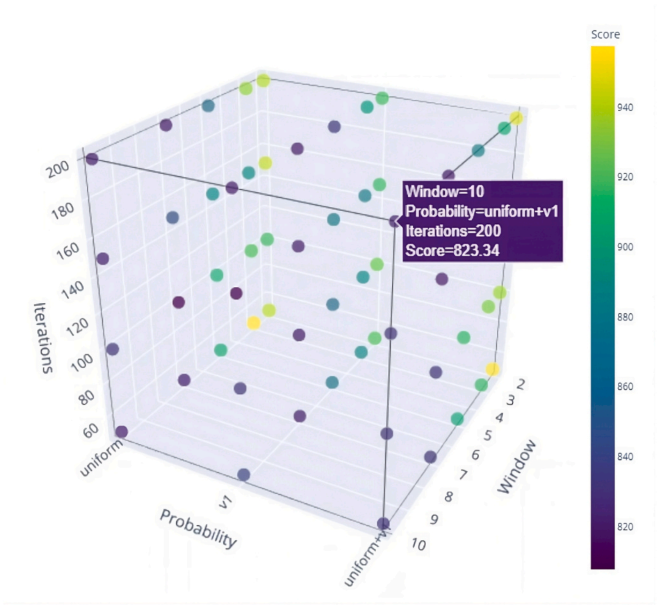


Fig. 6. 3D visualization of Grid Search results. Each node represents a unique combination of the three explored hyperparameters, along with its corresponding objective function value.

the next trip, and the e-bus has adequate time to reach the nearest charging station before the trip's departure time. This approach lacks pre-emptive charging, which could potentially restore feasibility for excluded subsequent trips. The second limitation lies in acknowledging that longer chains of trips do not necessarily translate into superior solutions. In fact, a chain of trips that deviates from the window constraint but exhibits a more optimal distribution can effectively act as a gap filler, reducing both the number of buses required and operational costs. Given these limitations, further exploration of the feasible solution space using metaheuristic approaches becomes imperative to avoid getting trapped in local minima.

4.3. Simulated Annealing: SA-2CM and SA-nCH

In this section, we present two customized versions of Simulated Annealing (Eglese, 1990), specifically tailored to address the challenges posed by the MFMTEBSP, with a primary focus on achieving scalability for handling large instances (Fig. 7). Several reasons support the adoption of this approximate approach for tackling the MFMTEBSP:

- **Promising algorithm.** SA has substantiated its potential as a promising methodology for addressing analogous problems, as evidenced by the empirical findings of previous research conducted by (Zhou et al., 2020; Ciancio et al., 2018). Nevertheless, the efficacy of SA is contingent upon precise hyperparameter calibration (i.e. initial temperature, cooling schedule type, etc.) and the design of the perturbation operator;
- **Single-solution vs population-based algorithm.** This motivation is mainly related to the metaheuristic comparison stage on real-size instances and the choice to preserve or not the solution feasibility. SA explores the solution space sequentially, one solution at a time, introducing small perturbations along the way. In contrast, GA operates with multiple solutions concurrently, employing processes like crossover and mutation. Furthermore, the acceptance mechanisms differ between these two methods, with SA relying on the probability of acceptance and GA employing a ranking-based approach;
- **Maintaining solution feasibility.** Still looking at the metaheuristic comparison, the crossover stage in GA frequently produces infeasible solutions, primarily due to constraints related to the autonomy of electric batteries. Conversely, the guiding principle behind the moves applied in the customized SA is to ensure the feasibility of solutions, thus obviating the need for repair rules. This principle aligns with the single-solution downhill approach, which relies on minor adjustments in the newly explored solutions.

The algorithm steps of the customized SA presented in our study closely mirror the conventional metaheuristic process. However, the primary focus of customization lies in the development of perturbation operators, the design of the cooling schedule, and the acceptance probability applied.

A detailed description of these phases follows below:

- **Data Pre-processing and CTB best solution.** As previously detailed in Section 3.2, the CTB leverages timetabled data pertaining to scheduled trips and finely-tuned hyperparameters. Upon completing all iterations, the CTB identifies the best block, which subsequently serves as the initial solution for the SA algorithm;
- **Perturbation.** At this stage, a potentially enhancing move is applied to the current solution, giving rise to two distinct SA versions: SA-2CM and SA-nCH. While these two variations employ different rescheduling techniques, they share a common underlying principle: the selection of two or more chains of trips from the current block and the subsequent rescheduling of trips, resulting in a new and feasible permutation. This pivotal phase of the SA process will be analyzed further below;
- **Cooling Schedule.** A plethora of cooling schedules have been rigorously examined in the literature, ranging from linear to quadratic, and spanning the spectrum from additive monotonic to non-monotonic adaptive schedules, as evidenced in studies by (Nourani and Andresen, 1998; Bertsimas and Tsitsiklis, 1993). In our investigation, we have adopted a logarithmic curve for its distinctive characteristics. The logarithmic curve starts with a steep decline in temperature, promoting rapid convergence, and gradually tapers off towards the end, allowing for a more demanding exploration of the solution space. Additionally, we explore an alternative variant of the original schedule, termed Logarithmical Multiplicative Cooling, which incorporates a cooling acceleration factor α (Eq. (35)). This adaptation proves to be useful in finding an effective balance between computational efficiency and the attainment of high-quality solutions.

$$T_k = \frac{T_0}{1+\alpha\log_{10}(1+k)}\#(35)$$

$$T_0 = \frac{\sum_{i:i\le k\wedge\Delta_i>0}\Delta_i}{|\{i : i\le k\wedge\Delta_i>0\}|}\#(36)$$

Another critical parameter to consider is the initial temperature T_0. An inaccurate calibration of this parameter can significantly impact the acceptance probability, leading to excessively high or low values. In our study, we incorporate an adaptive feature into the initial temperature selection process, which links it to the progress of the search (Eq.(36)). Initially, T_0 is determined based on the instance scale to address. Subsequently, should the acceptance probability fall below a threshold (<0.1) or exceed a threshold (>0.95) over a specified number of iterations, the temperature is recalibrated to maintain it as the average difference between the objective function values of the previous new and current solutions, specifically when the latter outperforms the former;

- **Acceptance Probability.** When a new solution is generated using the perturbation operator and yields a worse objective function value compared to the current solution, we leverage the Δ_k representing the difference in objective function values between the two solutions, in conjunction with the current temperature T_k to calculate the probability p_{acc} of accepting the new solution.

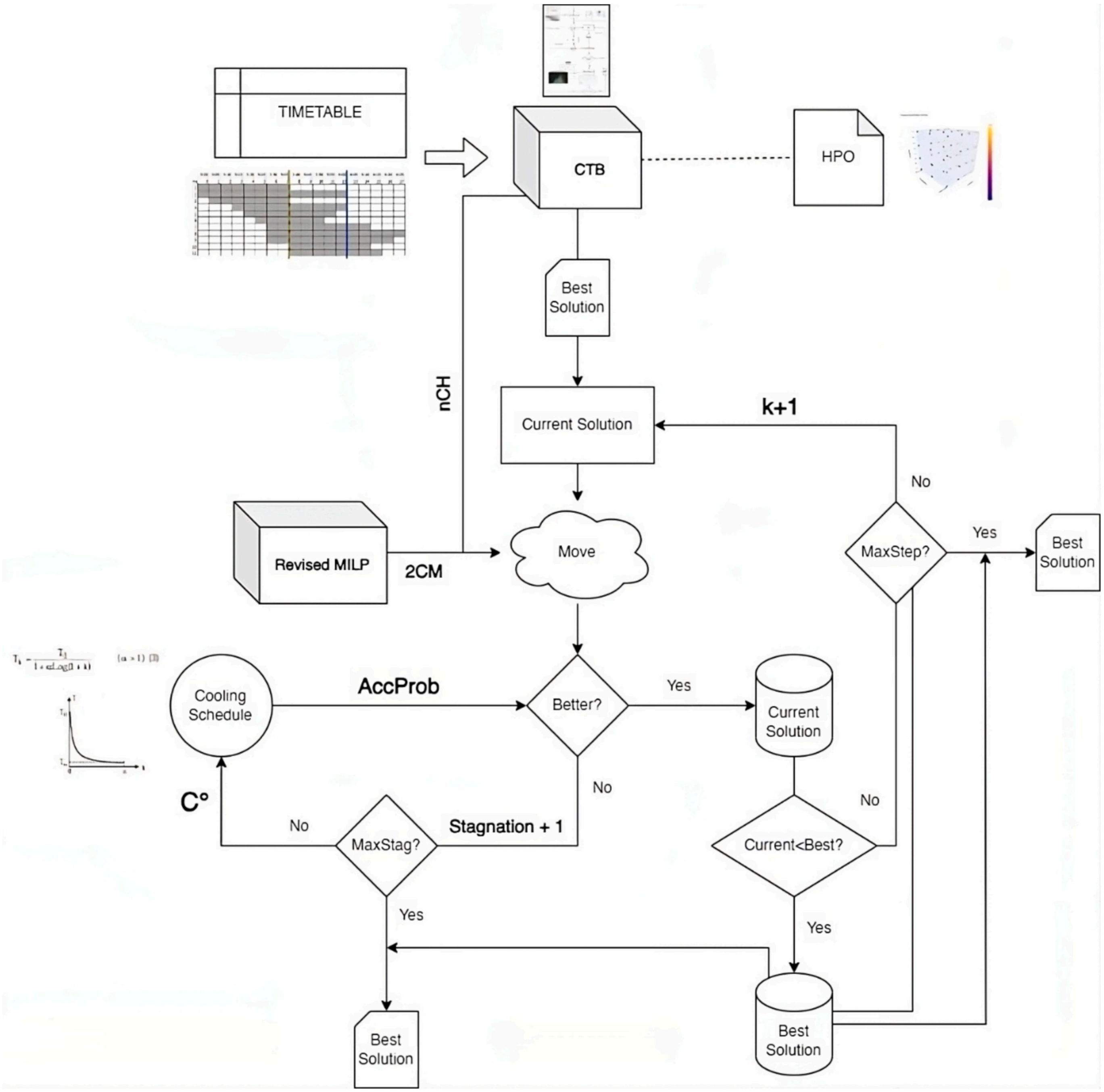


Fig. 7. Flowchart of the Simulated Annealing.

$$p_{acc} = e^{-\Delta_k/T_k}\#(37)$$

$$p_{acc} = \frac{1}{1+e^{\Delta_k/T_k}}\#(38)$$

Our preliminary tests have focused on evaluating two prominent acceptance probabilities: the Boltzmann and the Logistic probabilities, as described by Equations (37) and (38). Both of these probabilities are employed with equal weighting, each assigned a 0.5 probability of selection;

- **Exit conditions.** The while loop terminates based on two exit conditions – the best solution remaining stagnant and reaching the maximum allowable number of steps. Upon discovering a new best solution, the stagnation counter is reset; otherwise, it is incremented by one.

While various evolutionary metaheuristics, such as Genetic Algorithms, incorporate integrated moves like crossover, Simulated Annealing requires the development and customization of moves tailored to the specific problem at hand. In the context of the Electric Bus Scheduling Problem, applying perturbation faces a significant challenge due to the limited range of electric buses and the constraints imposed by charging schedules. This challenge pertains to maintaining a coherent energy consumption pattern throughout the chain of trips. In our study, we have devised two moves based on the same fundamental principle: the rescheduling of two or more chains of trips within the current solution block. It is important to note that randomly permuting trips

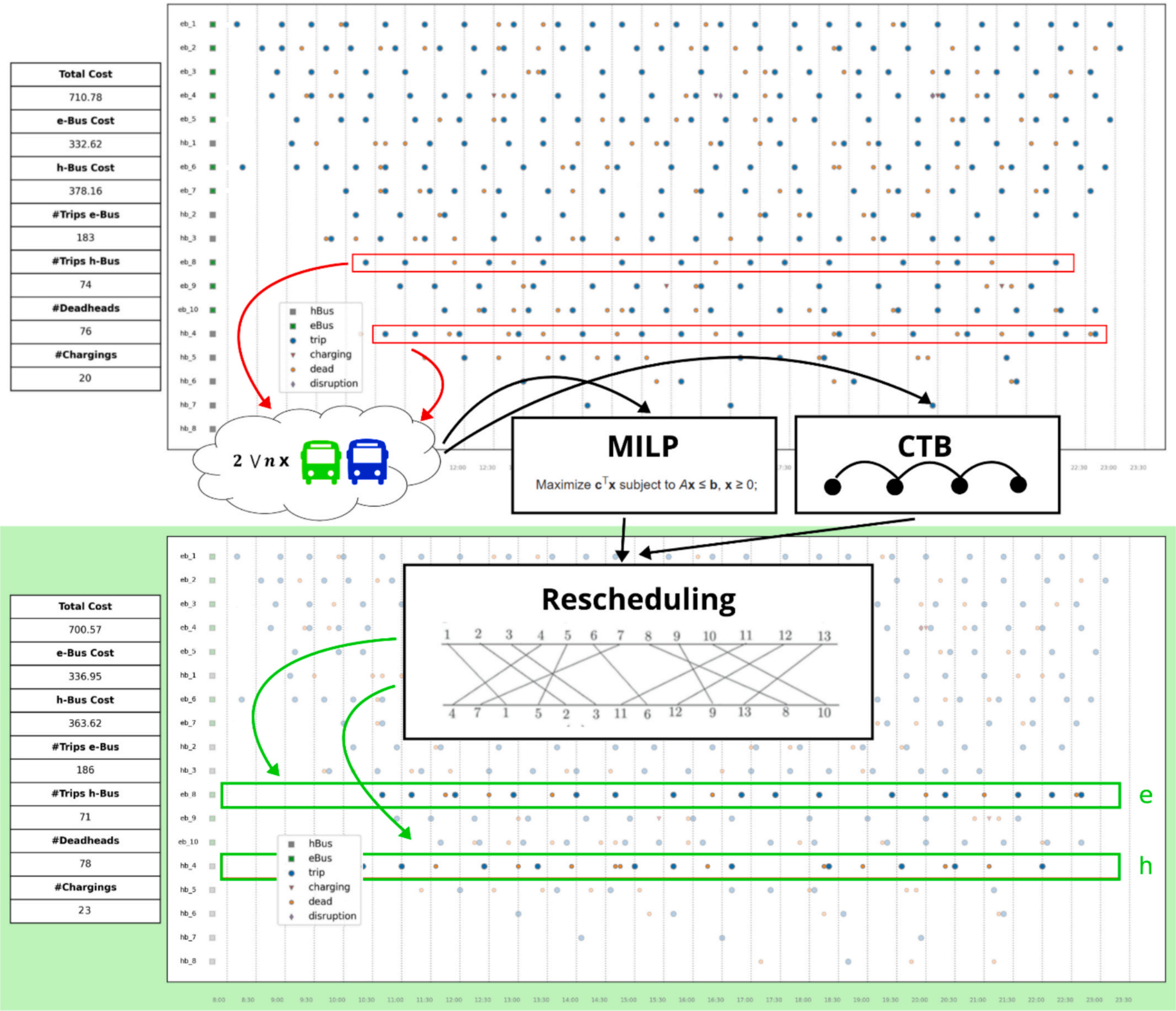


Fig. 8. The underlying rescheduling principle of perturbation operators.

between chains can result in infeasible new solutions with a high likelihood. This is because the execution of two consecutive trips is highly sensitive to both time constraints and the available energy, considering the possibility of extra events such as deadheads and full charges. To address this issue, the reassignment of trips to buses within the selected chain must leverage tools that ensure the feasibility of the trip sequence in the new solution. The rescheduling principle underpinning the developed moves is visually explained in Fig. 8.

The main steps involve:

- **Chains selection.** Entire or partial chains are randomly chosen from the current block. This random selection allows for the extraction of both heterogeneous and homogeneous sets of chains, considering bus type factors. This, in turn, could result in solutions with the same objective value but featuring rearrangements that can initiate the exploration of new regions in the solution space;
- **Tools selection.** Both moves employ customized versions of previously introduced tools designed for managing individual chains. The SA-2CM move selects exactly two chains, which are rescheduled using a modified MILP model This model is implemented through the Python library of IBM CPLEX, and it generates an optimal reassignment of trips between the two selected chains. On the other hand, the SA-nCH move selects a random number n of chains in each iteration, where n is proportional to the instance size, and uses a heuristic approach based on the CTB algorithm to find a feasible but sub-optimal reassignment. These differences, both in chain selection and rescheduling strategy, form the core distinction between SA-2CM and SA-nCH;
- **Integration of new chains.** The newly selected chains are then incorporated into the block, and the resulting objective function value is compared to the current one.

The search properties of the SA-2CM algorithm can be examined in Fig. 9. The instance involves 257 trips, with a maximum stagnation set at 20 and a maximum step limit of 100. Due to scalability concerns with solving exact MILP models, the number of chains selected for rescheduling in SA-2CM is fixed at two. The move consists of randomly selecting two chains from the current block and solving the MILP model via CPLEX to reassign trips in an optimal way. To reduce problem size before solving, a filtering step is applied that restricts the trip set to only the relevant time slots and terminals associated with the selected sub-timetable, significantly reducing the dimensions of sets T (time slots)

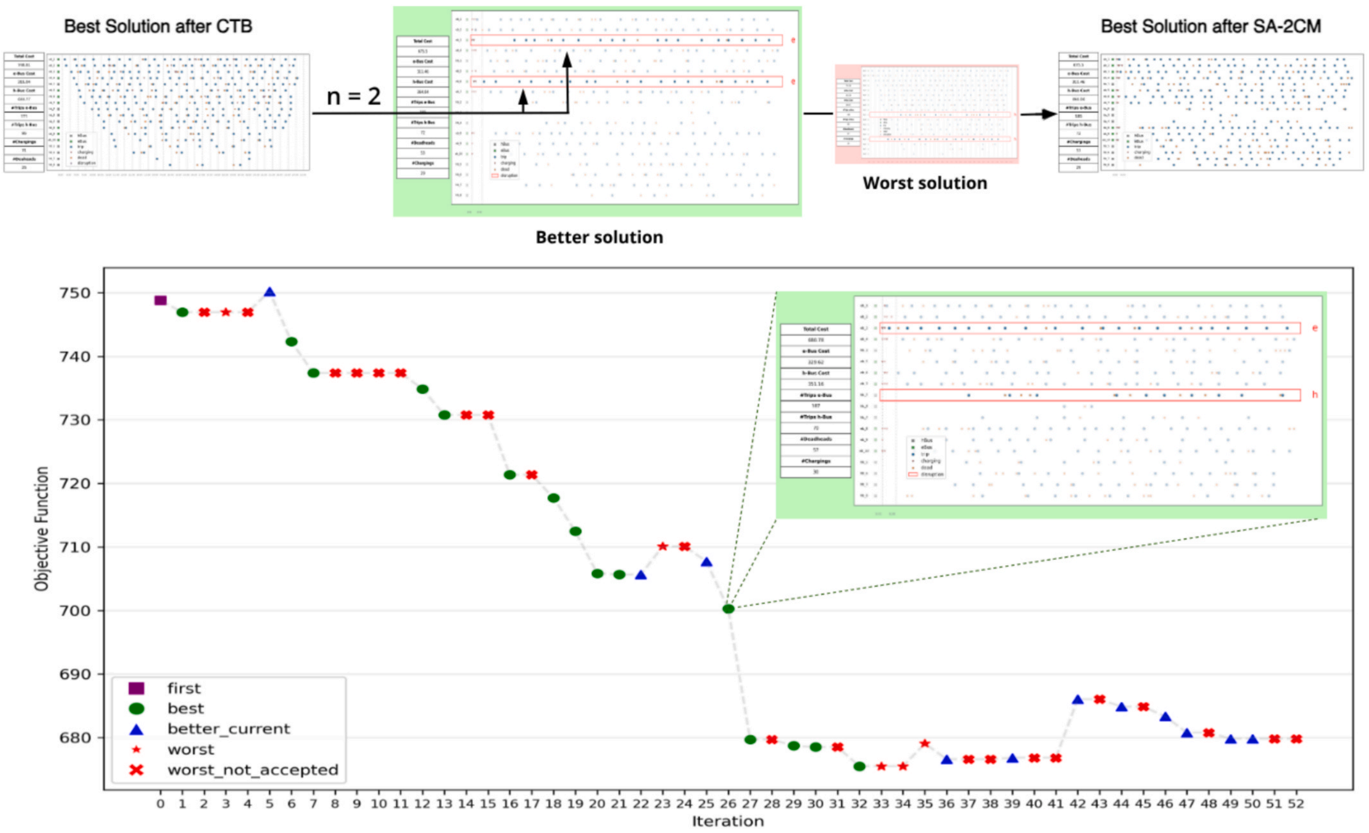


Fig. 9. SA-2CM search properties. The SA-2CM selects two chains from the current block and starts the descent.

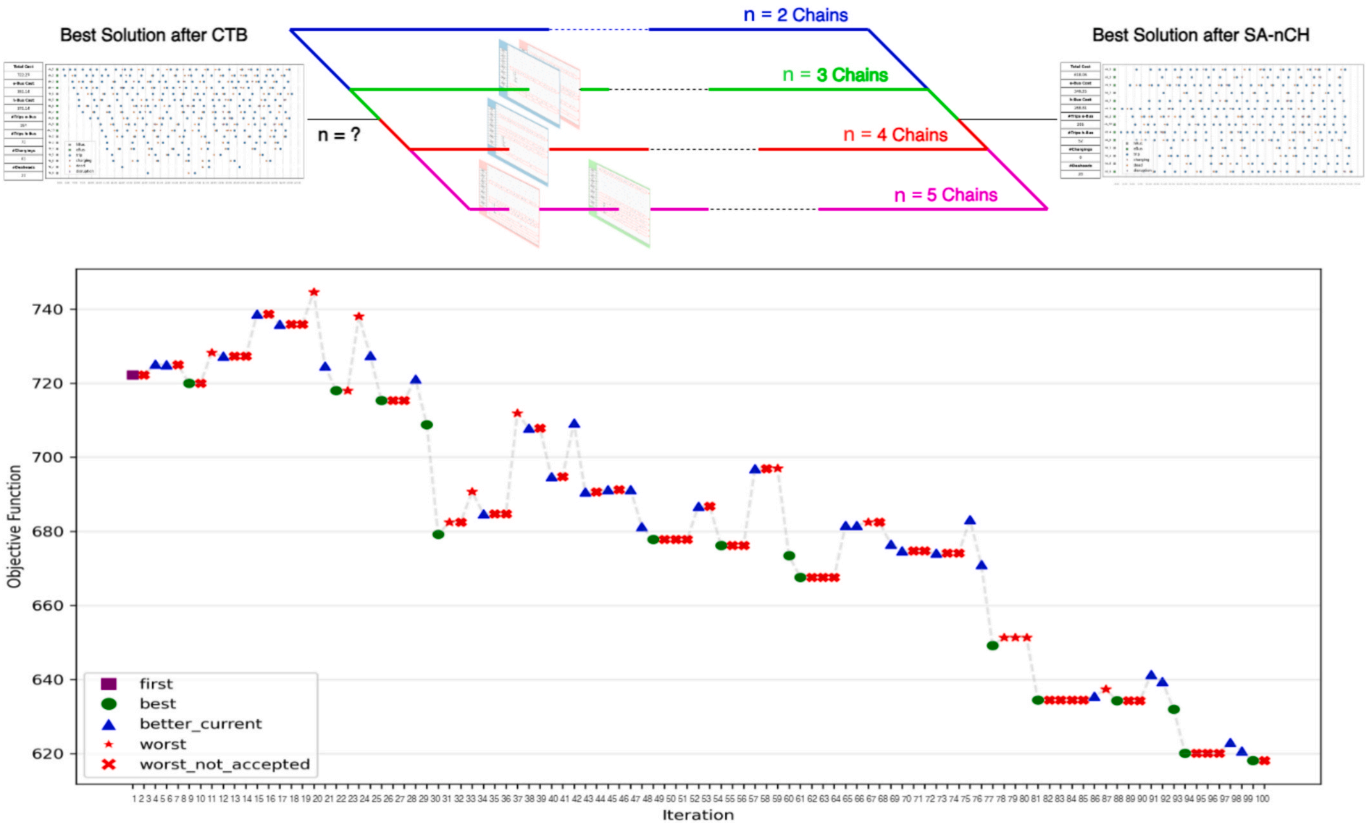


Fig. 10. SA-nCH search properties. The SA-nCH selects n chains from a trip-proportional range for each iteration.

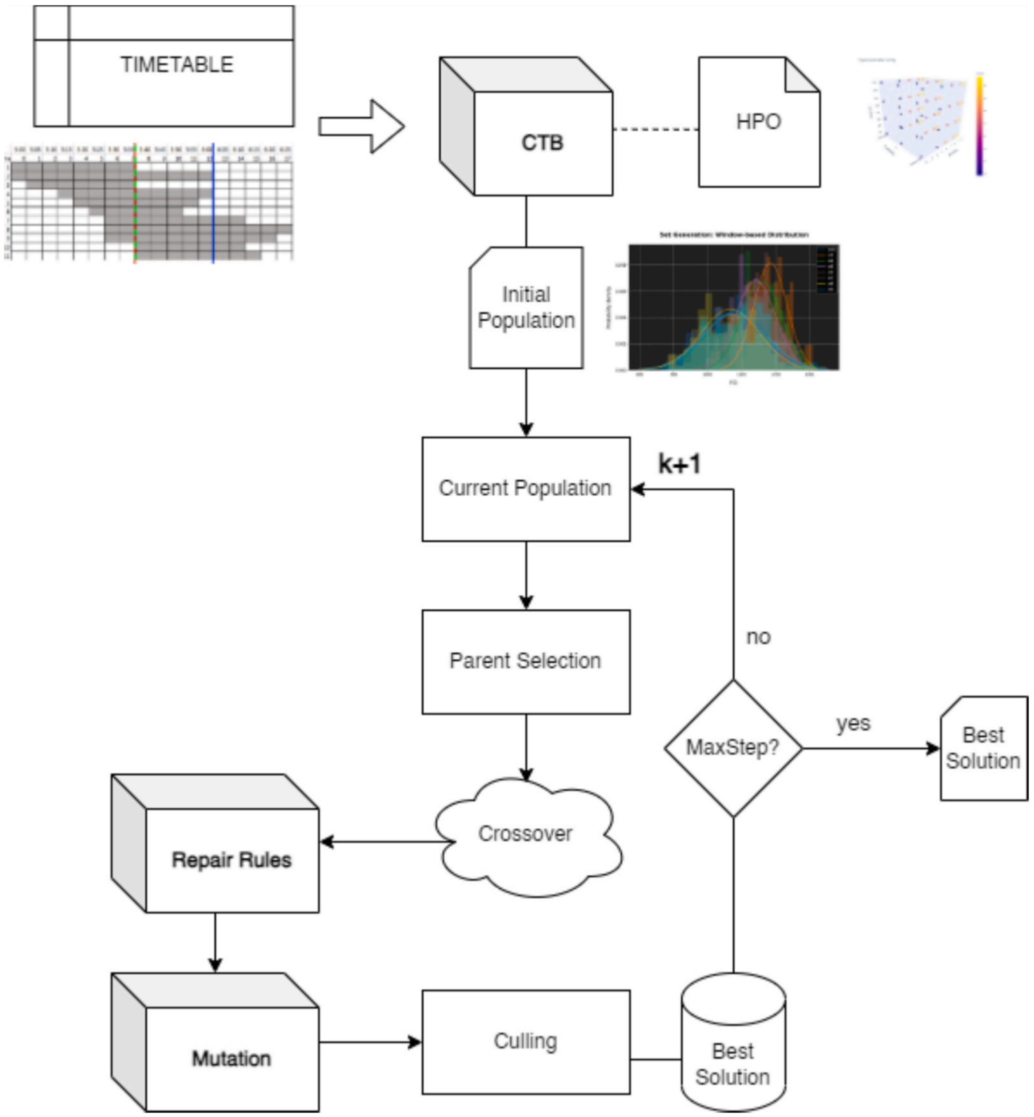


Fig. 11. Flowchart of the Genetic Algorithm.

and *B* (terminals), as previously discussed in Section 3.1. Similarly, Fig. 10 illustrates the search properties of the SA-nCH algorithm. Here, the number of chains n is randomly drawn from a range proportional to the total number of trips (e.g., 2 to 5 for 257 trips). This range allows for more diverse rescheduling opportunities. Because CTB is computationally lightweight compared to MILP, SA-nCH can handle larger sets of chains per iteration. Furthermore, the variability in n helps avoid stagnation by enabling exploration of different regions of the solution space. The CTB heuristic provides quick, feasible solutions, resulting in more dynamic search behavior characterized by sudden improvements in solution quality. In contrast, the gradual modulation of the range introduces the possibility of exploring new solution regions. This results in a more dynamic behavior marked by sudden, sharp improvements in the best solution found.

4.4. *Genetic algorithm and repair rules*

In this section, our focus turns to yet another metaheuristic — a customized Genetic Algorithm (Reeves, 1997). As with our approach to Simulated Annealing, the primary objective is to design a Genetic Algorithm capable of effectively dealing with scalability issues arising from urban-sized instances of the MFMTEBSP. To achieve this, GAs employ a highly modular composition, showcased in Fig. 11. The choice of a GA aligns with dual motivations, mirroring those for selecting SA. Firstly, GAs stand as one of the most frequently employed algorithms in prior literature, substantiated by various studies, including those by

Table 5
Genome representation example.

Trip #	Assigned bus ID	Recharging?	Encoded gene
1	1	N	2
2	2	N	4
3	1001	N (hybrid)	2002
4	1	N	2
5	2	Y	3

Table 6
Key hyper-parameters of the Chain–Trip Builder.

Parameter	**Symbol**	**Search range/rule**	**Value used**
Window size	w	$2-10$	7
Max Iterations	k	$\{50, 100, 150, 200\}$	100
Bus-type selection	p_1	$Uniform, V1, U+V1$	$U+V1$
Switch threshold	δ	$\{0.5, 0.6, 0.7\}$	0.7
Trip-choice weight	p_2, i	$Eq.(37-38)$	$Eq.(37-38)$

Table 7
Key hyper-parameters of the customized Simulated Annealing variants.

Parameter	Symbol	SA-2CM	SA-nCH
Chains per perturbation	w	2	$\propto\lvert J\rvert$
Rescheduling engine	–	$MILP(CPLEX)$	CTB
Cooling schedule	–	$Logarithmic, Eq.(35)$	$Logarithmic, Eq.(35)$
Cooling factor	α	$adaptive, Eq.(36)$	$adaptive, Eq.(36)$
Initial temperature	T_0	100	100
Max stagnation	S_{max}	20	20
Max steps	N_{max}	100	100

Table 8
Key hyper-parameters of the customised Genetic Algorithm.

Parameter	Symbol	Value/range
Population size	Pop_{max}	500
CTB window for pop. gen.	w	$\{0.7, 0.3, 0.1\} \times Pop_{max}$
Desiderable fleet size	Z	$\propto\lvert I\rvert + \lvert H\rvert$
Elitism rate	ρ_e	0.05
Parenthood probability	ρ_p	0.9
Mutation probability	ρ_m	0.1
Max generations	G_{max}	300
Stagnation limit	φ	30
Repair window width	a	$\lceil\sqrt{\lvert J\rvert}\rceil$
Crossover cut point	χ	$U(1, \lvert J\rvert)$

(Chao and Xiaohong, 2013; Rogge et al., 2018; Wang et al., 2021) and others. Secondly, GAs represent a population-based algorithm, offering a contrast to single-solution approaches, which is particularly relevant for performance comparisons. Additionally, GAs employ an integrated crossover operator, which often generates infeasible solutions in its pursuit of escaping local minima. Consequently, the application of repair rules becomes necessary.

In what follows we motivate and detail how each component of the GA approach has been customised in order to correctly characterise the problem at hand:

- **Genome representation.** A potential bus schedule is represented by a set $\Gamma = \left\{\gamma_1, .., \gamma_j\right\} \in \mathbb{Z}^{1\times|J|}$, where each member γ_j captures which e-bus or h-bus has performed trip $j \in J$. The genome elements are next characterised:

$$\gamma_j = \begin{cases} i\cdot 2 \forall j : {y^t}_{i,j} = 1, t = d_j \\ (i\cdot 2) - 1 \forall j : {y^t}_{i,j} = 1, t = d_j, i : \left(\varepsilon_i^{t+1} - u_j\right) < \mu\cdot E \ \#(39) \\ (1000 + h)\cdot 2 \forall j : z^t_{h,j} = 1 \end{cases}$$

We explore three potential genetic encodings for each GA solution: when an e-bus i can properly perform a trip j within its charging capabilities, the encoded result is $i\cdot 2$. However, if the e-bus is expected to need recharging after completing the service trip, the encoded result is $(i\cdot 2) - 1$. Trips executed by hybrid buses follow a similar representation but are offset by a label value of 1000. For clarity, we provide an illustrative schedule encoding example in Table 5.

It is important to remark that this (Tables 6-8) representation does not consider any possible departure time delay, hence assuming $\theta = 0$. As shown in Eq. (39), a genetic encoding featuring an odd e-bus IDentifier (ID) implies that bus $i = \left\lfloor \gamma_j/2 \right\rfloor$ must be recharged after performing trip j. However, this representation does not precisely dictate the time t at which this charging event must occur, rather imposing a softer constraint in the form $\sum_{t=d_j+t_j-1}^{T} x^t_{i,b,m} > 0, \forall b, m$, stating that at least a charging event must occur for bus i after the completion time of trip j. The specific moment of time in which this charging event occurs is implicitly determined by the next trip $\bar{j}$ performed by bus i, if any, as the problem's fitness function, detailed later, imposes that sufficient energy must be available for its execution, thereby generating an additional soft constraint in the following form:

$$\sum_{t=d_j+t_j-1}^{t=d_{\bar{j}}} x^t_{i,b,m} > 0, \forall b, m, d_{\bar{j}} > d_j + t_j - 1 \#(40)$$

- **Population generation.** To start the evolutionary process, an initial set of feasible individuals $pop = \left\{\Gamma_p\right\}$ must be generated, until reaching a specified maximum target population size. Each individual, representing a feasible bus schedule, is generated using the CTB heuristic. The heuristic terminates once the maximum population value is achieved. A well-generated population should balance both high-quality solutions and diversity among individuals. To ensure these two characteristics, preliminary experiments have been conducted on the CTB pool, varying the window size $|W|$. Results for the scenario involving 1084 trips are depicted in Fig. 12. To each window size considered, corresponds a set of individuals generated and consequently a gaussian distribution. Each window size corresponds to a set of generated individuals and forms a Gaussian distribution. The results indicate that wider windows lead to flatter bell-shaped distributions (higher variance σ^2) and lower means μ, which translates to a more desirable population for the GA. Nevertheless, overly

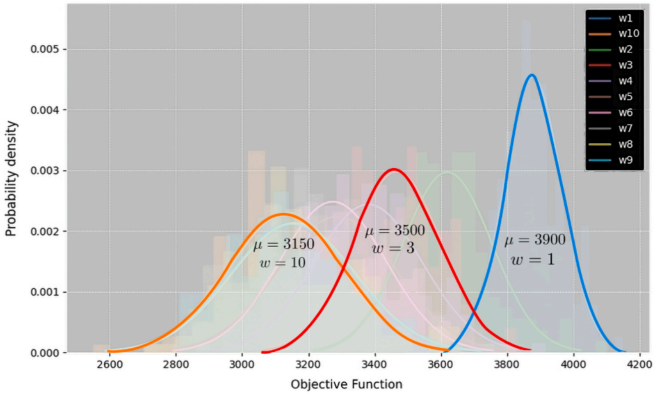


Fig. 12. Gaussian distributions corresponding to various window sizes.

large window sizes increase the probability of failures (blocks not generated) and subsequently prolong the computational time required to reach the maximum population size. To address this limitation and further enhance diversity, we explore three window sizes, corresponding to 70 %, 20 %, and 10 % of the population size, during the CTB execution;

- **Fitness function.** To effectively rank the population, we employ a fitness function that needs to be minimized, taking the following form:

$$\phi\left(\Gamma_p\right) = \begin{cases} +\infty, \exists(i,h,t) : \sum_j \left(y_{i,j}^t + z_{h,j}^t + \sum_{b_o, b_d \in B} \omega_{i,b_o,b_d}^t \right) > 1 \\ +\infty, \left(\exists(i,j) : g_{i,b}^{d_j} \neq \alpha_j \wedge \nexists \bar{t} < d_j : \omega_{i,b,\alpha_j}^{\bar{t}} = 1 \right) \vee \left(\exists(h,j) : p_{h,b}^{d_j} \neq \alpha_j \right) \\ +\infty, \exists(i,j) : \varepsilon_i^{d_j} < \left(u_j + \mu \cdot E \right) \\ (42), \textit{otherwise} \end{cases} \#(41)$$

that is, the fitness of a given individual Γ_p is set equal to infinity (indicating infeasibility) if (i) scheduling conflicts arise, such that an individual e-bus or h-bus is assigned to multiple concurrent trips, either service or deadheading; (ii) location conflicts arise such that an individual e-bus or h-bus is assigned to a trip without being located in the appropriate terminal and no feasible deadheading trip can be performed; (iii) charging conflicts arise due to residual charge being insufficient to perform an assigned trip *j*. Otherwise, the following penalised form of objective function (1) is employed:

$$v \bullet \left(\sum_t \sum_i \sum_j c \bullet y_{i,j}^t + \sum_t \sum_h \sum_j \widehat{c} \bullet z_{h,j}^t + \sum_t \sum_i \sum_{b_o} \sum_{b_d} \overline{c} \bullet \omega_{i,b_o,b_d}^t \right.$$
$$\left. + \sum_t \sum_i \sum_b \sum_m q^t \bullet x_{i,b,m}^t \right) \#(42)$$

$$v = \begin{cases} \left(\frac{(|I| + |H|\,)}{Z} \right), (|I| + |H|\,) > Z \\ 1, \textit{otherwise} \end{cases} \#(43)$$

where *Z* represents an exogenous parameter signifying the desirable fleet size limit. This penalization term serves the purpose of penalizing the GA approach when it resorts to solutions with excessively large fleet sizes. However, it also allows such solutions to remain within the realm of feasibility, even if they are poor performers. This approach ensures the preservation of substantial genetic variability;

- **Parent selection and Crossover operator.** When it comes to parent selection, the GA operates by randomly selecting two individuals from the population. This process is designed to guarantee each individual in the population an equal chance to contribute to the creation of a new offspring solution. Once these two individuals are chosen, they are removed from further consideration to prevent their repeated selection in subsequent iterations. Given a couple of parent individuals $\left(\Gamma_m, \Gamma_f\right)$, a cut point χ is determined by performing a random draw from the uniform distribution $U(1, |J|)$. The genome of the two parent individuals is then combined as follows: $\Gamma_c = \left\{ \left\{ \gamma_1, \gamma_2, \cdots \gamma_\chi \right\} \in \Gamma_m, \left\{ \gamma_{\chi+1}, \gamma_{\chi+2}, \cdots, \gamma_j \right\} \in \Gamma_p \right\} \#(44)$

that is, the two genomes are spliced such that the first individual's schedule is employed up to and including the cut point, while the second individual's schedule is employed thereafter. This direct splicing might cause conflicts in the resulting bus schedule, hence requiring genome repair;

- **Genome repair operator.** As the earlier introduced crossover operator may produce infeasible individuals ($\phi(\Gamma_c) = \infty$), we have devised a genome repair operator. This operator employs a series of four sequential rules to rectify the genome of the individual, aiming to restore its feasibility. The application of these rules proceeds sequentially and stops as soon as the individual regains feasibility:

1. **Rule i.** Given the cut point χ associated with the crossover operator, a subset of all trips composing the individual $\left[\gamma_{\chi-a}, \gamma_{\chi+a}\right]$ is selected, where *a* is a customisable window width parameter. These bus/trip assignments are then randomly permutated a maximum of $\sqrt{|J|}$ times. If any of these permutations yields feasibility to individual Γ_c the procedure is halted, and the individual is considered repaired. Otherwise, the original ordering is restored, and the following rule is applied;
2. **Rule ii.** All conflicting trips $j \in \Gamma_p$ are identified. For each identified conflict, a h-bus is sought such that $\{\overline{h}\} : \sum_j \sum_{t=d_j}^{d_j+t_j-1} z_{\overline{h},j}^t = 0$ (the given h-bus is available to perform trip *j*), and the offending bus/trip assignment is replaced. If the available h-bus set becomes empty at any point in the sequence of conflicting trips, the rule fails, and the partly repaired individual $\Gamma_{\overline{p}}$ proceeds to the next rule. Otherwise, the fitness of the repaired individual is evaluated, to ensure that no charging inconsistencies arise, and (if successful) the individual is considered repaired;
3. **Rule iii.** All conflicting trips $j \in \Gamma_{\overline{p}}$ are identified. The electric bus fleet is expanded by one additional e-bus, which is then assigned to all possible potential conflicts by using the events generation function from the CTB. If no further potential conflicts remain, the individual is considered repaired, otherwise, the next rule is applied;
4. **Rule iv.** In the same vein as the previous case, we first identify all conflicting trips $j \in \Gamma_{\overline{p}}$. Subsequently, we expand the conventional bus fleet by one additional h-bus, which is then allocated to address all potential conflicts. Similar to our previous approach, we utilize the events generation function for h-buses from the CTB. If, after this adjustment, no further potential conflicts persist, we consider the individual as successfully repaired; otherwise, we proceed to apply rule iii.

The order of rule execution reflects the prioritization of potential solutions. Rule i. is based upon the principle that rearranging a relatively small window of trips around a cut point can suffice to ensure that decisions made before the cut point do not adversely affect those made afterward; its design is based on the validation results of the previous proposed time-based decomposition scheme (Rinaldi et al., 2020), wherein comparisons between solutions identified by exact and decomposed approaches suggested that the overlap of decisions along the point of decomposition are a strong source of sub-optimality. Rule ii. addresses conflict situations stemming from e-buses by substituting problematic trips with h-buses. While this remedy alleviates conflicts, it often results in a less favorable solution compared to maximizing the utilization of the electric fleet, making it a lower-ranked solution. Moreover, rules iii. and iv. are applied consecutively until an individual's genome is ultimately repaired. As detailed earlier, this practice, however, incurs considerable penalisation in the fitness operator, as they increase the total fleet size potentially beyond the desired maximum;

- **Mutation operator.** Following standard GA practices, a given percentage of individuals in each subsequent generation undergoes mutation. Our proposed mutation operator identifies, for a given individual Γ_c, the hybrid bus h that performs the least number of trips, $h^* = argmin_h \sum_t \sum_j z^t_{h,j}$. Once all elements $\gamma_j \in \Gamma_p : \left\lfloor \frac{\gamma_j}{2} \right\rfloor = h^*$ are identified, a single trip j^* is randomly drawn therefrom. After determining whether an e-bus i^* that could feasibly perform trip j^* is available, if any, the bus/trip assignment is replaced such that $\gamma_{j^*} = 2{\cdot}i^*$. If no such e-bus is available, the procedure is repeated by considering h-buses other than h^*. If no such h-bus exists, the mutation does not occur. This logic mimics the properties of exact solutions found through CPLEX, wherein the percentage of time during which e-buses are not idling is maximised. This mutation operator eventually leads to the exclusion of low-performing h-buses from the overall bus fleet;

Following a standard Genetic Algorithm, some hyperparameter must be chosen including a maximum population size Pop_{max}, desirable fleet size Z, an elitism threshold ρ_e, a parenthood probability ρ_p, a mutation probability ρ_m, a maximum number of generations G_{max}, a maximum stagnation parameter φ. The proposed algorithm operates on consecutive generations. At each generation, an elitism parameter determines the pool of potential parent individuals to be selected for breeding. For each non-repeating couple of parent individuals, the crossover function is applied in order to generate a (potentially infeasible) child individual Γ_c, which is then repaired through the genome repair operator, if necessary, and undergoes mutation with the given probability. Before merging the newly generated individuals into the overall population, a stagnation check is performed, by comparing the best-ranking individual in the extant population and in the child set. If no improvement in fitness is found, a counter recording the amount of successive stagnation is increased, whereas if generation n provides an improvement, the counter is reset. The newly generated individuals are merged into the overall population, which is then sorted by fitness. The maximum population limit is upheld by culling individuals with lower rankings once the specified limit is reached.

5. Experimental results

This section will be structured as follows: Section 4.1 will discuss the case study of Luxembourg City's bus network, including the experimental setup and the configuration of key hyperparameters for the evaluated metaheuristic algorithms. Section 4.2 will introduce the MILP comparison, with a specific focus on micro instances, aimed at evaluating the quality gap of the approximate approaches. In Section 4.3, we will analyze the performance of various metaheuristics across small, medium, and real-size instances, considering multiple fleet and charging infrastructure scenarios to determine the best approach.

5.1. Case study and synthetic data generation

To evaluate the performance of the optimization model and the proposed algorithms, we introduce two Synthetic Data Generators (SDGs) as detailed in Section I.1 of the Supplementary Materials. This step is necessary for two primary reasons: The MILP model has limitations when it comes to handling real-world scale instances. Consequently, we need a tool capable of generating consistent micro instances that can be effectively processed by the model. Even when real-world datasets are available from operators, a thorough assessment of metaheuristic performance requires consideration of multiple fleet compositions and charging infrastructure scenarios. This approach not only enhances the robustness of the assessment but also helps prevent data overfitting. Thus, the creation of a real-world instance generator becomes essential to explore various setting scenarios.

Nevertheless, both the SDGs developed are based on datasets from the public transport network of the city of Luxembourg, located in the eponymous country. We extract regular weekday trip data for a total of 10 lines, handled by the Autobus de la Ville de Luxembourg (AVL) operator, from the openly available GTFS dataset provided within the nation's open data portal.[1] Specifically, we consider lines 1 (Bouillon – Kirchberg), 16 (Gare Centrale – Aeroport), 9 and 14 (Gare Centrale – Cents), 10 (Gare Centrale – Steinsel), 12 (Bouillon – Dommeldange), 13 (Gare Centrale – Centre Hospitalier), 17 (Bouillon – Monterey), 27 (Gare Centrale – Bertrange Belle Etoile) and 28 (Gare Centrale – Bertange Ecole Europeenne). This yields a total problem size of up to 10 lines, 12 terminals, and 1084 daily trips to be scheduled. Unless otherwise specified, we consider charging stations to be available in four terminals (Gare Centrale, Bouillon, Cents, and Bertrange B.E.). A map highlighting the chosen lines in the city's PT network is shown in Fig. 13.

Regardless of the SDG applied, we adopt problem parameters such as the two energy components η_1, η_2, battery pack sizing, energy cost q^t_i following the previous research contributions of our team. We therefore consider a consumption rate of 1.7 kWh/km for e-buses, assuming a battery pack capacity of 100 kWh, while for h-buses we deal with an adjusted consumption rate of 2.3 kWh/km, assuming that these buses are in fact part of the largely adopted plug-in hybrid fleet available to the city's operator, provided by Volvo Bus corporation. In order to monetise these values, we multiply both by a generalised cost of 0.23 EUR/kWh, which includes an estimation of direct and indirect costs in trip operations, beyond the direct consumption components. For the sake of recharging, we consider energy cost to be set at 0.15 EUR/kWh. The implementation of all models and algorithms is conducted in Python 3.9 using the PyCharm IDE. Our computational experiments are executed on a system powered by an Intel Core i7-1260P processor with 32 GB of RAM.

5.1.1. Key hyperparameter settings

This subsection presents a detailed summary of the key hyperparameter configurations adopted for the metaheuristic algorithms evaluated in this study. For each algorithm, the relevant parameter names, symbols, and either their fixed values or applicable search ranges are provided in tabular format. These configurations were determined based on a combination of preliminary tuning experiments and established practices found in the literature.

Unless otherwise specified, all hyperparameter values were held constant throughout the experimental analysis.

5.2. MILP comparison on micro instances

In Section I.2 of the Supplementary Materials, we subject the MILP model to a rigorous stress test, serving as a preliminary phase for the

[1] https://data.public.lu.

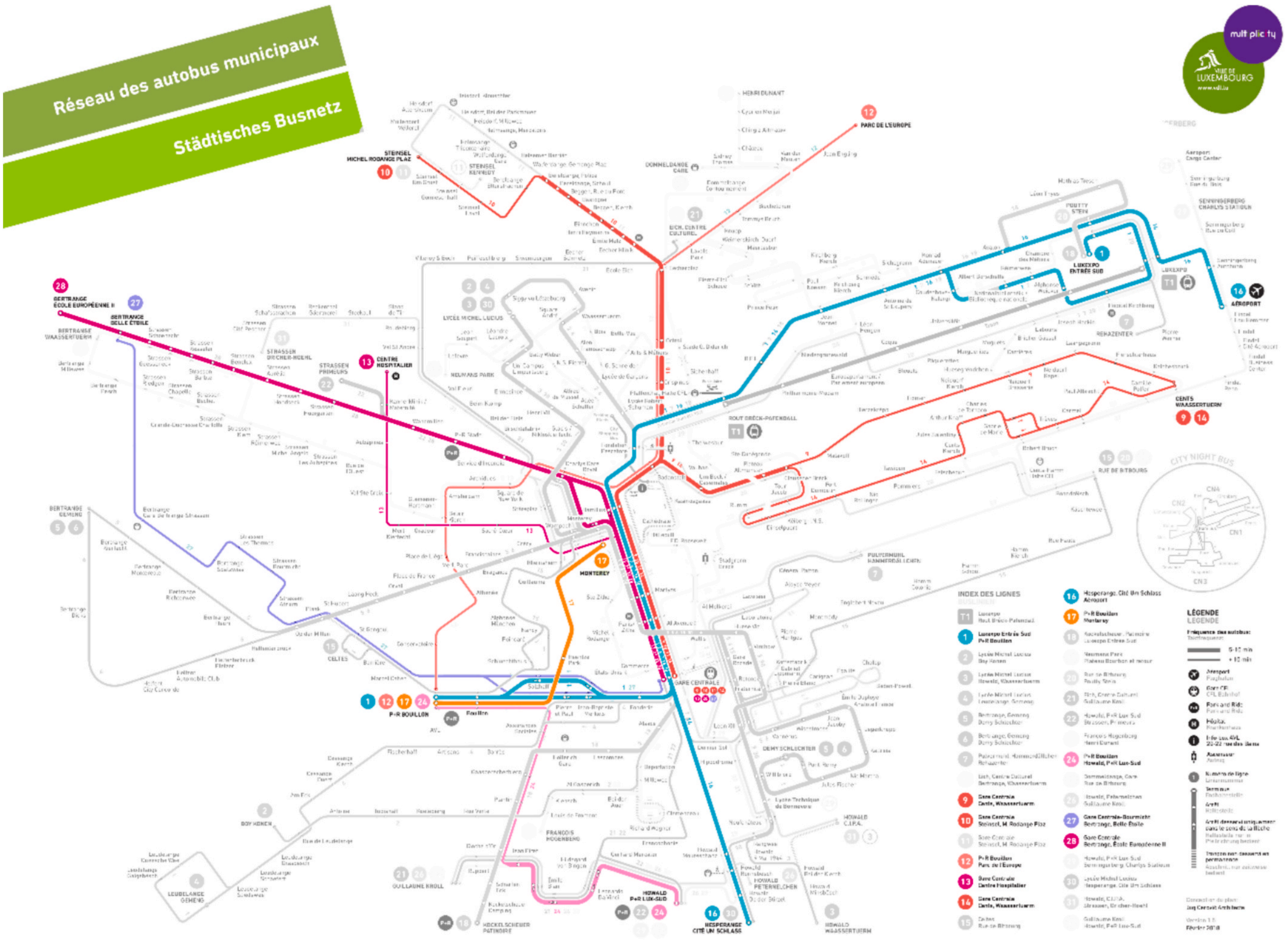


Fig. 13. Geographical distribution and shape of selected bus lines in Luxembourg City.

identification of critical instance attributes to be configured for the MIP comparison. Building upon these findings, we have defined a set of micro instances to assess the solution quality gap for each algorithm. These experiments encompass instances featuring a number of trips ranging from 40 to 60, with an available fleet size of 14 buses and a battery pack capacity of 60 kWh. For each tool, we have selected the best solution among ten runs. Detailed data can be found in Table 16 of the Supplementary Materials. The outcomes reveal the following insights:

- **MILP model.** In the MILP model runs, we have set a time limit of 1 h and a tolerance of 5 %. However, it is worth noting that in certain cases, like the instances with 48 and 60 trips, the optimum gap from the lower bound (LB) reaches as high as 7 %. The processing times, along with other data such as the number of simplex iterations, variables, and constraints, underscore the significant complexity of the problem, even for instances of this size, making it challenging to find a solution within a reasonable time frame;
- **SA-2CM.** Among the approximate approaches, the SA-2CM stands out with the best average gap from the MILP model solutions. It even manages to achieve a superior optimum gap from the LB for instances with 54 and 60 trips (Fig. 14). The processing times required are significantly lower than those of the MILP model but are of a second-order magnitude when compared to SA-nCH, CTB, and GA (as depicted in Fig. 15). Notable characteristics include the higher CPU memory requirement, as it leverages up to 6 cores for parallel executions, and a consistent descent due to the reduced number of chains compared to SA-nCH, which are rescheduled at each iteration. In general, the SA-2CM exhibits a trend of objective function values that closely aligns with the MILP model, as it optimally schedules deadheading and charging events for each pair of chains. However, as we will observe in the context of real-world scenarios with a higher number of buses involved, the solution quality begins to decline;
- **SA-nCH.** SA-nCH emerges as the second best tool in terms of objective function value, trailing behind the other SA version. However, it takes the lead in terms of processing times, completing all the computations within a minute. Given the limited size of the available bus fleet, the number of buses employed matches that of the MILP model and SA-2CM. This implies that these tools primarily focus on optimizing the allocation of trips to the remaining buses and fine-tuning the scheduling of charging and deadheading events. An interesting behaviour, which will also be discussed in the context of real-world scenarios, is that increasing the number of chains rescheduled tends to reduce the likelihood of getting stuck in a local minimum. However, it is crucial to strike a balance with the acceptance probability, as this could potentially lead to significantly inferior solutions as well;
- **CTB.** This algorithm stands out as the fastest performer, boasting an average processing time of just 4.57 s − a characteristic that will be further confirmed in our analysis of real-world scenarios. However, its performance seems slightly less impressive compared to the larger scenarios. This observation suggests a potential misalignment between its constructive approach and the intricacies inherent to micro-level scenarios;

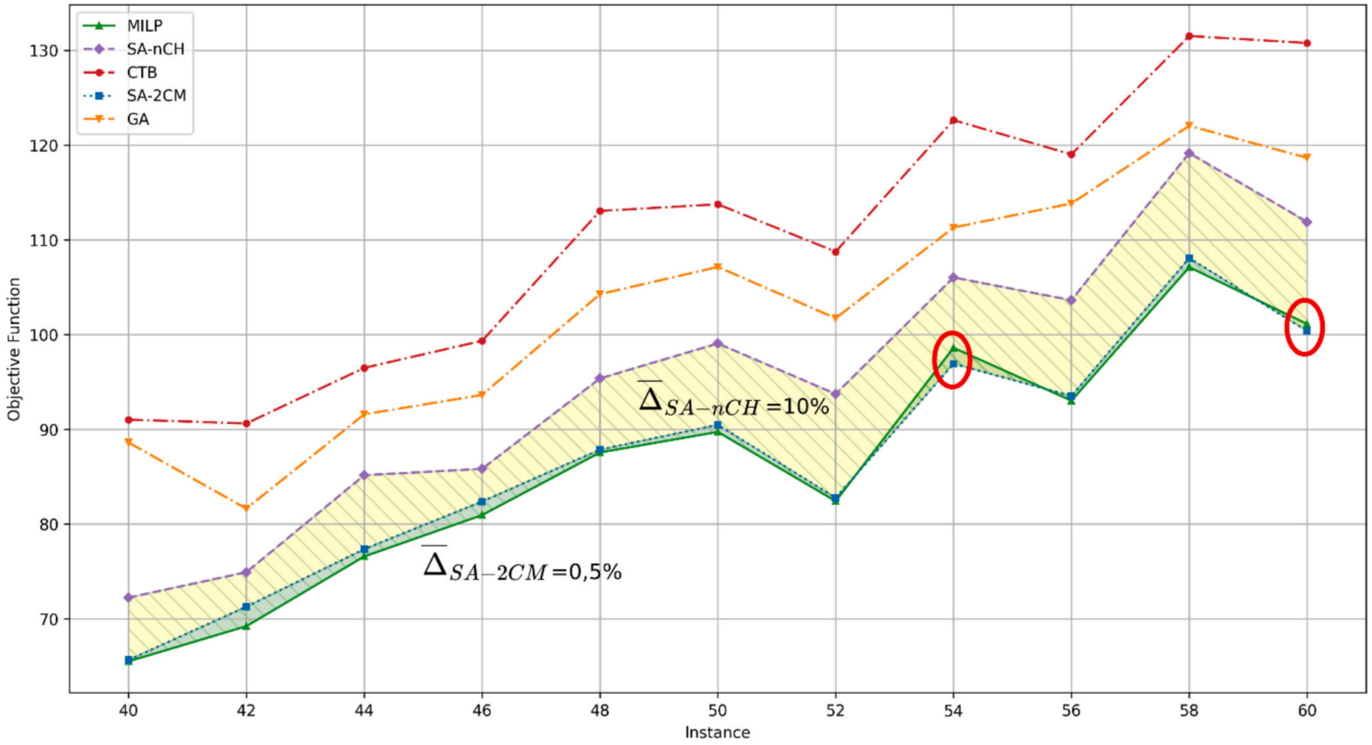


Fig. 14. Trends in objective function values for the MILP model and each algorithm. The SA-2CM exhibits a nearly overlapping trend, achieving superior solutions (red circles) compared to the MILP model for instances with 54 and 60 trips.

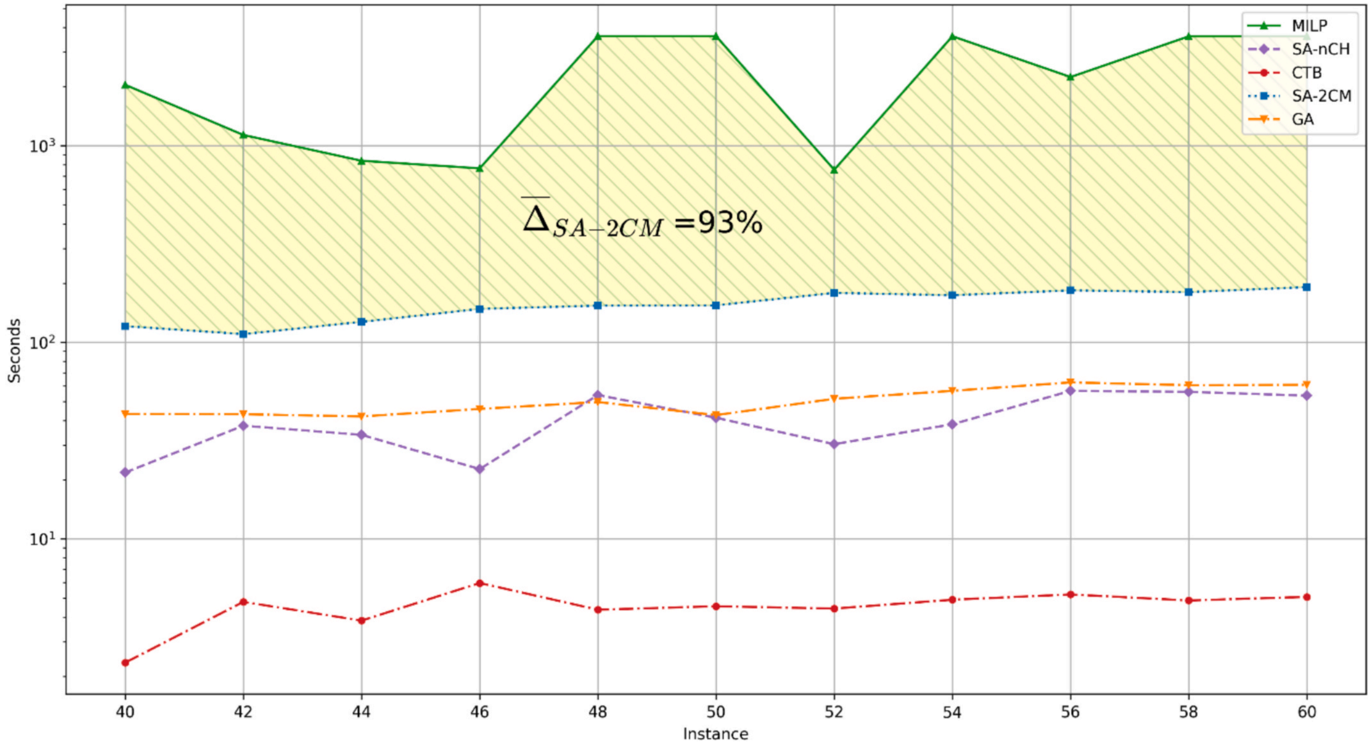


Fig. 15. Trends in processing times for the MILP model and each algorithm. The MILP model exhibits processing times on a vastly different scale compared to the SA-2CM and the other algorithms.

- **GA.** As we will observe in the forthcoming real-world scenarios, the GA falls between the CTB and the SA-nCH in terms of both objective function values and processing time performance. When assessing the trends in objective values, it exhibits an irregular descent pattern (Fig. 16), characterized by abrupt improvements in the objective function scores following a certain number of population crossovers (Fig. 17)

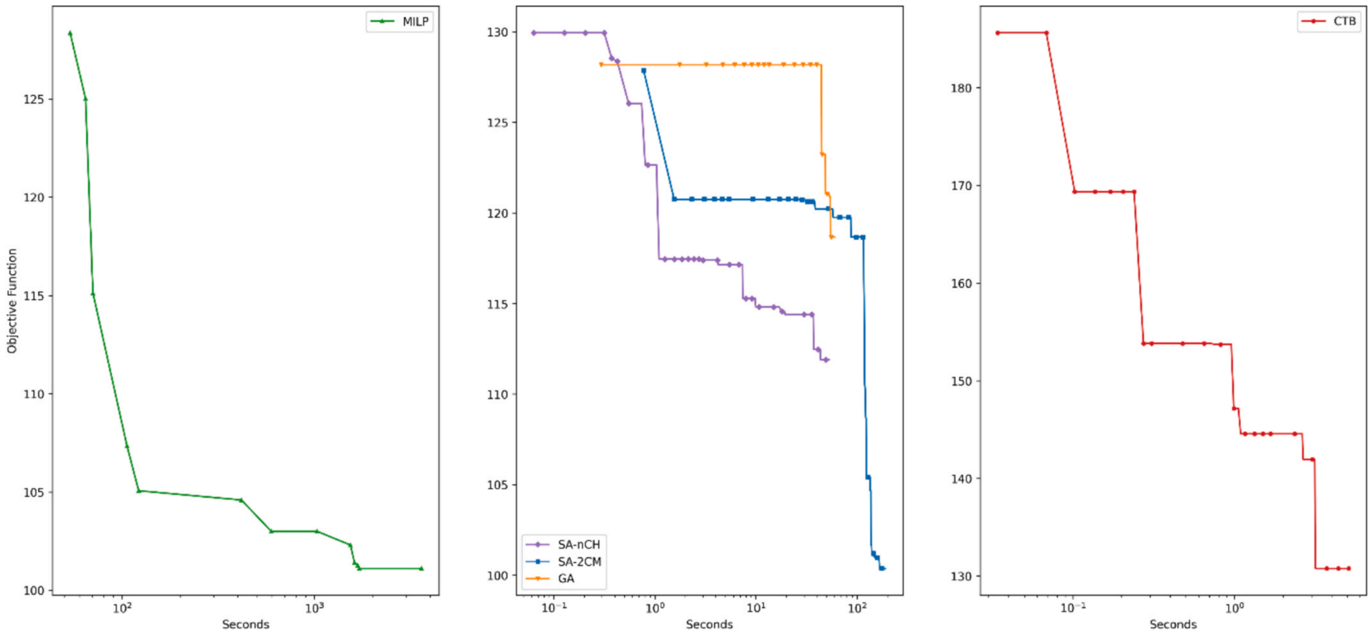


Fig. 16. Trends in objective function values over time for the MILP model and each algorithm.

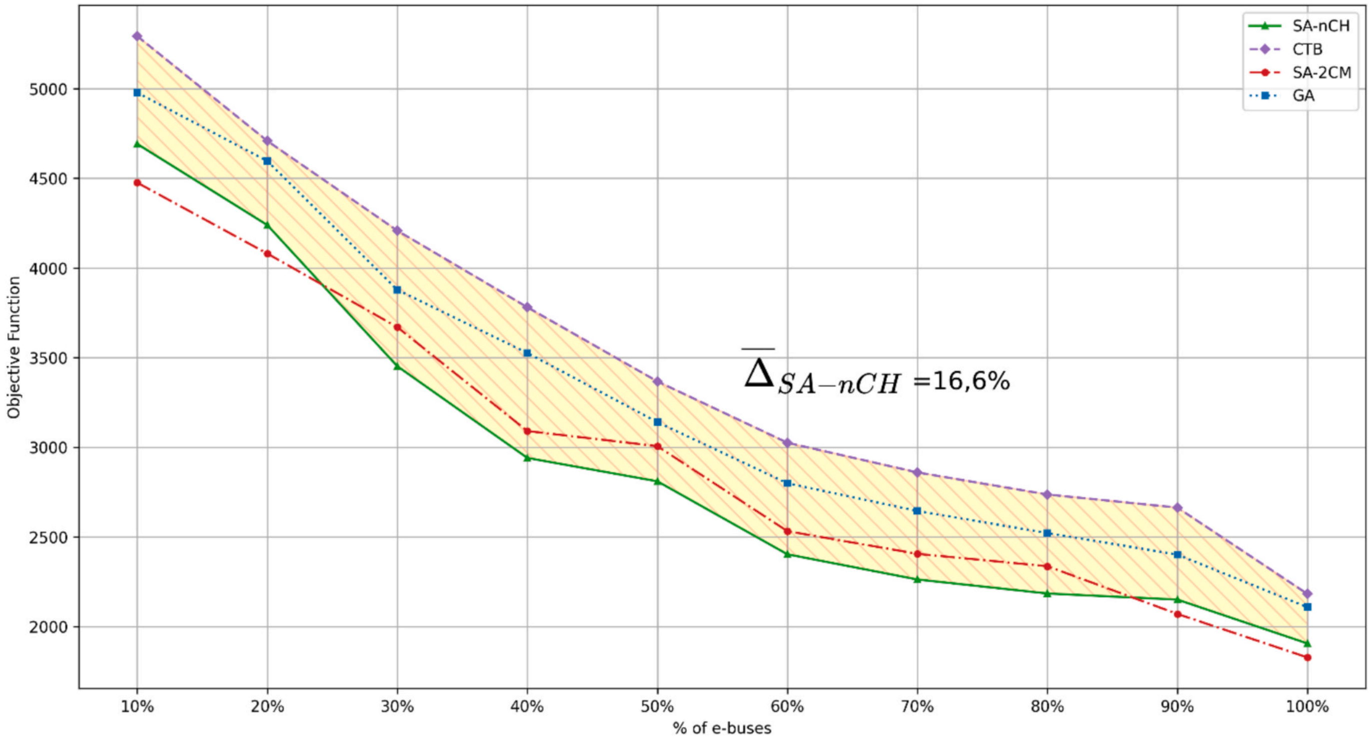


Fig. 17. Objective function score trends for the 1084 trip scenario with an incremental percentage of e-buses in the fleet.

Table 9
Parameter settings for each experiment.

Setting 1	$\lvert J \rvert$	$\lvert I \rvert + \lvert H \rvert$	$\lvert B \rvert$	$\lvert \overline{B} \rvert$	E(kWh)	% e-bus
	257	20	5	2	100	Variable
	628	40	5	2	100	Variable
	1084	60	12	4	100	Variable
Setting 2	1084	80	12	Variable	100	50 %

Table 10
Statistical comparison of metaheuristic performance across real-world instances.

Instance size	Avg. rank SA-2CM	Avg. rank SA-nCH	Avg. rank GA	Avg. rank CTB
257 trips	1.00	2.00	3.00	4.00
628 trips	1.6	1.4	3.10	3.90
1084 trips	1.6	1.4	3.00	4.00

Statistical Validation. To assess whether the performance differences among the four metaheuristics on the micro instances are statistically significant, we applied the non-parametric Friedman test. This test ranks the algorithms within each instance and evaluates whether the resulting rank distributions differ. The test was applied to the objective-function values reported in Table 16 of the Supplementary Materials (k=4 algorithms, N=11 instances), yielding $\chi^2(3)$=33.0, p<0.0001. Thus, the null hypothesis that all algorithms perform equivalently is decisively rejected. To determine which algorithms differ, we conducted a post-hoc Nemenyi multiple-comparison test. In this test, two algorithms are significantly different if the absolute difference in their average ranks exceeds the critical difference (CD), which in our case is 1.48. The average ranks were: SA-2CM = 1.05, SA-nCH = 2.18, GA = 3.00, and CTB = 3.77. The largest difference (SA-2CM vs CTB) is 2.72, which exceeds the CD. All other relevant pairwise differences also exceed this threshold. Therefore, the post-hoc analysis confirms that SA-2CM outperforms all other heuristics, SA-nCH outperforms GA and CTB, and GA outperforms CTB. These results support the performance trends discussed earlier.

Up to this point, a key takeaway becomes evident. While employing MILP for real-sized instances might be impractical due to the extensive computation times, SA-2CM emerges as a highly viable alternative for offline runs, particularly if the processor has the capability to harness multiple cores for parallel optimization. Concurrently, SA-nCH and GA present a favorable balance between solution quality and processing time. Furthermore, even CTB, when situated within real-time scenarios involving short-term fluctuations in exogenous variables like energy costs, could prove to be a valuable tool, thanks to its efficiency.

5.3. *Metaheuristic comparison on real-world instances*

In our analysis of real-world instances, we have considered three distinct instance sizes, each varying in the number of trips, terminals, and terminals equipped with charging stations. We conducted two separate analyses to explore different aspects of our study. The first analysis involved a progressive increase of 10 % in the proportion of e-buses within the bus fleet. This allowed us to examine the performance behaviors of our tools and understand how this percentage affects operational costs. In the second analysis, we focused on increasing the percentage of terminals equipped with charging stations to assess its influence on deadheading and charging activity. However, this analysis was limited to urban-sized instances of 1084 trips. This limitation was necessary because with a small set of terminals, increasing the percentage did not result in a significant change in the number of terminals equipped with charging stations. For this part of our study, we utilized the CopulaGauss SDG method, which we introduced in the Supplementary Materials, to generate multiple infrastructure configurations for urban-sized instances. Table 9 provides a comprehensive overview of the exogenous parameters relevant to the experiments we conducted (Table 10)

5.3.1. *Setting 1: Variable percentage of e-buses*

In these experiments, we investigate the impact of an increasing percentage of e-buses within the overall fleet composition. Fig. 18 illustrates the trends in objective function scores considering the instance with 1084 trips for the CTB, SA-nCH, SA-2CM, and the GA as we vary the fleet composition. We collected the top-performing solution from ten runs for each algorithm, which aggregated data for all instance sizes can be found in Table 17 of the Supplementary Materials. Two fundamental behaviors emerge from our analysis:

- When we examine the initial and final phases of each tool's performance, we observe a flattening of marginal gains. In instances where there is a significant imbalance in the fleet composition, the task of optimizing bus type assignments becomes straightforward, essentially reducing it to a scheduling exercise for charging. This occurs because each tool will prioritize the bus type that is in the minority. In contrast, with a more balanced mix of available fleets, each tool has greater room for manoeuvre and the variety of bus types contributes significantly to the overall reduction in operational costs;
- Building on the previous point, we notice a general trend characterized by convexity. There is a notable gap in the quality of solutions

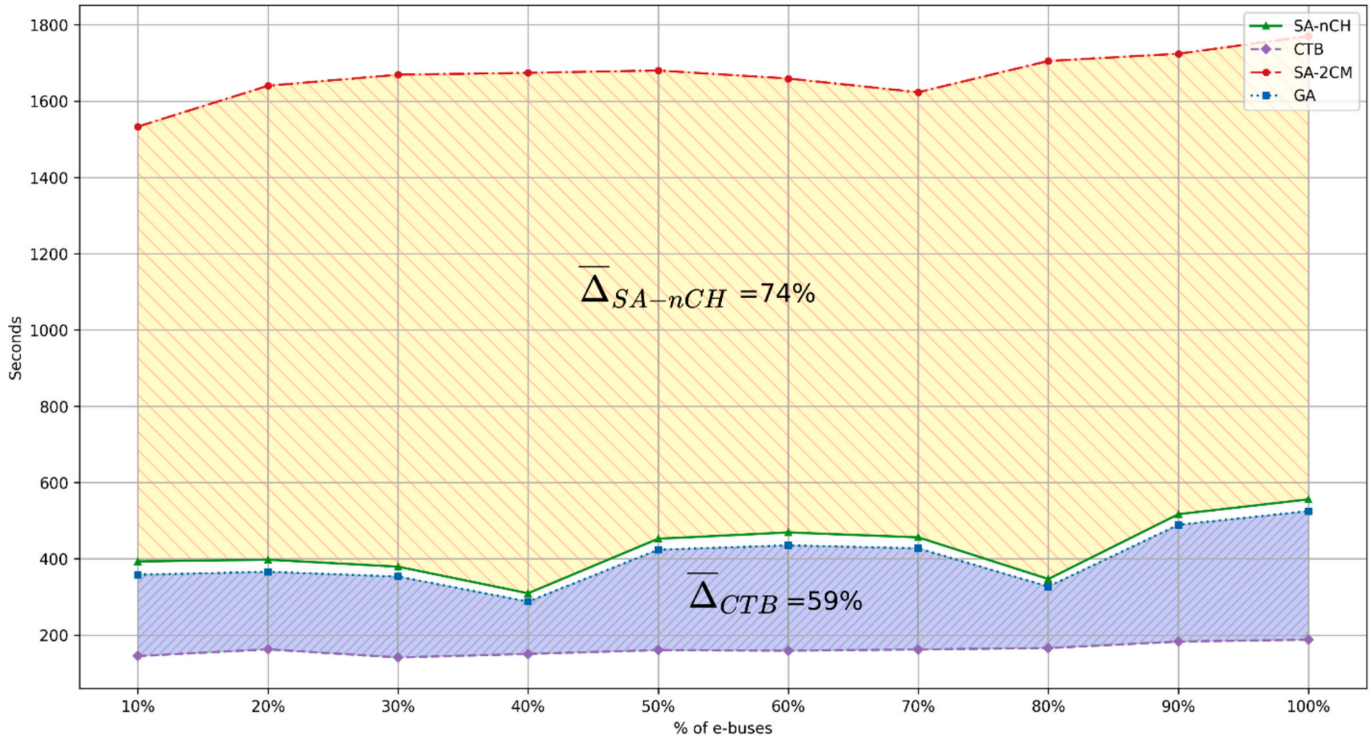


Fig. 18. Computational time trends for the 1084 trip scenario with an incremental percentage of e-buses in the fleet.

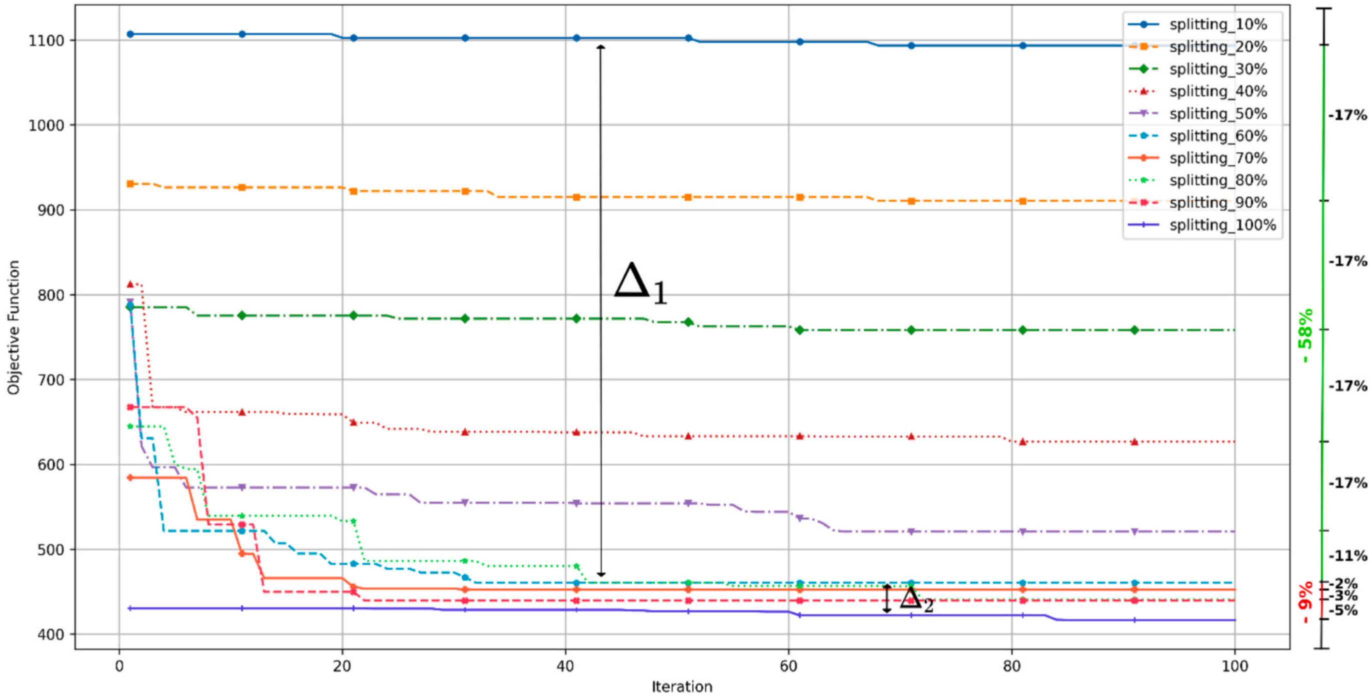


Fig. 19. Objective function score trends and marginality gains for the SA-nCH algorithm in the 257 trip scenarios with an incremental percentage of e-buses in the fleet.

found between the 30 % and 60 %-70 % thresholds of total e-buses in the fleet.

Additional insights can be observed from the performance of each proposed tool:

- **SA-nCH.** SA-nCH stands out as the tool with the most favorable trade-off between computational time and solution quality. On average, it exhibits a solution gap ranging from 14 % to 16.6 % when compared to CTB, and a 10 % solution gap when compared to GA. SA-nCH excels in finding optimal solutions within the 30 %-70 % range of e-bus proportions. Furthermore, it minimizes fleet size usage, a factor that should be taken into account when evaluating

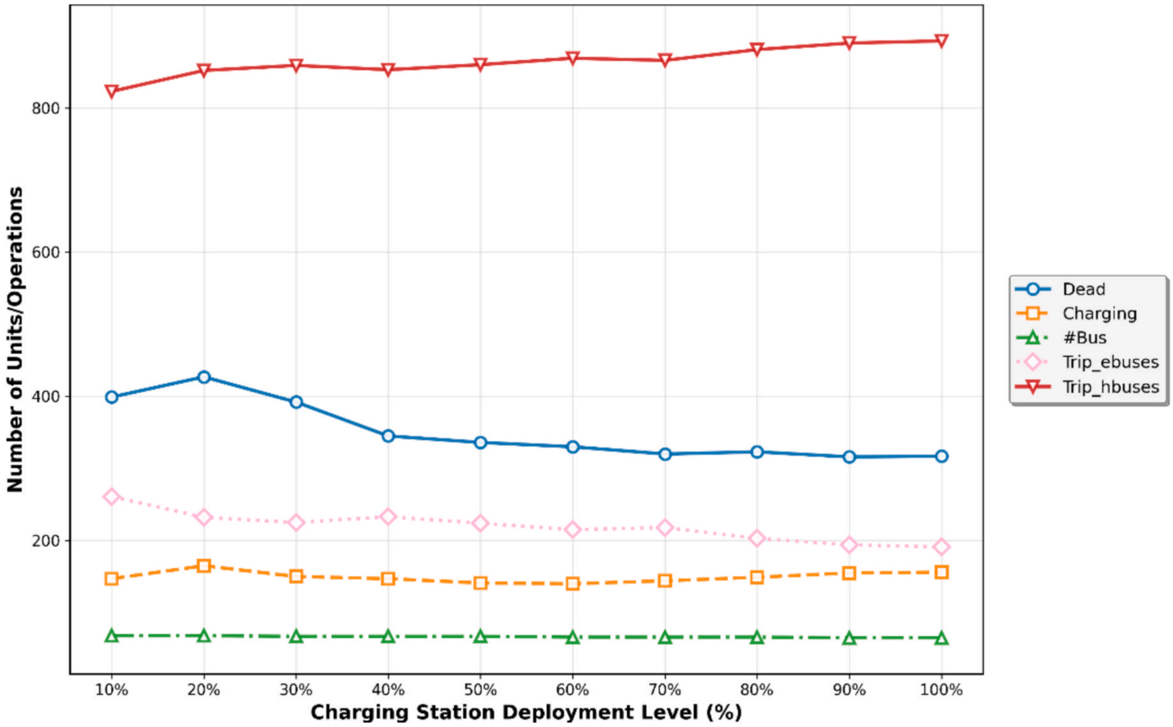


Fig. 20. Number of deadheads and chargings performed, buses employed, and trips covered by electric and hybrid buses at different levels of charging station deployment.

procurement costs and the Total Cost of Ownership. It is worth noting that SA-2CM tends to perform better for lower or higher percentages due to its MILP-based approach that allows it to find optimal charging scheduling for each pair of chains processed. Another key consideration is the range of solution windows considered in each iteration. A wider range demands increased computational effort (Fig. 18) but also opens up possibilities to make substantial improvements or potentially worsen current solutions by venturing into remote regions. This can be a double-edged sword, and careful tuning of the acceptance probability is essential;

- **CTB.** As expected from a heuristic algorithm, CTB is the fastest among the tools. While its solution quality may not be the best, it can serve as a practical tool, particularly in scenarios where frequent rescheduling is required or real-time adjustments are necessary due to changes in various exogenous parameters, such as energy costs;
- **SA-2CM.** SA-2CM operates on a different computational time scale and is the slowest among the algorithms (as shown in Fig. 19), which is unsurprising given its hybridization with the MILP model. Nevertheless, SA-2CM emerges as the optimal choice in scenarios featuring 257 trips or instances where fleet mixing is limited, consistently delivering optimal solutions for e-bus proportions below 30 % and above 80 %. In the scenario with 1084 trips, the rescheduling move has been adjusted to address the segmentation of two selected chains, thereby reducing the computational effort required by the MILP model;
- **GA.** The GA strikes a balance between CTB and SA-nCH. It is slightly faster than SA-nCH and offers superior solutions compared to CTB. However, the gain in computational efficiency does not sufficiently justify the compromise in solution quality, as its quality gap from CTB remains at approximately 6.4 %. Additionally, the GA employs a higher number of buses in comparison to SA-nCH, and this number escalates with the size of the timetable. This phenomenon can likely be attributed to the application of repair rules, which lead to an increased utilization of buses for resolving conflicts in infeasible solutions.

Statistical validation. For each large-scale timetable we compared the same four metaheuristics across the same ten fleet-composition levels (10 % – 100 % e-buses), yielding related samples ($k = 4$, $N = 10$). Because normality cannot be assumed, we again applied the non-parametric Friedman test to the FO values in Table 17 of Supplementary Materials. The results strongly rejected the null hypothesis of equal performance across algorithms for all instance sizes: 257 trips ($\chi^2(3) = 30.0, p < 0.0001$), 628 trips ($\chi^2(3) = 26.0, p < 0.0001$), and 1084 trips ($\chi^2(3) = 27.1, p < 0.0001$). To identify which algorithms differ we conducted a post-hoc Nemenyi multiple-comparison test. With k = 4 and N = 10, the CD is 1.48.

For the 257-trip instance, SA-2CM significantly outperformed all others (pairwise rank gaps $\geq 2.0 >$ CD). In the 628 and 1084-trip cases, SA-nCH held the best average rank but was statistically tied with SA-2CM (rank gap 0.2 $<$ CD). Both significantly outperformed GA and CTB (rank gaps $\geq 1.5 >$ CD). The advantage of GA over CTB was not statistically significant.These statistical findings reinforce the qualitative trends discussed earlier: SA-2CM dominates in the smallest large-scale case, while SA-nCH matches or exceeds SA-2CM as instance size grows; GA offers a middle-ground solution quality, and CTB remains the fastest but least effective in terms of cost minimisation. In Fig. 19, we depict the exploration trends of SA-nCH, specifically focusing on the scenario involving 257 trips, with variations in the percentage of e-buses (the corresponding data is summarized in Table 18 of the Supplementary Materials). This figure further emphasizes what we mentioned earlier about the gradual flattening of marginal gains. It is worth noting that the same pattern holds true for the scenarios involving 628 and 1084 trips; however, due to space constraints, we have chosen not to display those graphs. We can categorize these trends into two distinct groups, distinguished by the range of percentage improvements observed between pairs of trends.

The first group spans from 10 % to 50 % of e-buses, where the average marginal gain for consecutive percentage increments stands at a consistent 15.8 %. The cumulative improvement Δ_1 in the objective function score between the 10 % and 60 % scenarios amounts to a substantial 58 %. Conversely, the second group encompasses the range of 60 % to 100 % of e-buses. Within this range, we observe a significant drop in the overall improvement Δ_2 to 9 %, particularly evident after the 60 % scenario. This analysis highlights two significant observations. First, the choice of whether to apply a specific algorithm should be context-dependent. For instance, SA-2CM could serve as an offline tool for managing small or segmented timetables. In contrast, CTB could be seamlessly integrated as a real-time algorithm, capable of accommodating rapidly changing exogenous trends like energy costs. Lastly, SA-nCH and GA demonstrate their effectiveness when dealing with real-size timetables. Second, as the percentage of e-buses increases, the marginal gains progressively diminish. This phenomenon had already been highlighted in previous studies involving smaller instances and has now been reaffirmed in real-size scenarios. Consequently, it suggests that complete electrification of the bus fleet may not be economically advantageous for transport company operators, especially when considering other fleet or infrastructure-related costs.

5.3.2. Setting 2: variable percentage of terminals equipped with charging stations

In this section, we delve deeper into various charging infrastructure scenarios. Our aim is to explore the impact of increasing the percentage of terminals equipped with charging stations on KPIs, such as the number of deadhead and charging events, as well as the fleet size in use. We refrain from examining the trend in the objective function score across various instances, as it is intricately tied to the unitary cost values utilized in this study. In contrast, the count of events can yield more universally valuable insights. A similar experiment has previously been conducted by (Olsen et al., 2022) examining different scenarios that vary in terms of the proportion of charging stations at highly frequented stops within the complete set of stop points. They considered proportions of 10 %, 20 %, and 50 % charging stations. In our case study, we will increment the percentage by 10 % with respect to the number of terminals equipped with charging stations. We will also work with urban-sized instances, comprising timetables of 1084 trips generated using the CopulaGauss SDG method introduced in Section I.1 of the Suppementary Materials. Additionally, we will gather results by utilizing a bus fleet composition of 50 % e-buses and hybrid buses and employing the SA-nCH tool, which has demonstrated superior performance in real-size scenarios.

Fig. 20 illustrates the KPIs recorded at varying charging station deployment percentages, as detailed in Table 19 of the Supplementary Materials. As expected, higher charging station coverage leads to a decrease in deadheads and an increase in trips performed by electric buses.

Additionally, we observe a reduction in the number of hybrid buses in service, as well as a convex trend in the number of charging events. Surprisingly, unlike our analysis of fleet composition, increasing the distribution of charging stations in our urban context seems to have a relatively limited impact on the KPIs considered and the overall objective function scores.

Nonetheless, even a slight reduction or increase in the number of deadheads or charging events can yield a significant impact on solution feasibility, delay minimization, and operational or fixed costs when contextualized. Particularly, in regional scenarios where terminals with charging stations span vast areas, extended deadhead trips may have a cascading effect, influencing the feasibility of solutions and, consequently, the necessary fleet size and deadheading cost implications. Simultaneously, the utilization of technologies differing from fast charging can have a direct bearing on charging durations. Therefore, a reduction in the number of charging events assumes critical importance in optimizing service quality to the maximum extent possible.

6. Conclusions

In this study, we have devised a range of heuristic and metaheuristic techniques tailored specifically for addressing the Mixed-Fleet Multi-Terminal Electric Vehicle Scheduling Problem. Our primary focus has been on achieving scalability for handling large instances while still maintaining a reasonable level of optimality. This problem is characterized by the use of a mixed-fleet consisting of both electric and hybrid buses, as well as a limited number of terminals equipped with a restricted number of charging stations. Consequently, it necessitates careful scheduling of deadheading and charging events. The proposed methodologies include a Repeated Local Search heuristic which generates feasible solutions; a Simulated Annealing that makes use of two moves based on the heuristic and a MILP formulation proposed; and a Genetic Algorithm that employs repair rules to restore the feasibility of solution offspring.

We validated the performance of our approaches against an exact MILP formulation solved with IBM CPLEX, and conducted an extensive computational study over several real-world instances involving varying fleet compositions and infrastructure configurations. Our results confirm that: (i) All proposed methods approximate the optimal MILP solutions well in small-to-medium-sized instances, and (ii) They maintain strong scalability for urban-scale networks, where exact methods are computationally infeasible. Each algorithm exhibits distinct performance characteristics, thus, selection should be informed by the specific operational context, performance priorities (e.g., speed vs. solution quality), and available computational resources.

6.1. Managerial implications

Our findings have several direct implications for transportation planners and transit system managers tasked with real-size fleet electrification:

- While increasing the number of charging stations can reduce deadheading and charging delays, the marginal benefit decreases beyond a certain coverage level. Our results suggest that strategic terminal placement and targeted charging infrastructure at high-traffic hubs may offer better cost-efficiency than uniform full-terminal coverage. Planners should prioritize charging density over geographical dispersion in early-stage deployments.
- The operational benefit of fleet electrification exhibits diminishing marginal returns. Adding more electric buses does not linearly reduce operational costs or emissions. This highlights the need for phased electrification guided by scenario-specific optimization studies. For mixed fleets, retaining a share of hybrid or alternative-fuel buses can act as a buffer against network disturbances or insufficient charging capacity.
- Our metaheuristic approaches are modular and computationally efficient, but differ in their suitability for operational contexts. The CTB heuristic is particularly well-suited for real-time decision-support systems, even on limited hardware, and can be extended to accommodate dynamic factors such as variable energy prices. In contrast, SA-2CM and GA are more appropriate for planned or offline scheduling, where longer computation times are acceptable and higher-quality solutions are prioritized, such as in scenario planning or strategic timetable design.
- Our case study in Luxembourg City, characterized by short trip lengths and moderate urban density, indicates that compact urban areas are well-suited for early electrification. However, generalization to larger cities with higher route complexity requires caution. Decision-makers in such contexts should use scalable heuristics to simulate various configurations before committing to infrastructure changes.

6.2. Limitations and future research

This study's empirical foundation is based on a single real-world case: Luxembourg City. Its unique characteristics, including a compact urban structure, low population density, and relatively short trip lengths, have likely influenced the scheduling dynamics, energy demands, and infrastructure usage patterns observed in our experiments. These features limit the direct generalizability of the findings to larger or more congested metropolitan areas, where fleet electrification faces different logistical and infrastructural challenges. Consequently, while our methods and findings are informative, they should be interpreted with respect to this specific operational context. To address these limitations and broaden the applicability of our work, future research will explore the following directions: (i) Introducing uncertainty models to account for real-time disruptions, such as delayed trip completions, variable energy consumption, or dynamic traffic conditions; (ii) Incorporating more realistic, non-linear models of battery charging and discharging, including degradation effects over time; (iii) Embedding constraints and interactions arising from Smart-to-Bus Grid systems, especially under scenarios of limited grid capacity or variable electricity pricing. (iv) Alternative propulsion technologies: Expanding the model to include a broader mix of vehicle types, such as hydrogen-powered or biofuel-based buses, to reflect evolving trends in sustainable transport.

The modular design of our heuristic framework lends itself well to these extensions. By adapting and layering additional components, our methods can evolve to handle more diverse, uncertain, and dynamic scheduling environments in the future.

CRediT authorship contribution statement

Tommaso Bosi: Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Data curation, Conceptualization. **Marco Rinaldi:** . **Andrea D'Ariano:** . **Francesco Viti:** .

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cie.2025.111782.

Data availability

The data that has been used is confidential.

References

European Environment Agency, "Greenhouse gas emissions from transport in Europe," 2022.

Pelletier, S., Jabali, O., Mendoza, J. E., & Laporte, G. (Dec. 2019). The electric bus fleet transition problem. *Transportation Research Part C: Emerging Technologies, 109*, 174–193.

International Energy Agency, "Global EV Outlook 2023 - Catching up with climate ambitions," 2023.

D. Jefferies and D. Göhlich, "A Comprehensive TCO Evaluation Method for Electric Bus Systems Based on Discrete-Event Simulation Including Bus Scheduling and Charging Infrastructure Optimisation," *World Electric Vehicle Journal*, vol. 11, no. 3, 2020.

Hall, D., & Lutsey, N. (2020). Electric Vehicle Charging Guide for Cities. *International Council on Clean Transportation*.

Amiri, A., Amin, S. H., & Zolfagharinia, H. (2023). A bi-objective green vehicle routing problem with a mixed fleet of conventional and electric trucks: Considering charging power and density of stations. *Expert Systems with Applications, 213*, Article 119228.

Y. Feng, A. (Avi) Ceder, S. Zhang, and Z. Cao, "Bus routing fine-tuning for integrated network-based demand and bus bridging for a disrupted railway system," *Expert Systems With Applications* vol. 242, p. 122825, May 2024.

Fei, F., Sun, W., Iacobucci, R., & Schmöcker, J.-D. (2023). Exploring the profitability of using electric bus fleets for transport and power grid services. *Transportation Research Part C: Emerging Technologies, 149*, Article 104060.

Noel, L., & Mccormack, R. (2014). A cost benefit analysis of a V2G-capable electric school bus compared to a traditional diesel school bus. *Applied Energy, 126*, 246–255.

Ceder, A. (2016). *Public Transit Planning and operation*. CRC Press.

A. Costa, I. Branco, and J. M. Pinto Paixão, "Vehicle Scheduling Problem with Multiple Type of Vehicles and a Single Depot," in *Computer-Aided Transit Scheduling*, J. R. Daduna, I. Branco, and J. M. P. Paixão, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 115–129.

Picarelli, E., Rinaldi, M., D'Ariano, A., & Viti, F. (2020). Model and solution methods for the mixed-fleet multi-terminal bus scheduling problem. *Transportation Research Procedia, 47*, 275–282.

Rinaldi, M., Picarelli, E., D'Ariano, A., & Viti, F. (2020). Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications. *Omega (Westport), 96*, Article 102070.

Perumal, S. S. G., Lusby, R. M., & Larsen, J. (2022). Electric bus planning & scheduling: A review of related problems and methodologies. *European Journal of Operational Research, 301*(2), 395–413.

Ibarra-Rojas, O. J., Delgado, F., Giesen, R., & Muñoz, J. C. (Jul. 2015). Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological, 77*, 38–75.

Shang, H., Liu, Y., Wu, W., & Zhao, F. (Oct. 2023). Multi-depot vehicle scheduling with multiple vehicle types on overlapped bus routes. *Expert Systems with Applications, 228*, Article 120352.

Gintner, V., Kliewer, N., & Suhl, L. (Aug. 2005). Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. *OR Spectrum, 27*(4), 507–523.

Kliewer, N., Mellouli, T., & Suhl, L. (Dec. 2006). A time–space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research, 175*(3), 1616–1627.

Pepin, A.-S., Desaulniers, G., Hertz, A., & Huisman, D. (Feb. 2009). A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling, 12*(1), 17–30.

Marín Moreno, C. A., Escobar Falcón, L. M., Bolaños, R. I., Subramanian, A., Escobar Zuluaga, A. H., & Granada Echeverri, M. (2019). A hybrid algorithm for the multi-depot vehicle scheduling problem arising in public transportation. *International Journal of Industrial Engineering Computations*, 361–374.

Chao, Z., & Xiaohong, C. (2013). Optimizing Battery Electric Bus Transit Vehicle Scheduling with Battery Exchanging: Model and Case Study. *Procedia-Social and Behavioral Sciences, 96*, 2725–2736.

J. Reuer, N. Kliewer, and L. Wolbeck, "The Electric Vehicle Scheduling Problem - A study on time-space network based and heuristic solution approaches," Conference on Advanced Systems in Public Transport 2015 - CASPT, 2015.

Van Kooten Niekerk, M. E., van den Akker, J. M., & Hoogeveen, J. A. (2017). Scheduling electric vehicles. *Public Transport, 9*(1–2), 155–176.

Rogge, M., van der Hurk, E., Larsen, A., & Sauer, D. U. (Feb. 2018). Electric bus fleet size and mix problem with optimization of charging infrastructure. *Applied Energy, 211*, 282–295.

Liu, T., & Ceder, A. (May 2020). Battery-electric transit vehicle scheduling with optimal number of stationary chargers. *Transportation Research Part C: Emerging Technologies, 114*, 118–139.

Olsen, N., Kliewer, N., & Wolbeck, L. (Sep. 2022). A study on flow decomposition methods for scheduling of electric buses in public transport based on aggregated time–space network models. *Cent Eur J Oper Res, 30*(3), 883–919.

Zhou, G.-J., Xie, D.-F., Zhao, X.-M., & Lu, C. (2020). Collaborative Optimization of Vehicle and Charging Scheduling for a Bus Fleet mixed with Electric and Traditional Buses. *IEEE Access, 8*, 8056–8072.

Wang, C., Guo, C., & Zuo, X. (2021). Solving multi-depot electric vehicle scheduling problem by column generation and genetic algorithm. *Applied Soft Computing, 112*, Article 107774.

Perumal, S. S. G., Dollevoet, T., Huisman, D., Lusby, R. M., Larsen, J., & Riis, M. (Aug. 2021). Solution approaches for integrated vehicle and crew scheduling with electric buses. *Computers and Operations Research, 132*, Article 105268.

Sistig, H. M., & Sauer, D. U. (Jun. 2023). Metaheuristic for the integrated electric vehicle and crew scheduling problem. *Applied Energy, 339*, Article 120915.

Parmentier, A., Martinelli, R., & Vidal, T. (May 2023). Electric vehicle fleets: scalable route and recharge scheduling through column generation. *Transportation Science, 57*(3), 631–646.

Zhou, Y., Meng, Q., Ong, G. P., & Wang, H. (Apr. 2024). Electric bus charging scheduling on a bus network. *Transportation Research Part C: Emerging Technologies, 161*, Article 104553.

Fusco, G., Alessandrini, A., Colombaroni, C., & Valentini, M. P. (Oct. 2013). A Model for Transit Design with Choice of Electric Charging System. *Procedia-Social and Behavioral Sciences, 87*, 234–249.

Xylia, M., et al. (Feb. 2019). Impact of bus electrification on carbon emissions: The case of Stockholm. *Journal of Cleaner Production, 209*, 74–87.

Lajunen, A. (Jan. 2014). Energy consumption and cost-benefit analysis of hybrid and electric city buses. *Transportation Research Part C: Emerging Technologies, 38*, 1–15.

Goeke, D., & Schneider, M. (Aug. 2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research, 245*(1), 81–99.

Feng, W., & Figliozzi, M. (Jan. 2013). An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market. *Transportation Research Part C: Emerging Technologies, 26*, 135–145.

Bektaş, T., Ehmke, J. F., Psaraftis, H. N., & Puchinger, J. (May 2019). The role of operational research in green freight transportation. *European Journal of Operational Research, 274*(3), 807–823.

Steward, D. (2017). "Critical elements of vehicle-toGrid (V2G) Economics," Report National Renewable Energy. *The Laboratory*.

J. A. Manzolli, J. P. F. Trovão, and C. Henggeler Antunes, "Electric bus coordinated charging strategy considering V2G and battery degradation," Energy, vol. 254, p. 124252, Sep. 2022.

Thingvad, A., Calearo, L., Andersen, P. B., & Marinelli, M. (2021). Empirical capacity measurements of electric vehicles subject to battery degradation from V2G services. *IEEE Transactions on Vehicular Technology, 70*(8), 7547–7557.

Li, J. Q. (2014). Transit bus scheduling with limited energy. *Transportation Science, 48*(4), 521–539.

Zhang, M., Yang, M., Li, Y., & Huang, S. (2024). Optimization of electric bus scheduling considering charging station resource constraints. *Transportation Research Record: Journal of the Transportation Research Board, 2678*(3), 13–23.

Jovanovic, R., Bayhan, S., & Voß, S. (2023). "Matheuristic Fixed Set Search Applied to Electric Bus Fleet Scheduling," in Learning and Intelligent. *Optimization*, 393–407.

Vendé, P., Desaulniers, G., Kergosien, Y., & Mendoza, J. E. (2023). Matheuristics for a multi-day electric bus assignment and overnight recharge scheduling problem. *Transportation Research Part C: Emerging Technologies, 156*, Article 104360.

Yao, E., Liu, T., Lu, T., & Yang, Y. (Jan. 2020). Optimization of electric vehicle scheduling with multiple vehicle types in public transport. *Sustainable Cities and Society, 52*, Article 101862.

Kost, V., Merakou, M., & Gkiotsalitis, K. (Apr. 2025). Electric Bus Scheduling Problem with Time Windows and Stochastic Travel Times. *Information, 16*(5), 376.

Eglese, R. W. (1990). Simulated annealing: A tool for operational research. *European Journal of Operational Research, 46*(3), 271–281.

Ciancio, C., Laganà, D., Musmanno, R., & Santoro, F. (2018). An integrated algorithm for shift scheduling problems for local public transport companies. *Omega (Westport), 75*, 139–153.

Nourani, Y., & Andresen, B. (Oct. 1998). A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General, 31*(41), 8373–8385.

Bertsimas, D., & Tsitsiklis, J. (Feb. 1993). Simulated Annealing. *Statistical Science, 8*(1).

Reeves, C. R. (Aug. 1997). Feature Article—Genetic Algorithms for the Operations Researcher. *INFORMS Journal on Computing, 9*(3), 231–250.