PhD-F-2025-042
The Faculty of Science, Technology and Medicine

DISSERTATION

Defense held on 27 March 2025 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN SCIENCES DE L'INGÉNIEUR

by

Yumeng FANG

ELECTRIC VEHICLE DEMAND-RESPONSIVE TRANSPORT WITH TRANSIT INTEGRATION

Dissertation defense committee

Dr Tai-Yu Ma, dissertation supervisor
Research Scientist, Luxembourg Institute of Socio-Economic Research

Dr Francesco Viti, Vice Chairman
Professor, Université du Luxembourg

Dr Joachim Arts, Chairman
Professor, Université du Luxembourg

Dr Joseph Chow
Professor, New York University

Dr Pieter Vansteenwegen
Professor, Katholieke Universiteit Leuven (KU Leuven)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ALNS	Adaptive Large Neighborhood Search
BKS	Best Known Solution
CS	Charging Station
CSCC	Charging Station Capacity Constraint
DA	Deterministic Annealing
DARP	Dial-A-Ride Problem
DRFS	Demand-Responsive Feeder Service
DRT	Demand Responsive Transport
EDARP	Electric Dial-A-Ride Problem
EDRT	Electric Demand-Responsive Transport
EIDARP	Electric Integrated Dial-A-Ride Problem
EIDRT	Electric Integrated Demand-Responsive Transport
EV	Electric vehicle
EVRP	Electric Vehicle Routing Problem
IDARP	Integrated Dial-A-Ride Problem
IDRT	Integrated Demand Responsive Transport
KPI	Key Performance Indicator
LNS	Large Neighborhood Search
MILP	Mixed-Integer Linear Programming
OD	Origin – Destination
PDP	Pickup and Delivery Problem
PT	Public Transportation
SBRP	School Bus Routing Problem
SoC	State of Charge
VKT	Vehicle Kilometers Traveled
VNS	Variable Neighborhood Search
VRP	Vehicle Routing Problem

ACKNOWLEDGEMENTS

The journey of my PhD has been filled with both challenges and triumphs. I am deeply grateful to everyone I have met and to those who have supported me along the way, and LISER and the university for caring about the well-being of PhD students with warm administrative support and a comfortable working environment.

First, I would like to express my gratitude to my supervisor Dr. Tai-Yu Ma for his guidance. From having almost no research experience to gradually growing into a researcher, I have learned so much under your mentorship. Your dedication to keeping high standards and precision has influenced me. I will take these good qualities in my future work.

I am also grateful to Prof. Francesco Viti for always giving me positive feedback and introducing me to MobiLab. My PhD journey would not have been the same without my time at MobiLab. The positive energy from both past and present team members has always been a source of motivation for me. Thanks to everybody in the team. A special thanks to Dr. Richard Connors and Dr. Haruko Nakao, who are also the M-EVRST project members and co-authored one of the papers in the dissertation. Your suggestions and advice throughout project meetings are valuable, and you are always willing to help whenever I have questions.

I would also like to extend my gratitude to my defense committee members, Prof. Joseph Chow, Prof. Joachim Arts, and Prof. Pieter Vansteenwegen, for taking the time to read and evaluate my dissertation. I would like to express my special gratitude to Prof. Joseph Chow, also a member of my CET, for your valuable feedback throughout the CET meetings and for traveling to Luxembourg for my defense. I greatly appreciate your support and commitment.

Finally, to my dearest friends in China, Alicia, Wenwen, Xuan, and Yan. Despite the physical distance between us, we always encourage and being supportive to each other. To my family, especially my mother, I am forever grateful; without your encouragement, I would not have made it to pursue my PhD. And last but not least, thank you, Shihui, for always standing by my side, offering encouragement through my ups and downs.

PUBLICATIONS

Peer-reviewed journal articles

Published

Ma, T.Y., Fang, Y., 2021. Survey of charging management and infrastructure planning for electrified demand-responsive transport systems: methodologies and recent developments. Eur. Transp. Res. Rev.

Ma, T.-Y., Fang, Y., Connors, R.D., Viti, F., Nakao, H., 2024. A hybrid metaheuristic to optimize electric first-mile feeder services with charging synchronization constraints and customer rejections. Transp. Res. Part E Logist. Transp. Rev. 185, 103505. https://doi.org/10.1016/J.TRE.2024.103505

Under review

Fang, Y., Ma, T.-Y., Viti, F., 2024. Electric integrated dial-a-ride services with capacitated charging stations, multiple depots, and customer rejections (submitted to Public Transport).

Fang, Y., Ma, T.-Y., 2025.A hybrid large neighborhood search algorithm for the integrated dial-a-ride problem using electric vehicles. (submitted to Transporation Research Part E: Logistics and Transportation Review).

Peer-reviewed conference proceedings

Fang, Y., Ma, T.Y., 2023. Demand Responsive Feeder Bus Service Using Electric Vehicles with Timetabled Transit Coordination, in: Lecture Notes in Intelligent Transportation and Infrastructure. Springer Nature, pp. 91–103. https://doi.org/10.1007/978-3-031-23721-8_7/COVER

Ma, T.Y., Fang, Y., Connors, R., Viti, F., Nakao, H., 2023. A Fast Algorithm to Optimize Electric First-Mile Feeder Services with Charging Synchronization Constraints and Customer Rejections. IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC 133–139. https://doi.org/10.1109/ITSC57777.2023.10422459

Conference presentations

Fang, Y., Ma, T.-Y., 2022. Departure-expanded network for electric demand responsive feeder bus coordination with timetabled transit. Presented at 10th symposium of the European Association for Research in Transportation (hEART), June 1-3, 2022, Leuven, Belgium.

Fang, Y., Ma, T.-Y., 2022. Demand responsive feeder bus service using electric vehicles with timetabled transit coordination. Presented at the 6th Conference on Sustainable Urban Mobility (CSUM2022), August 31-September 2, 2022, Skiathos Island, Greece.

Fang, Y., Nakao, H., Connors, R., Ma, T.-Y., and Viti, F., 2023. Impact of urban morphology on the reliability of electric on-demand feeder services. Presented at International Symposium on Transport Network Resilience (INSTR), December 13-14, 2023, Hong Kong, China.

Fang, Y., Ma, T.-Y., Viti, F., 2024. The electric dial-a-ride problem with capacitated charging stations, multiple depots, and customer rejection. Presented at the 33rd European Conference on Operational Research (EURO 2024), June 30-July 3, 2024, Copenhagen, Denmark.

ABSTRACT OF THE DISSERTATION

Integrated Demand-Responsive Transport (IDRT), which combines demand-responsive transport services with regular transit systems, is widely recognized as an effective strategy to mitigate the impact of standalone demand-responsive services on traffic congestion and the environment. However, successful implementation of such service must address customer inconvenience, as transfers between demand-responsive vehicles and transit services often discourage ridership. As the transition towards sustainable mobility accelerates, it is essential to incorporate electric vehicles into IDRT to enhance environmental benefits. Involving electric vehicles in the IDRT system also brings additional challenges, particularly in managing charging operations and ensuring service reliability. This dissertation introduces an Electric Integrated Demand-responsive Transport (EIDRT) service, in which electric buses operates with fixed-route transit to effectively meet customer demand.

To address the complexity of the EIDRT problem, we first investigate a meeting-point-based first mile feeder service utilizing electric buses. The objective is to minimize bus operational costs and customer inconvenience, including reducing customers' waiting time at transit stations. Bus charging operations consider capacitated charging stations. A Mixed-Integer Linear Programming (MILP) formulation is developed using a layered graph structure and a metaheuristic solution algorithm is proposed. Computational experiments demonstrate that the metaheuristic produces good-quality solutions with around 1 or 2 minutes to solve 100-customer test instances. Results also show that the layered-graph significantly reduces computational time.

Next, we extend the meeting-point-based first-mile feeder service to a many-to-many EIDRT service that connects customers from their origins to destinations. A MILP formulation is proposed with the objective function minimizing both bus operational costs and customer travel times. To reduce customer inconvenience, the service incorporates synchronization between demand-responsive buses and transit departures, along with a maximum inter-modal transfer time. Capacitated charging stations are also included to reflect realistic recharging operations. The MILP formulation is built on a departure-expanded transit graph, incorporating the layered graph concept to better represent the transit network within a service area. A hybrid large neighborhood search algorithm is proposed to efficiently solve the problem, addressing the challenges of multi-modal routing and capacitated charging stations. The algorithm is benchmarked against eight-hour solutions by a MILP solver, demonstrating efficiency and better solution quality for up to 100 customers with two transit lines.

Lastly, the performance of the EIDRT is assessed by a set of experiments on scenarios reflecting real-world problem size and a case study. The findings provide valuable insights for operators regarding the trade-offs between operational costs and customer convenience, particularly focus on fleet size, buses' state-of-charge, and transit networks. Results indicate that while the EIDRT service saves vehicle kilometers traveled compared with non-integrated demand-responsive services, maintaining a high-quality service might still require the same or a larger bus fleet size. Moreover, the results show that the service performs the best when the transit network in the service area is well-connected. The case study further compares the EIDRT service with existing transportation options, confirming significant reductions in bus operating costs relative to non-integrated demand-responsive services. Similar results are found that bus travelling costs are saved significantly compared to non-integrated demand-responsive services. In terms of customer travel experiences, EIDRT achieves substantially lower customer travel times compared to traditional public transport.

Chapter 1 Introduction

With the advent of digital platforms and real-time data processing, ride-hailing services have gained popularity for their ability to provide customized and efficient mobility solutions. Transport network companies such as Uber, Lyft and DiDi have significantly contributed to on-demand transport services, transforming the urban mobility landscape and altering the public's travel habits. While these services complement public transportation and offer increased convenience to passengers, their rapid expansion has also introduced new challenges and problems. Henao and Marshall (2019) conducted a survey in Denver, USA, analyzing passengers using on-demand services such as Lyft, UberX, LyftLine, and UberPool. The survey revealed that on-demand services increased the Vehicle Kilometers Traveled (VKT) by approximately 83.5% compared with their absence, with most induced VKT stemming from public transport demand. Similarly, Tirachini and Gomez-Lobo (2020) found that on-demand trips primarily replace demand from public transport and create induced demand, with each new trip increasing VKT by an average of 1.79 km. This increased VKT led to greenhouse gas emissions and potential traffic congestion, as shown by Schaller (2021), who investigated data from Uber and Lyft in both dense and suburban areas in the US. Expanding infrastructure, implementing traffic zone restrictions, introducing congestion pricing, or fleet capping are common strategies for managing the impact of on-demand services (Behroozi, 2023; Zha et al., 2018). Alternatively, integrating Demand-Responsive Transport (DRT) services with existing fixed-route transit offers a sustainable solution to mitigate these negative impacts while maintaining high customer service levels (ARUP, 2021).

To accelerate the shift toward sustainable mobility, the European Parliament and Council reached an agreement requiring all new cars and vans registered in Europe to be zero-emission by 2035[1]. In response, transport network companies are increasingly involving Electric Vehicles (EVs) into their fleets to lower operational costs and contribute to environmental sustainability. However, this transition introduces new operational complexities. Research on the electrification of ride-hailing services in the United States indicates that EVs often require multiple daily charging operations, primarily relying on fast chargers to reduce vehicle downtime (Jenn, 2019). That can lead to a substantial rise in charging expenses, with estimates suggesting an increase of up to 25% for operators (Pavlenko et al., 2019). Furthermore, growing congestion at public charging stations (CSs) may force companies to develop private charging infrastructure to meet their operational needs (George and Marzia, 2018). These challenges highlight the need for in-depth research on optimizing charging strategies and infrastructure planning for electrified DRT systems to enhance efficiency and cost-effectiveness.

This chapter begins by introducing DRT services, followed by an exploration of transit integration and EV incorporation within DRT systems. Lastly, the research gap, questions and structure of this dissertation are outlined.

1.1 Demand-responsive transport

Demand-Responsive Transport (DRT) is a flexible transportation service in which vehicle routes and schedules are generated based on passengers' demand (Parragh et al., 2011). This service is neither fully

[1]https://www.consilium.europa.eu/en/policies/fit-for-55/

fixed-route like traditional public buses nor as flexible as individual taxi rides. DRT typically operates within a predefined service area and time frame, providing door-to-door or meeting-point-based services for user requests. This transportation system is particularly effective in areas with low demand for traditional public transportation, such as rural regions, or as a feeder service for mainline transit systems in urban areas (Calabrò et al., 2022). A DRT service does not have to be fully flexible. A semi-flexible DRT service operates with schedule and/or route deviation from a fixed service to accommodate demand changes and ensure lower operational costs. A comprehensive survey of planning a semi-flexible system can be found in Errico et al. (2013) A feeder DRT service is defined as many-to-one DRT, while many-to-many DRT serves customers between different origins and different destinations (Vansteenwegen et al., 2022). The authors also classify DRT systems as dynamic and static, based on the minimum time required between the time of request and the preferred departure time. A dynamic DRT service accepts customers during operations (online) or before starting the service (offline), whereas a static one plans all customer trips before the start of the planning horizon.

To design and manage efficiently DRT systems, researchers and practitioners employ various modelling approaches. These methods aim to optimize different aspects of DRT operations, including demand forecasting, vehicle routing, scheduling, and service quality assessment. The main modelling approaches include optimization-based approaches and simulation-based approaches.

Optimization-based approaches focus on determining the most efficient allocation of vehicles and routes to meet passenger demand. These problems are often formulated as extensions of the Dial-A-Ride Problem (DARP) (Cordeau and Laporte, 2007). Exact algorithms, such as the branch-and-cut and dynamic programming, are used for small-scale problems. However, for larger and more complex problems, heuristic and metaheuristics, such as genetic algorithms or Large Neighborhood Search (LNS), are applied to find near-optimal solutions within a reasonable time frame. Simulation-based approaches are used to assess the performance of DRT systems under various operational scenarios and system parameter configurations. The approaches help assess the impact of changes in demand patterns, fleet size, or routing strategies on system performance. Particularly, agent-based simulation is a widely used approach when it is relevant to simulating interactions between individual passengers and vehicles, which are treated as autonomous agents. These models are particularly useful for examining how real-time decision-making and user behavior affect overall system performance. The models can apply at operational, tactical, and strategic levels, with optimization models often required for trip scheduling at the operational level, and for determining bus fleet size and operational duration at the tactical level (Ronald et al., 2015).

1.2 Integrated demand-responsive transport

Although DRT services improve customers' inconvenience and accessibility in low-demand areas, 67% of DRT services fail in UK and 54% in Australasia (Currie and Fournier, 2020). Studies suggested that the high operational costs are strongly associated with these failures (Currie and Fournier, 2020; Enoch et al., 2006). Integrated Demand-Responsive Transport (IDRT) presents a promising solution to mitigate the high operational costs of DRT and the inefficiencies of traditional fixed-route transportation systems. IDRT is defined as a transportation system that integrates DRT services with fixed-route public transit networks. This integration aims to optimize its operational efficiency, enhance service accessibility, and improve the overall travel experience for users by leveraging the strengths of both demand-responsive transport services and fixed-route transport systems. The concept, first proposed in the 1980s as the Integrated Dial-A-Ride Problem (IDARP) (Wilson et al., 1976), involves

leveraging transit services for portions of customer journeys. Studies have modeled the service with optimization-based approaches (Aldaihani and Dessouky, 2003; Molenbruch et al., 2021; Posada and Häll, 2020) and simulation-based approaches (Fielbaum et al., 2024; Mortazavi et al., 2024). These studies all show that IDRT significantly reduces VKT, and improves transit ridership. This innovative service addresses critical challenges in traditional transportation, such as the inefficiency of fixed-route services in sparsely populated areas and environmental concerns. Despite the benefits for operators, transfers between different transport modes can inconvenience customers. Therefore, designing an efficient IDRT service that maintains desirable customer convenience is critical for the service implementation.

IDRT is similar to intermodal transportation, which is defined as transporting a passenger or load from its origin to destination using a sequence of at least two transportation modes (Crainic and Kim, 2007). While intermodal transportation focuses on coordinating transfers between modes, IDRT emphasizes on routing DRT vehicles and its integration with existing transit services to minimize a user-defined objective function. Table 1.1 summarizes the key differences between these two services.

Table 1.1: Comparison between intermodal transportation and IDRT

	Intermodal transportation	IDRT
Users	Freight or passengers	Passengers
Task	Schedule multiple modes of transport	Routing for passengers and vehicles
Purpose	Reduce inter-modal transfer time	Optimize cost and time efficiency
Flexibility	Use pre-defined routes for all transport modes	Use pre-defined transit routes and schedules, but flexible vehicle routes

Both intermodal transportation and IDRT are considered forms of multi-modal transportation services. However, IDRT not only improves the connectivity of different transport modes but also manages passenger and vehicle routes, making it more flexible and significantly more complex to model. By integrating DRT with mass transit, IDRT provides a viable approach to addressing urban mobility challenges while balancing sustainability and service quality.

1.3 Operating DRT systems using EVs

With the accelerating climate crisis, transportation agencies are urged to adopt cleaner vehicles to reduce CO2 emissions. Incorporating EVs in DRT service is becoming essential for operators. The limited driving range of EVs poses challenges for designing Electric Demand-Responsive transport (EDRT) services across three decision levels:

1. Strategic level: Decisions involve charging infrastructure, such as the location of charging stations, the number of chargers, and charger types (e.g., fast or slow).
2. Tactical level: Key considerations include EV fleet size and configuration (e.g., battery capacity, passenger capacity) to meet customer demand.
3. Operational level: Charging scheduling decisions address when, where, and how much to recharge.

At the operational level, the charging scheduling is often managed with vehicle routing in optimization models in the DRT services. The key elements related to EVs in the Electric Vehicle Routing Problem (EVRP) and Electric Dial-A-Ride Problem (EDARP) are summarized as follows (Ma and Fang, 2021).

Battery energy consumption

EV battery energy discharge/consumption depends on numerous factors such as vehicle speed, load, road gradient, temperature, and acceleration/breaking, etc. (Fiori et al., 2016). For simplification, it is often modeled as a linear function of distance traveled of vehicles.

Charging process modeling

Battery charging typically follows a non-linear pattern: a constant charging rate until about 70–80% of battery capacity, after which it slows down. Some studies use a simplified concave function to represent this non-linear charging process (Zalesak and Samaranayake, 2021). While many assume a linear-charging-speed to 80%-full charging policy. Recent studies incorporate non-linear charging function for more accurate charging process modeling (Kancharla and Ramadurai, 2020; Montoya et al., 2017a).

Charging operation policy

Charging policies are generally modeled based on full-recharge or partial-recharge policy for recharging EVs during each charging operation (Kucukoglu et al., 2021). The full-recharge policy involves recharging EVs to their maximum battery capacity or a predefined State of Charge (SoC) (e.g., 80%). while partial charging policy allows EVs to recharge only the amount they need. However, the full-recharge policy is often impractical near the end of service time when vehicles only need enough charge to return to the depot. The partial charge policy, on the other hand, reduces charging time and costs by recharging only the minimum amount energy to meet vehicles' energy demand (Felipe et al., 2014; Keskin and Çatay, 2016). An alternative approach is battery swapping, replacing depleted batteries with fully charged ones to minimize downtime, though it comes with higher installation costs (Vallera et al., 2021).

Charging costs

Most models assume charging costs are proportional to the amount of charged energy with constant electricity prices (Schneider et al., 2014). Some studies include access, parking, or opportunity costs during charging (Ma, 2021; Wang et al., 2018). Recent studies account for time-dependent energy prices and location-specific pricing to reflect dynamic charging prices, showing that such approaches can significantly reduce charging costs (Fehn et al., 2019).

Waiting at charging stations

Many models assume EVs can charge immediately upon arrival, ignoring chargers' might be occupied by other EVs. More advanced approaches model waiting times at charging stations based on the queueing theory, accounting for charger's occupancy and EVs' arrival patterns (Ammous et al., 2019; Jung et al., 2014; Keskin et al., 2019). Schoenberg and Dressler (2021) included route planning that integrates real-time charging station utilization data and multi-criteria decision-making. Some studies incorporated the capacitated CSs in the optimization model by only allowing EVs to visit CSs when chargers are not occupied by other EVs (Bruglieri et al., 2019; Ma et al., 2024; Pantelidis et al., 2022).

1.4 Scope and structure of the dissertation

Despite significant advancements in DRT, existing studies have largely focused on either IDRT services or EVs implementation in DRT, with limited emphasis on introducing EVs within IDRT. Furthermore, while some research has explored the use of optimization models for vehicle routing or charging scheduling, there is still a lack of studies addressing capacitated charging stations. Moreover, performance analysis of the IDRT system across various scenarios and in comparison with other existing services remain insufficiently explored, with even less focus on analyzing the impacts of integrating EVs into IDRT services.

This dissertation seeks to bridge these gaps by developing optimization models for efficient Electric Integrated Demand-Responsive (EIDRT) service operations management, incorporating charging scheduling and transit integration. The EIDRT service proposed can be classified as a static, many-to-many and fully flexible DRT (Vansteenwegen et al., 2022). The primary scope is at the operational level, focusing on DRT bus routing and charging scheduling to minimize a system costs considering both operator's costs and customer's inconvenience. Strategic and tactical decisions, such as CS location and bus fleet configuration, are beyond the scope of this dissertation.

The following research questions are brought:

Q1: How can the recharging of electric vehicles in EDRT services be efficiently scheduled while ensuring a high level of customer service?

Q2: How to manage customers' inter-modal transfers in EIDRT services to minimize their inconvenience?

Q3: What are the key factors that determine a desirable service operation for real-world applications of EIDRT services?

Q4: Are EIDRT services more beneficial for customers and service operators compared to existing transport services?

To answer these research questions, this dissertation sets out the following research objectives.

1. Design a novel Electric Integrated Demand-Responsive Transport service that integrates existing transit services with demand-responsive electric vehicles to enhance the system efficiency.
2. Develop an optimization-based approach for modeling for the EIDRT service considering both operational costs and customer inconvenience. Realistic charging scheduling considering capacitated charging stations is jointly optimized with vehicle routing.
3. Develop efficient solution algorithms for solving the problem of the EIDRT service for larger problem sizes and real-world applications.
4. Evaluate the EIDRT services under different system configurations and demand scenarios.

As discussed in previous sections, both the implementation of EVs and integration of transit into DRT presents significant challenges, applying EVs within the integrated DRT introduce even greater complexities. To address the challenges, this dissertation first models the first-mile feeder (many-to-one) EDRT service to connect to transit stations and subsequently extends the model to a many-to-many EIDRT service. To enhance the system efficiency, the first-mile feeder EDRT service adopts a meeting-point-based (stop-based) approach, where customers are required to walk to assigned meeting points, reducing vehicle kilometers traveled. The service is not restricted to a single transit station; instead, it incorporates multiple transit stations to enhance connectivity for customers. The two types of services are illustrated in Figure 1.1. Compared to the first-mile EDRT, the DRT buses in EIDRT serve for both

feeder service and door-to-door service, depending on the origins and destinations of customers and the characteristics of the transit network.

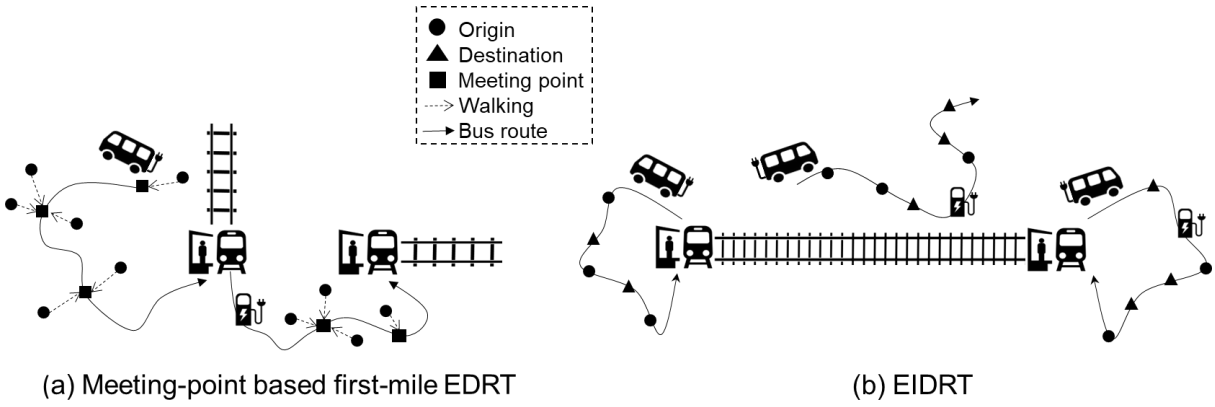


Figure 1.1: Illustrative example of the electric first-mile DRT and EIDRT services

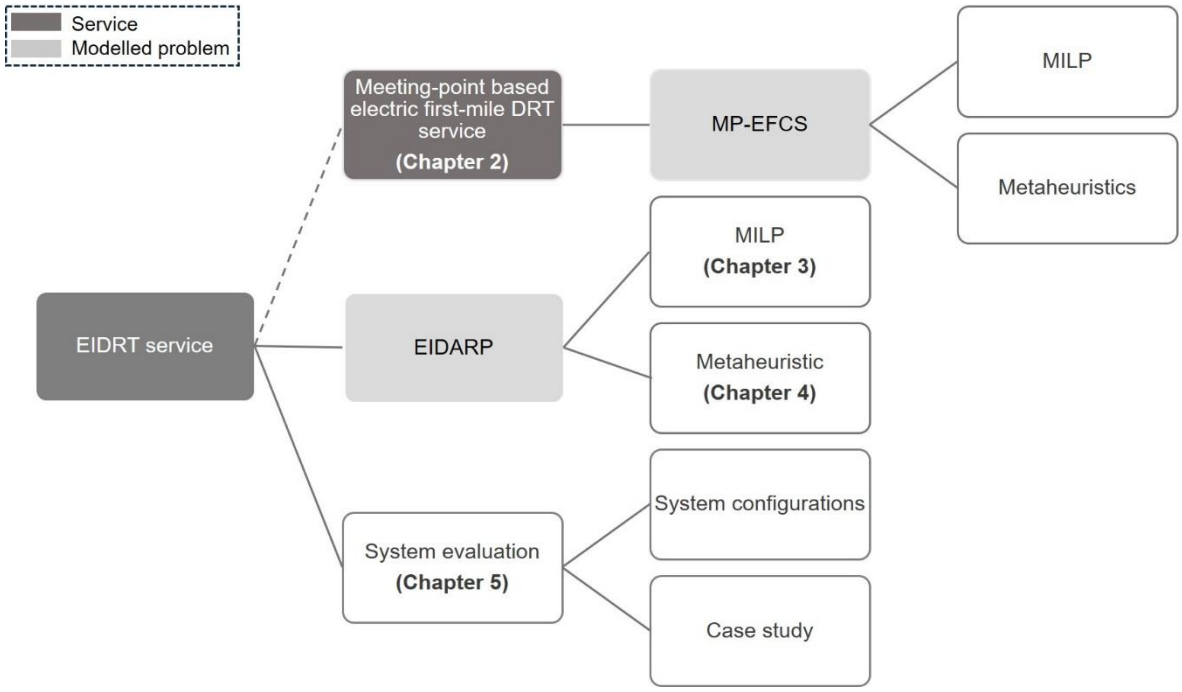


Figure 1.2: Structure of the dissertation

Figure 1.2 presents the structure of the remainder of the dissertation, and each chapter is organized as follows:

Chapter 2 develops a Mixed-Integer Linear Programming (MILP) approach for the Meeting-Point based Electric Feeder service problem with Charging Synchronization (MP-EFCS) to model the meeting-point-based first-mile EDRT service. A literature review on related studies is presented. A MILP model is developed with a departure-expanded graph to address the transfers between DRT buses and transit services. A metaheuristic solution method is proposed to optimize the problem, and computational studies are conducted to assess the performance compared with the state-of-the-art MILP solver.

Chapter 3 proposes a MILP approach for the Electric Integrated Dial-A-Ride Problem (EIDARP) to model the EIDRT service. A literature review on related studies is presented, focusing on modelling DRT services with transit integration. The departure-expanded graph in Chapter 2 is expanded for the EIDRT, and an efficient method for modelling the capacitated charging stations is developed. The modelling approach is compared with the state-of-the-art methods to analyze its efficiency by conducting numerical studies.

Chapter 4 develops a hybrid LNS metaheuristic to efficiently solve the problem of EIDRT for instances of realistic problem sizes. The literature review focuses on existing solution algorithms for related problems. The new contributions of the LNS metaheuristic focus on handling the charging scheduling of electric buses and transit integrated bus routes optimization. The performance of the metaheuristic algorithm is analyzed and benchmarked against the MILP solutions obtained by the state-of-the-art MILP solver.

Chapter 5 evaluates the performance of the EIDRT service through a series of experiments designed to assess its effectiveness under various system configurations. A case study using the ride data from a real-world DRT service is conducted to compare the performance of the EIDRT service with other existing transportation options.

Chapter 6 concludes the dissertation by summarizing key findings and discusses relevant directions for future research. It provides the answers to the research questions in Chapter 1.

Chapter 2 The meeting-point-based electric first-mile demand-responsive service

This chapter is based on the two following published articles.

Fang, Y., Ma, T.-Y., 2023. Demand Responsive Feeder Bus Service Using Electric Vehicles with Timetabled Transit Coordination, in: Lecture Notes in Intelligent Transportation and Infrastructure. Springer Nature, pp. 91–103. https://doi.org/10.1007/978-3-031-23721-8_7.

Ma, T.-Y., Fang, Y., Connors, R.D., Viti, F., Nakao, H., 2024. A hybrid metaheuristic to optimize electric first-mile feeder services with charging synchronization constraints and customer rejections. Transp. Res. Part E Logist. Transp. Rev. 185, 103505. https://doi.org/10.1016/J.TRE.2024.103505.[2]

This chapter addresses the vehicle routing and charging operations for the meeting-point-based first-mile DRT service using EVs. A MILP model is proposed to minimize customers inconvenience by incorporating waiting times at transit stations, customer rejection, partial recharge, and realistic capacitated charging stations. The addressed problem is referred to as the Meeting-Point-based Electric Feeder service problem with Charging Synchronization (MP-EFCS).

As the problem size of the MILP grows exponentially with the number of requests and vehicles, a layered (departure-expanded) graph structure is employed to efficiently eliminate unnecessary nodes and arcs. A two-stage hybrid metaheuristic algorithm is proposed to solve the MP-EFCS problem efficiently. The first stage assigns customers to meeting points, modeled as a variant of the capacitated facility location problem, while the second stage applies a deterministic annealing (DA) based algorithm to jointly optimize bus routes and charging operations. The algorithm is validated and compared with the MILP solution, showing its efficiency to obtain good solutions.

This chapter is organized as follows. Section 2.1 reviews the related literature and identifies the research gaps. Section 2.2 describes the problem, introduces the layered graph structure, and presents the MILP formulation. Section 2.3 details the hybrid metaheuristic algorithm, which consists of three sub-problems: 1) customer-to-meeting-point assignment; 2) bus optimization with charging synchronization; 3) post-optimization (reinsert unserved customers). Section 2.4 presents the computational studies, including the test instance generation, algorithm parameter settings, and the performance analysis of the hybrid algorithm. Finally, Section 2.5 draws the conclusion.

2.1 Literature review

This section reviews related studies on Demand-Responsive Feeder Service (DRFS), meeting-point-based demand-responsive services, and modeling EVs' charging operations on capacitated charging stations for EVRP and EDARP.

[2] The author's contribution: MILP model formulation, computational studies, and manuscript preparation.

2.1.1 Demand-responsive feeder service

A DRFS provides an on-demand dial-a-ride service to increase public transport ridership (e.g. by transporting customers to transit stations). It is a cost-effective solution in rural areas where public transportation services are not well developed. Lee and Savelsbergh (2017) reviewed previous works and classified the full spectrum of DRT services according to different flexible route design concepts (zone-based vs. point-based deviation, route deviation, flexible route segments, or demand-responsive connection to a transit hub). The authors formulate the demand-responsive connector problem as a MILP to minimize the total routing cost. When booking a trip, each customer indicates their earliest pickup time and specifies their latest arrival time at the transit station (corresponding to scheduled transit departure time). The transit connector service ensures that there are no late drop-offs so that customers do not miss their trains. In addition, customers can be dropped off at alternative stations instead of predetermined transit stations (regional systems). Their computational results show that this flexibility provides cost savings of up to about 29% compared to the conventional regional systems. Chen and Nie (2017) analyzed an idealized demand-responsive connector system that uses demand-adaptive services to connect to fixed-route transit lines for customer transfers. They propose an analytical model to evaluate the impact of various system design parameters on system performance, including e.g. road spacing distance, value of travel time, walking distance, vehicle speed, and operating cost per vehicle mile/hour. Montenegro et al. (2021) further considered two types of bus stops: mandatory stops (must be visited by buses) and optional stops (visited when assigned with customers). Customers are assigned to the optional stops within a predefined maximum walking distance. The objective is to minimize a weighted sum of bus routing times, customer walking times, and the penalty associated with early/late arrivals with respect to the customer's desired arrival times. Bian and Liu (2019) analyzed the mechanism of first-mile ridesharing service from both the operator (customer-route matching) and customer (incentives, customized pricing) perspectives. Several studies evaluate the performance of demand-responsive feeder service using simulation approaches or empirical trip data from operators (see, for example, Alonso-González et al., 2018; Haglund et al., 2019; Ma et al., 2019, among many others). However, existing studies assume that all requests need to be served with sufficient fleet size. This assumption might not be practical for operators when operation costs for serving certain customers are relatively high. Few studies allow customers to be rejected by considering the trade-off between operator’s operational costs and customers’ travel time in the objective function.

2.1.2 Meeting-point-based demand-responsive service

Meeting-point-based DRT services require customers to walk a certain distance to an assigned meeting point for pickups or drop-offs, enhancing the operational efficiency compared to traditional door-to-door services. This concept is widely adopted in ride-sharing services (Aïvodji et al., 2016; Czioska et al., 2019; Stiglic et al., 2015). Its application needs to generate relevant meeting points near customers’ Origins – Destinations (ODs). For example, Czioska et al. (2019) proposed a clustering approach based on customers' spatial-temporal characteristics of their ODs. A set of meeting point candidates are generated based on i) identifying different clusters, ii) generating meeting point candidates within these clusters, and iii) selecting feasible meeting points for each customer. Finally, a neighborhood search approach algorithm was applied for vehicle routing optimization of the selected meeting points.

A similar approach is proposed in the School Bus Routing Problems (SBRP), where bus stop selection is integrated with the Vehicle Routing Problem (VRP). Students are assigned to nearby pickup stops and transported to school(s) (Ellegood et al., 2020; Park and Kim, 2010). To decrease inconvenience,

constraints such as maximum ride time, time windows, and walking distance are incorporated. Most heuristics for solving the SBRP separate bus stop selection and route generation (Ellegood et al., 2020). However, Schittekat et al. (2013) propose a method to address it simultaneously by decomposing the problem into a master problem (bus routing with stop selection) and a sub-problem (bus stop-passenger allocation). The authors use variable neighborhood decent metaheuristics for solving the master problem and solve the sub-problem exactly using the primal-dual labeling method. Melis and Sörensen (2022) combine DARP and SBRP to create an on-demand bus routing problem. However, the objective function does not consider customers' walking time, neglecting the trade-offs between customers' walking time and bus routing time.

This meeting-point-based concept has also been applied to DRFS. Nickkar et al. (2022) examined DRFS with temporary stops, aiming to minimize customer's and operational costs. The authors developed an algorithm for clustering customers and then solve the resulting bus routing problem. Montenegro et al. (2021) also introduce optional bus stops for DRFS, which buses visit only if customers are assigned. They utilize a LNS algorithm, starting with the initial solution assigning customers to their closest bus stops. The destroy operators randomly remove customers and then reassigned bus stops, while the repair operator evaluates alternatives starting with the second-closest stops based on predefined probabilities.

2.1.3 Capacitated charging stations in routing problems

The EVRP has been extensively studied in the past decades (see the comprehensive review of Kucukoglu et al., 2021). When both partial recharge and capacitated charging stations are considered, the problem becomes more challenging (Bruglieri et al., 2019; Froger et al., 2021; Lam et al., 2022). To allow multiple visits of vehicles at charging stations, most studies adopt a replication-based approach in which each charger is duplicated with an ordered set of dummy charger nodes (Bruglieri et al., 2019; Froger et al., 2019; Keskin et al., 2019; Montoya et al., 2017). The starting and terminating times of each charging operation are explicitly handled to ensure no capacity violation for multiple charging visits at each charger or charging station. Bruglieri et al. (2019) proposed a replication-based approach for modeling Charging Station Capacity Constraints (CSCCs) with a full-recharge policy. Each charging plug is duplicated with an ordered set of dummy charger nodes. A dummy charger node i cannot be visited by vehicles if its preceding node $i-1$ is not visited yet to avoid the symmetry issue. Instead of modeling charging station capacity, Keskin et al. (2019) introduced time-dependent waiting time at public charging stations to handle the expected charging waiting time when arriving at the charging station.

However, the replication-based approach is not efficient when the number of necessary dummy charging nodes becomes large for high charging demand with many charging stations (Froger et al., 2019). To address this issue, more efficient path-based approaches have been proposed (Bruglieri et al., 2019; Froger et al., 2019). The path-based approach is based on directed multigraph to model vehicle flow at charging stations under CSCCs. A similar idea has been developed for joint vehicle balancing and charging optimization of electric rideshare services (Pantelidis et al., 2022). Froger et al., (2021) improved the path-based formulation by introducing outgoing and ingoing flows for each charging station. They showed that the EVRP problem with path-based CSCCs could be optimally solved for the test instance with up to 10 requests. Lam et al. (2022) further developed a branch-and-cut-and-price approach using logic-based Benders decomposition and constraint programming to solve EVRP with CSCCs. The CSCCs are handled in the sub-problems to ensure no charging station capacity violation. If charging station capacity is violated, combinatorial Benders cuts are added to the master problem. This method can optimally solve the test instances with up to 25 requests.

In summary, meeting-point-based DRT shows significant potential for reducing operational costs. However, the joint optimization of customer walking time to meeting points and vehicle travel costs remains a challenging problem, especially as the number of available bus stops increases. While additional stops reduce customers' inconvenience, they also increase the complexity of the problem. When EVs are considered, the problem becomes even more complex due to the need to satisfy vehicle energy constraints and optimize charging schedules simultaneously.

2.2 Mathematical formulation

Notation

Sets	
$\mathcal{L}$	Set of layer indexes (set of departures), i.e. $\mathcal{L} = \{1, \dots, \lvert\mathcal{L}\rvert\}$
G	Set of physical meeting points, i.e. $G = \{1, \dots, \lvert G\rvert\}$
G'	Set of dummy (duplicated) meeting point vertices (nodes), i.e. $G' = \{g_i^\ell\}_{i \in G, \ell \in \mathcal{L}}$
D	Set of physical transit stations, i.e. $D = \{1, \dots, \lvert D\rvert\}$
D'	Set of dummy (duplicate) transit station vertices, i.e. $D' = \{d_i^\ell\}_{i \in D, \ell \in \mathcal{L}}$
S	Set of physical chargers, i.e. $S = \{1, \dots, \lvert S\rvert\}$
S'	Set of dummy (duplicated) charger vertices
R	Set of customers (i.e. location of origin of customers)
K	Set of electric buses
$\bar{V}$	Set of all vertices, i.e. $\bar{V} = G' \cup D' \cup S' \cup R \cup \{0, N+1\}$
V	Subset of vertices, i.e. $V = G' \cup D' \cup S'$
$V_0, V_{N+1}, V_{0,N+1}$	$V_0 = V \cup \{0\}, V_{N+1} = V \cup \{N+1\}, V_{0,N+1} = V \cup \{0, N+1\}$
$\mathcal{A}_c$	Set of walking arcs from customers' origins to meeting points, *i.e.* $\mathcal{A}_C = \{(r,j) \lvert r \in R, j \in G'\}$
$\mathcal{A}_B$	Set of bus arcs
Parameters	
$0, N+1$	Two duplicate instances of the depot
T	Planning horizon
Q^k	Capacity of bus k
$E^k_{min}, E^k_{max}, E^k_{init}$	Minimum, maximum, initial SoC of bus k
ℓ_{dep}, ℓ_d	The transit departure time and transit station node at layer ℓ
w_{ri}	Walking distance from customer r origin to meeting point $i \in G'$
c_{ij}	Distance from vertex i to vertex j
t_{ij}	Bus travel time from vertex i to vertex j $\forall i, j \in V_{0,N+1}$. Note that t_{rj} is the walking time from customer r origin to meeting point j, $\forall r \in R, j \in G'$.
φ	Detour factor
L_i	Maximum ride time for customers picked-up at node i. Calculated as 'straight line' ride time multiplied by a detour factor φ.
w_{max}	Maximum walking distance
u_i	Service time at vertex $i \in V$
e_i, l_i	Earliest and latest starting times of service at vertex $i \in V_{0,N+1}$
d_r, d_i	Drop-off transit station dummy node of customer $r \in R$ and $i \in G'$
ω	Penalty cost if a customer is rejected

α_s	Charging rate of charger $s \in S'$
β^k	Energy consumption rate per kilometer traveled for bus k
M	Large positive number
$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	Weighting coefficient for the objective function
ρ	Parameter used in the customer-to-meeting-point assignment problem (Eq. (2.36))
Auxiliary variables	
A_i^k	Arrival time of bus k at vertex i
W_i^k	Waiting time of bus k at node i
q_i^k	Passenger load of bus k when leaving vertex i
E_i^k	Battery energy level of bus k when arriving at the vertex i
p_i^k	Indicator: 1 if node i is visited by bus k, 0 otherwise
θ_i	Indicator being 1 if a dummy node is visited by a bus (used in Eq.(2.36)-Eq.(2.46))
Decision variables	
y_{ri}^k	1 if customer r is assigned to bus k and meeting point i, 0 otherwise
x_{ij}^k	1 if arc (i, j) is traversed by bus k, 0 otherwise
B_i^k	Beginning time of service of bus k at vertex i
τ_s^k	Charging duration for bus k at charger s, $s \in S'$

2.2.1 Problem description

We consider a first-mile DRFS operated by an operator in a rural area using a heterogeneous (in terms of capacity, battery size, and energy consumption rate) fleet of electric buses to complement the public transport system. To enhance system efficiency and reduce operational costs, the DRFS system adopts the concept of **meeting points** i.e. customers are offered a limited number of pick-up/drop-off meeting points, rather than a door-to-door service (Czioska et al., 2019; Ma et al., 2021) and the service is **punctuated** (e.g. the bus arrives at a transit station every 10-20 minutes to drop off the transit passengers). The system is operated as follows. For a given planning period, customers submit their ride requests in advance indicating their origin, the transit station to be dropped off, and their desired arrival time (corresponding to the pre-defined departure time of the transit service). Each request (customer) contains at least one passenger. The operator collects these ride requests and communicates whether they are accepted, the pickup time, and suggested meeting points. The operator's objective is to optimize bus routes so as to arrive at transit stations within a fixed buffer time (e.g. ≤ 10 minutes before the timetabled transit departure). We assume that customers are willing to walk from their origins to the suggested meeting points, up to some maximum acceptable walking distance. The SoC of the buses cannot fall below the reserve battery level throughout the route. Buses can be recharged only at operator-owned charging stations with partial recharge policy. Each station has a limited number of chargers. Charging operations cannot overlap at any charger i.e. waiting of a bus is not allowed at a charger/charging station. Given a set of customer requests, the objective is to optimize bus routes to meet these requests while considering the trade-off between system costs and customer inconvenience.

2.2.2 Layered (departure-expanded) graph

The MP-EFCS problem is defined on a directed graph where the set of nodes includes the set of customers R and the set of meeting point nodes (G'), transit station nodes (D'), charger nodes (S') and the two copies of the depot $\{0, N+1\}$. There are two sets of arcs: walking arcs $\mathcal{A}_c$ for customers walking from their origins to the meeting point nodes, and bus arcs $\mathcal{A}_B$ for bus routes starting from the depot node 0 and terminating at $N+1$. Each arc is associated with a distance and a travel time, calculated as the Euclidean distance divided by the average walking/bus speed.

The classic DARP manages buses' departure and arrival times and load changes easily, as each node is visited by at most one bus. However, in the MP-EFCS, the set of meeting points G' and that of transit stations D' are visited multiple times, necessitating multiple duplicates of these points/stations. It is crucial to reduce the duplication of theses nodes to a necessary minimum to reduce the problem size. To address this, we propose a **layered (departure-expanded) graph** model that organizes the node duplications according to the sorted departure timetables at transit stations. The following definitions are introduced for constructing the layered graph.

- **Layer**. A layer is a subset of meeting point and transit station dummy nodes at each ordered departure of the transit service timetable (see an example in Figure 2.1). The index set of the layers is defined as $\mathcal{L} = \{1,2,\dots,|\mathcal{L}|\}$, where the latest departure times at the higher layers are not less than those at the lower layers. A particular layer is characterized by the pair: scheduled arrival time ℓ_{dep} and associated transit station ℓ_d.
- **Compatible layers**: Two layers are compatible if the meeting point nodes (or transit station nodes) on the two layers can be visited by the same bus, i.e. given a meeting point, there is a vertical arc connecting a dummy node on one level to the one on the other level and vice versa. This compatibility can be verified by checking whether the arcs connecting the two dummy transit station nodes of the two layers can be part of the feasible solution. Let $[e_i, l_i]$ and $[e_j, l_j]$ denote the beginning and end of the time window at the dummy transit station node i and j, respectively. The end time window of a transit station node is the transit departure time at its associated layer. The width of the time window corresponds to the predefined buffer time. Two layers $(i, j), i < j$ are compatible if $e_i + t_{ij} \leq l_j$. If two layers are compatible, there are vertical arcs connecting the dummy nodes of the same meeting point or same transit station (layer 5 and 6 are compatible layers in Figure 2.1).
- **Layered graph**: We denote a layered graph as $\mathcal{G} = (V_{0,N+1}, \mathcal{A})$, where $V_{0,N+1}$ is a set of nodes structured with a ground layer with two copies of the depot and the dummy charger nodes, and a set of the layers with meeting point dummy nodes, i.e., $G' = \{g_i^\ell\}_{i\in G, \ell\in\mathcal{L}}$, and transit station dummy nodes, i.e., $D' = \{d_i^\ell\}_{i\in D, \ell\in\mathcal{L}}$, sorted according to the increasing order of departure time at transit stations. The latest time window of a transit station node d_i^ℓ is the transit departure time at layer ℓ, while the earliest time window is the departure time minus a predefined buffer time. The transit departure (layer) that customer r specifies is denoted as $\ell(r)$. $\mathcal{A} = \mathcal{A}_B \cup \mathcal{A}_C$ is a set of arcs where $\mathcal{A}_B$ is a set of arcs for bus routing after trimming infeasible arcs, and $\mathcal{A}_C$ is a set of (walking) arcs connecting customer's origins to reachable meeting points within a predefined maximum walking distance.

Figure 2.1 provides an example of a layered graph for the MP-EFCS problem. The example includes six customers, three meeting points, two transit stations, one charger and one depot. Each transit station has three departures, with the first station operating at a service frequency of 30 minutes and the second at 20 minutes. The layers corresponding to the transit departures selected by customers are retained in the graph. Layers 5 and 6 are considered compatible, as bus 1 can drop off a customer at transit station 2 before 6:50 and still have enough time to drop off two customers at station 1 in layer 6 before 7:00.

Travel between layers is accessed by predefined vertical arcs. For charging operations, bus 1 visits charger 1, utilizing the first dummy charger node, while bus 2 recharges at a different dummy node. In the illustrative example, it becomes evident that unnecessary layers can be easily pruned, significantly reducing the number of nodes and arcs in the graph. Since each node is associated with a specific departure time, this approach simplifies the management of time windows. Section 2.3.1 further details the methods employed to minimize the problem size through the layered graph approach.

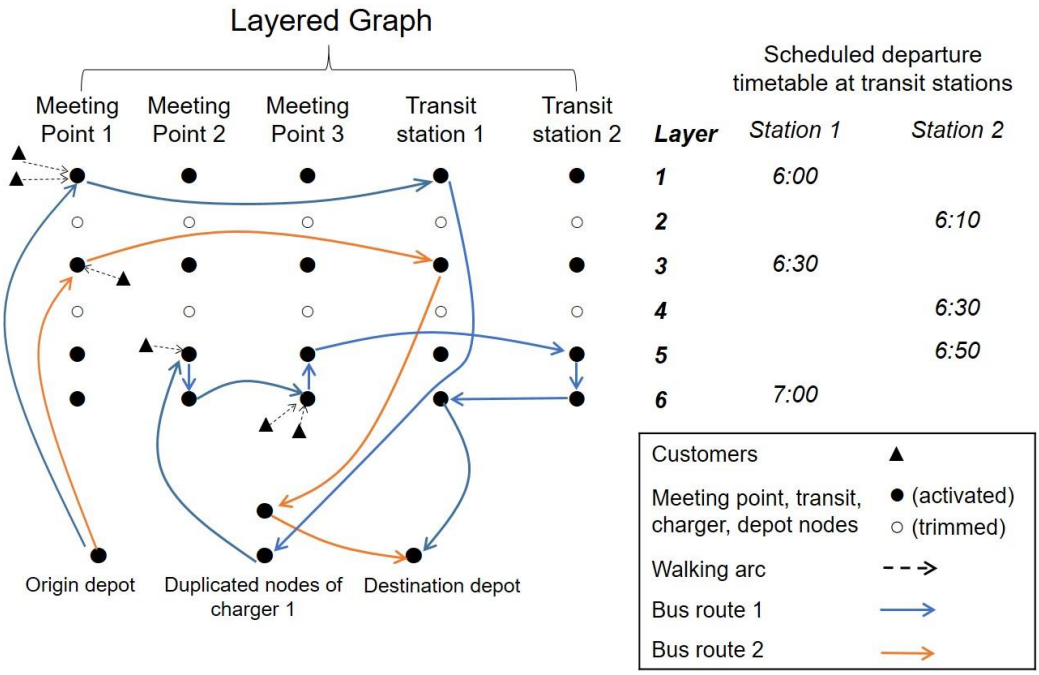


Figure 2.1: An illustrative example of bus routes in a layered graph (arcs omitted) for the MP-EFCS

2.2.3 MILP formulation

The MP-EFCS problem is formulated using a MILP approach. Note that a 3-index formulation is necessary since we consider a fleet of heterogeneous buses with different initial states of charge and bus energy consumption rate. To simplify the analysis, we formulate our model in a single-depot setting. This can be extended to a multi-depot model without difficulty by indicating the starting and returning depots of each bus (Bongiovanni et al., 2019; Braekers et al., 2014). The objective function minimizes the weighted sum of total bus travel time and total bus charging time (the first term), customer's total walking time (the second term), total 'excess' bus waiting time at transit stations (exceeding the acceptable fixed buffer time), and the total penalty for unserved requests. The weighting factors $\lambda_1, \lambda_2, \lambda_3, \omega$ are user-specified parameters to account for trade-offs between users' and operators' objectives.

$$Min\ Z = \lambda_1 \sum_{k\in K}\Big(\sum_{(i,j)\in \mathcal{A}_B} t_{ij}\, x_{ij}^k + \sum_{s\in S'} \tau_s^k\Big) + \lambda_2 \sum_{(r,i)\in \mathcal{A}_c}\sum_{k\in K} y_{ri}^k t_{ri} + \lambda_3 \sum_{k\in K}\sum_{i\in D'} W_i^k + \omega \sum_{(r,i)\in \mathcal{A}_c}\Big(1 - \sum_{k\in K} y_{ri}^k\Big) \tag{2.1}$$

The constraints can be grouped into four categories in terms of customer-to-meeting-point assignment constraints, bus routing constraints, bus energy constraints, and charging scheduling (synchronization) constraints.

$$\sum_{k\in K}\sum_{i\in G'} y_{ri}^{k} \le 1, \qquad \forall r \in R \tag{2.2}$$

$$\sum_{k\in K}\sum_{i\in G'} w_{ri} y_{ri}^{k} \le w_{max}, \qquad \forall r \in R \tag{2.3}$$

Constraint (2.2) ensures that each customer is served at most once. Constraint (2.3) imposes the maximum walking distance of customers to access a meeting point.

$$\sum_{j\in G'\cup S'\cup\{N+1\}} x_{0j}^{k} = 1, \forall k \in K \tag{2.4}$$

$$\sum_{i\in\{0\}\cup S'\cup D'} x_{i,N+1}^{k} = 1, \forall k \in K \tag{2.5}$$

$$\sum_{i\in V_0} x_{ij}^{k} \le 1, \qquad \forall k \in K, j \in G' \tag{2.6}$$

$$\sum_{i\in V_0} x_{ij}^{k} - \sum_{i\in V_{N+1}} x_{ji}^{k} = 0, \qquad \forall k \in K, j \in V \tag{2.7}$$

$$\sum_{r\in R} y_{ri}^{k} \le M_1 \sum_{j\in V_{\mathrm{N+1}}} x_{ij}^{k}, \qquad \forall k \in K, i \in G' \tag{2.8}$$

$$y_{ri}^{k} = 1 \Rightarrow \sum_{j\in V_0} x_{ji}^{k} = \sum_{j\in G'\cup D'} x_{jd_r}^{k}, \qquad \forall k \in K, i \in G', r \in R \tag{2.9}$$

$$x_{ij}^{k} = 1 \Rightarrow q_j^{k} = q_i^{k} + \sum_{r\in R} y_{rj}^{k}, \qquad \forall k \in K, i \in V_0, j \in G' \tag{2.10}$$

$$x_{ij}^{k} = 1 \Rightarrow q_j^{k} = q_i^{k} - \sum_{r\in R}\sum_{g\in G'} y_{rg}^{k}, \qquad \forall k \in K, i \in G', j \in D' \tag{2.11}$$

$$0 \le q_i^{k} \le Q^{k}, \qquad \forall k \in K, i \in V_{0,N+1} \tag{2.12}$$

$$B_j^{k} \ge B_i^{k} + u_i + t_{ij} - M_2(1 - x_{ij}^{k}), \qquad \forall k \in K, i \in V_0, j \in V_{N+1} \tag{2.13}$$

$$B_j^{k} \ge B_s^{k} + \tau_s^{k} + t_{sj} - M_2(1 - x_{sj}^{k}), \qquad \forall k \in K, s \in S', j \in \{G' \cup N + 1\} \tag{2.14}$$

$$x_{ij}^{k} = 1 \Rightarrow A_j^{k} = B_i^{k} + t_{ij} + u_i, \qquad \forall k \in K, i \in G' \cup D', j \in D' \tag{2.15}$$

$$W_i^{k} \ge B_i^{k} - A_i^{k} - M_2(1 - p_i^{k}), \qquad \forall k \in K, i \in D' \tag{2.16}$$

$$p_i^{k} = \sum_{j\in V} x_{ji}^{k}, \qquad \forall k \in K, i \in D' \tag{2.17}$$

$$A_{d_r}^{k} - B_i^{k} - u_i \le L_i + M_2\left(1 - y_{ri}^{k}\right), \qquad \forall k \in K, (r,i) \in A_C \tag{2.18}$$

$$e_i \le B_i^{k} \le l_i, \qquad \forall k \in K, i \in V \tag{2.19}$$

In terms of bus routing constraints, constraints (2.4) and (2.5) state that each bus leaves and returns to the depot. Constraint (2.6) ensures that each meeting point (dummy) node can be visited at most once by the same bus. Constraint (2.7) ensures bus flow conservation, while constraint (2.8) ensures

consistency between y_{ri}^k and x_{ij}^k. Constraint (2.9) ensures that the pickup and drop-off of a customer is served by the same bus. Constraints (2.10) and (2.11) update the bus occupancy at meeting points (pick-up locations) and transit stations (drop-off locations). Constraint (2.12) states the capacity (passenger load) constraint of the bus. Constraint (2.13) states that the beginning time of service at node j can start when the bus arrives at j. Constraint (2.14) states that when bus k leaves a charger node s, the starting time of service at successive node j is greater or equal than its starting time of service at the preceding node s plus the service time, charging time, and the travel time traversing arc (i, j). Constraint (2.15) computes the arrival time of buses at transit stations. Note that the hard time windows (fixed buffer time associated with the transit service timetable) are associated with transit station nodes only, not for meeting point nodes. To determine the excess waiting time when arriving at a transit station (before the buffer time), we introduce the arrival time variable, relevant only for the transit stations. Constraint (2.16) then measures the excess bus waiting time. Constraint (2.17) determines the value of the auxiliary variable indicating whether there are buses dropping off customers at transit station node i. Constraint (2.18) ensures the ride time of customers cannot exceed the maximum ride time, characterized as the shortest travel time (direct ride) multiplied by a pre-defined detour factor. Constraint (2.19) states that the beginning time of service at node i is constrained by the time window associated with that node.

$$E_0^k = E_{init}^k, \qquad \forall k \in K \tag{2.20}$$

$$E_{min}^k \le E_i^k \le E_{max}^k, \qquad \forall k \in K, i \in V \tag{2.21}$$

$$x_{ij}^k = 1 \Rightarrow E_j^k = E_i^k - \beta^k c_{ij}, \qquad \forall k \in K, i \in V_0 \backslash S', j \in V_{N+1} \tag{2.22}$$

$$x_{sj}^k = 1 \Rightarrow E_j^k = E_s^k + \alpha_s \tau_s^k - \beta^k c_{sj}, \qquad \forall k \in K, s \in S', j \in \{G' \cup N + 1\} \tag{2.23}$$

For bus energy constraints, Constraints (2.20) and (2.21) state the initial SoCs of the buses and their constraints. Constraints (2.22) and (2.23) ensure energy conservation with and without recharged energy when traversing an arc (i, j).

$$v_s = \sum_{k \in K} \sum_{j \in G' \cup N+1} x_{sj}^k, s \in S' \tag{2.24}$$

$$v_h \le v_l, \forall h, l \in S_o', o \in S, h < l \tag{2.25}$$

$$\sum_{k \in K} B_h^k \ge \sum_{k \in K} B_l^k + \sum_{k \in K} \tau_l^k - M_2(2 - v_h - v_l), \forall h, l \in S_o', o \in S, h < l \tag{2.26}$$

$$\tau_s^k + B_s^k \le M_2 \sum_{j \in G' \cup N+1} x_{sj}^k, \forall s \in S', k \in K \tag{2.27}$$

$$v_s \le 1, s \in S' \tag{2.28}$$

With respect to the constraints on charging scheduling (synchronization), constraints (2.24) - (2.28) ensure that each charger can be occupied by no more than one bus at a time, i.e., if multiple charging events are scheduled at the same charger, they cannot overlap. Creation of dummy charger nodes allows multiple visits by buses to the same chargers. Constraint (2.24) introduces an auxiliary binary variable to indicate whether a dummy charger node is visited or not. Constraint (2.25) states that the dummy charger nodes are visited in reverse order as they appear in the list of their associated physical charger nodes to eliminate the symmetry problem (see Figure 2.2) (see e.g. Froger et al., 2017; Lee and

Savelsbergh, 2017). Each dummy node can be connected maximum once by buses, and follows the inverse order from node l, then node *h*, (i.e., *l*-1), ..., 2,1. Constraint (2.26) ensures that charging of a bus can start only after the previous charging has finished. This means that the start time of a charging visit cannot be earlier than the start time of the previous charging visit of a bus plus its charging duration. In constraint (2.27), if a bus is not connected to a dummy charger node, its start time and charge duration are set to zero to determine which bus is connected to the dummy charger node in constraint (2.26). Since each dummy charger node can only be visited once, the unique connected bus and its associated charge start time and charge duration can be determined by constraint (2.26). Constraint (2.28) states that each dummy charger node can be visited at most once. Note that if the symmetry issue is not addressed when modeling multiple visits to charging stations, the computation time will increase significantly even for moderate problem sizes.

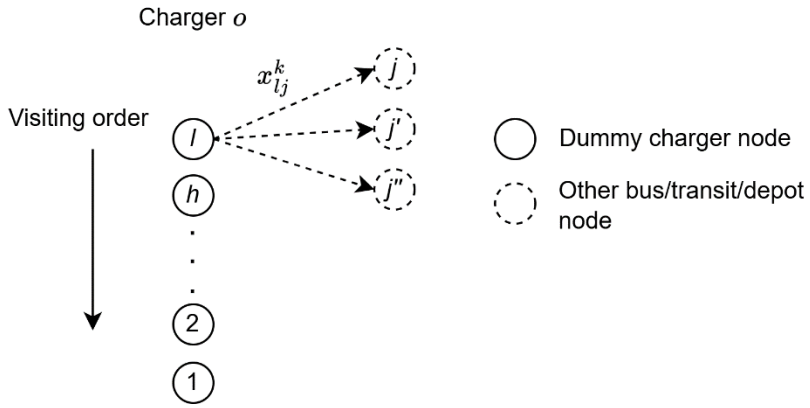


Figure 2.2: Illustration of visit order of dummy charger nodes for a charger

Finally, constraints (2.29) – (2.33) define the domain of the decision/auxiliary variables.

$$x_{ij}^k \in \{0,1\}, \qquad \forall k \in K, i,j \in V_{0,N+1} \tag{2.29}$$

$$y_{ri}^k \in \{0,1\}, \qquad \forall k \in K, r \in R, i \in G' \tag{2.30}$$

$$\tau_s^k \geq 0, v_s \in \{0,1\}, \forall k \in K, s \in S' \tag{2.31}$$

$$A_i^k \geq 0, B_i^k \geq 0, \forall k \in K, i \in V_{0,N+1} \tag{2.32}$$

$$p_i^k \in \{0,1\}, W_i^k \geq 0, \forall k \in K, i \in D' \tag{2.33}$$

Note that constraint (2.9) can be linearized equivalently as the following two constraints.

$$\sum_{j \in V_0} x_{ji}^k \geq \sum_{j \in G' \cup D'} x_{jd^r}^k - M_1\left(1 - y_{ri}^k\right), \qquad \forall k \in K, i \in G', r \in R \tag{2.34}$$

$$\sum_{j \in V_0} x_{ji}^k \leq \sum_{j \in G' \cup D'} x_{jd^r}^k + M_1\left(1 - y_{ri}^k\right), \qquad \forall k \in K, i \in G', r \in R \tag{2.35}$$

Constraints. (2.10) – (2.11), (2.15), and (2.22) – (2.23) can be extended equivalently in the similar way as above. The big positive numbers *M* are set as $M_1 = |R|$, $M_2 = T$.

2.3 Hybrid metaheuristic algorithm

Different from existing EDARP problem formulations (see e.g. Bongiovanni et al., 2019) and on-demand bus routing problems (Melis and Sörensen, 2022), the proposed MP-EFCS (Section 2.2) introduces greater flexibility by allowing partial bus recharging, minimizing buses' waiting times at transit stations, and rejecting customers. This added complexity necessitates to develop efficient metaheuristics to solve the problem efficiently for real-world applications.

We propose a hybrid metaheuristic with a **two-stage solution scheme**. The first stage assigns customers to meeting points, while the second stage solves the routing problem with charging synchronization constraints and a post-optimization procedure. The metaheuristic is hybrid as it integrates two additional subproblems formulated as two MILPs—customer-to-meeting-point assignment and post-optimization — into the solution framework. The bus routing procedure is based on DA metaheuristic that has been successfully implemented in solving the general heterogeneous DARP using non-EVs (Braekers et al., 2014). While the algorithm's structure is inspired by Braekers et al. (2014), it is tailored to the problem of MP-EFCS with additional procedures to reduce the computational time. The DA-based metaheuristic uses local search operators to refine solutions and employs a temperature parameter to accept worse solutions, enabling it to escape from local optima. This parameter gradually decreases during the optimization process until no further improvements or some stopping criteria are reached.

Key challenges for the MP-EFCS include synchronizing vehicle charging schedules and reassigning unserved customers to meeting points, which are addressed in subsequent sections and computational studies. This section first describes the preprocessing procedures on the layered graph to reduce the problem size, followed by the overall algorithm structure. Details on customer-to-meeting-point assignment, bus routing optimization, and post-optimization are provided in Section 2.3.3 – Section 2.3.5.

2.3.1 Preprocessing

Building on the preprocessing procedures for the DARP outlined by Cordeau (2006), the preprocessing steps for MP-EFCS involve time-window tightening and arc elimination in the layered graph described in Section 2.2.2. However, in the context of the meeting-point-based first-mile service, all the customers are treated as outbound users, and the time-window tightening is applied exclusively at meeting points. As customers specify their transit station and transit departure time, and each layer in the graph corresponds to a transit departure, non-selected layers are identified and eliminated in the beginning. The time windows at a meeting point $i \in G'$ at layer $\ell \in \mathcal{L}$ is tightened as $e_i = \max\{0, e_{\ell_d} - \varphi t_{i\ell_d} - \mu_i\}$ and $l_i = \min\{T, l_{\ell_d} - t_{i\ell_d} - \mu_i\}$, where ℓ_d is the transit station node at layer ℓ, and l_{ℓ_d} represents its transit departure time.

We eliminate the bus arc $\forall (i,j) \in \mathcal{A}_B$ if $e_i + t_{ij} + \mu_i > l_j$, or if they are **not** in one of the following situations:

- Arcs from node 0 to *G'*, *S'*, and $\{N+1\}$.
- Arcs from *D'* to S' and $\{N+1\}$.
- Arcs connect from meeting point nodes to different meeting point nodes and transit station nodes on the same layer.
- Arcs from transit station nodes to meeting point nodes on the higher layers.
- Arcs connect the meeting point and transit station nodes of the same physical location on two different layers if their layers are compatible.
- Arcs from *S'* to G' and $N+1$.

For the walking arc set $\mathcal{A}_C$, it contains only the arcs within the maximum walking distance of customers on the layer of the customer's desired train departures.

2.3.2 The structure of the hybrid metaheuristic

The overall structure of the algorithm is outlined in Algorithm 1 and consists of three main components: customer-to-meeting-point assignment, electric bus routing, and post-optimization to reinsert unserved customers. The input includes a preprocessed layered graph of the problem instance. The algorithm begins with a customer-to-meeting-point assignment (line 2). Once the meeting points are assigned, the problem is transformed into an EDARP for bus routing optimization, with the EDARP instance constructed in line 3. Let s represent the current solution, s_{best} the current best solution, and $c(s)$ the cost (objective function value of Eq. (2.1)) of solution s. An initial feasible solution s_{init} is generated as the best feasible solution found for n random solutions (i.e. n=100) using a greedy insertion approach (line 4). If a feasible solution is not found, n is increased to 1000 to improve the probability to find one.

The algorithm applies a randomly selected local search operator ls on the current solution s and obtains a temporary solution s' (lines 11-12). If the cost of s' is smaller than that of the current solution s plus a threshold value T (i.e. a worse solution but within the threshold), and there are no charging operation conflicts, a bus exchange operator (line 14) is applied on s' to further reduce the charging time of the buses of s'. This is because we assume that the initial SoCs of the buses are heterogeneous. Given two routes, this route switcher could reduce the total charging time if the bus with the higher SoC travels the longer route, thus reducing the amount of energy to be charged. To handle this operation efficiently, this operator sorts the buses by their total charging time, including the additional time to access the charging stations. A first bus is exchanged with a second bus (without changing the sequence of pickups and drop-offs of the route). If the resulting bus exchange and rescheduled charging operations (if any) improves cost without charging conflicts for all buses, then the current solution is updated. When the cost of s is smaller than that of the best solution s_{best}, and the number of used buses $n_k(s)$ does not exceed the fleet size $|K|$, update the best solution (lines 16-18), and reset the non-improvement count $i_{imp} = 0$. When s_{best} is not improved ($i_{imp} > 0$), reduce the threshold value by T_{max}/T_{red}, where T_{red} is the threshold reduction parameter. If $T < 0$, reset T randomly between 0 and T_{max}. When i_{imp} exceeds the maximum number of iterations, reset $i_{imp} = 0$ and the current solution as s_{best}.

The algorithm incorporates an early stop criteria (line 8-10), which stops the process if solution shows no more improvements after $100 \times n_{stagnant}$ consecutive stagnations (we check whether no improvement (relative improvement is no more than 0.05%) occurs for every 100 iterations of the main algorithm loop). Otherwise, the algorithm stops when the maximum number of iteration, $iter_{max}$, is reached. As the bus routing optimization relies on the customer-to-meeting-point assignment in the first stage, unserved customers may potentially be accommodated by reassigning their meeting points. To address this, we propose an efficient post-optimization procedure to refine the best solution obtained from the DA algorithm (line 30). The parameters T_{max}, T_{red}, n_{imp} and $n_{stagnant}$ are required to be tuned to optimize the algorithm's performance. The following sections present the mathematical formulations and algorithmic description of the three main components of the hybrid metaheuristic.

Algorithm 1. Hybrid metaheuristic algorithm for solving MP-EFCS

1. $\mathcal{G} \leftarrow$ *createLayerdGraph*$(V_{0,N+1})$
2. $(q, \mathcal{G}') \leftarrow$*customer-meeting-pointAssign*$(R, \mathcal{G})$
3. Set up EDARP$(q, \mathcal{G}')$ instance
4. $s_{init} \leftarrow$*generateInitSol(EDARP)*
5. Set the current solution $s = s_{best} = s_{init}$, $i_{imp} = 0$, $count_{stagnant} = 0$, $T = T_{max}$
6. **for** $iter = 1$: $iter_{max}$
7. $i_{imp} \leftarrow i_{imp} + 1; count_{stagnant} \leftarrow count_{stagnant} + 1$
8. **if** $count_{stagnant} = 100 \times n_{stagnant}$
9. Go to line 30
10. **end if**
11. $ls \leftarrow rand(LS)$
12. $s' \leftarrow ls(s)$
13. **if** $c(s') <$ c$(s) + T$ and *noChargingConflict*(s')=*true*
14. $s'' \leftarrow$ *busExchange*(s')
15. $s \leftarrow s''$
16. **if** $c(s) <$ c(s_{best}) and $n_k(s) < |K| + 1$
17. $s_{best} \leftarrow s; i_{imp} = 0; count_{stagnant} = 0$
18. **end if**
19. **end if**
20. **if** $i_{imp} > 0$
21. $T \leftarrow T - T_{max}/T_{red}$
22. **if** $T < 0$
23. $T \leftarrow random(0,1) * T_{max}$
24. **if** $i_{imp} > n_{imp} * n_k(s_{best})$
25. $s \leftarrow s_{best}; i_{imp} = 0$
26. **end if**
27. **end if**
28. **end if**
29. **end for**
30. **return** $s^*_{best} \leftarrow$*postOptimization*(s_{best})

2.3.3 Customer – meeting points assignment

Given customers' origins and the maximum walking distance constraint, it is necessary to determine the assignment of customers to meeting points by balancing bus routing times and customer inconvenience (walking time). This customer-to-meeting-point assignment problem is formulated using a MILP approach, which is a variant of the capacitated facility location problem.

Recall that G'_ℓ represents the subset of dummy meeting point nodes for layer ℓ, where ℓ belongs to the set of layers $\mathcal{L}$. The objective function (Eq. (2.36)) minimizes the weighted sum of total customer walking time and bus travel time between activated meeting points (with positive assigned customers). The weights λ_1 and λ_2, defined in the original objective function (Eq. (2.1)), control the relative importance of these two components. ρ_ℓ is a non-negative coefficient to be tuned to balance the trade-off between customer walking time and bus travel time (see Section 2.4.2 for the tuning method). The

decision variable y_{rj} determines whether customer r is assigned to meeting point j. θ_j and z_{ij}^{ℓ} are two intermediate binary variables indicating whether meeting point *j* has positive assigned customers, and whether arc (i, j) in layer ℓ is used, respectively. Constraint (2.38) ensures that each customer is assigned to one meeting point, while constraint (2.39) restricts the number of customers assigned to each meeting point to not exceed the maximum bus capacity. Constraint (2.40) ensures a meeting point *j* is activated if any customers are assigned to it. Note that the second term in the objective function (Eq. (2.36)) does not represent the actual total bus traveling cost. Instead, it provides an estimation designed to assist in the customer-to-meeting-point assignment process.

$$\text{Min } \lambda_2 \sum_{r\in R} \sum_{j\in G'_{\ell(r)}} t_{rj}\, y_{rj} + \lambda_1 \rho_{\ell} \sum_{\ell\in\mathcal{L}} \sum_{i\in G'_{\ell}} \sum_{j\in G'_{\ell}} t_{ij}\, z_{ij}^{\ell} \tag{2.36}$$

Subject to

$$c_{rj} y_{rj} \le w_{max}, \forall r \in R, j \in G'_{\ell(r)} \tag{2.37}$$

$$\sum_{j\in G'_{\ell(r)}} y_{rj} = 1, \forall r \in R \tag{2.38}$$

$$\sum_{r\in R} y_{rj} \le Q_{max}, \forall j \in G'_{\ell(r)} \tag{2.39}$$

$$\sum_{r\in R} y_{rj} \le M\theta_j, \forall j \in G'_{\ell(r)} \tag{2.40}$$

$$z_{ij}^{\ell} \ge \theta_i + \theta_j - 1, \forall i \in G'_{\ell}, j \in G'_{\ell}, \ell \in \mathcal{L} \tag{2.41}$$

$$z_{ij}^{\ell} \in \{0,1\}, \forall i \in G'_{\ell}, j \in G'_{\ell}, \ell \in \mathcal{L} \tag{2.42}$$

$$\theta_j \in \{0,1\}, \forall j \in G'_{\ell(r)}, \forall r \in R \tag{2.43}$$

$$y_{rj} \in \{0,1\}, \forall\, j \in G'_{\ell(r)}, \forall r \in R \tag{2.44}$$

2.3.4 Bus route optimization with charging synchronization

2.3.4.1 Initial solution generation and bus charging scheduling

We randomly generate *n* solutions based on the greedy insertion operator. This operator inserts one customer at a time at the least cost and feasible position of that route by checking the time window, maximum ride time, precedence, and bus capacity constraints using the eight-step evaluation scheme (Cordeau and Laporte, 2003). Afterwards, the energy constraints resulting from that insertion are checked. If violated, charging operations (insertion of visits to chargers) are scheduled.

The charging scheduling algorithm is outlined in Algorithm 2. It begins by identifying energy-feasible positions for recharging (line 2), which are limited to after leaving the depot or a transit station, as recharging is not allowed with customers on board. Direct connections between chargers are also prohibited. For the current route, the algorithm calculates forward slack time (line 5), which determines the available time for recharging without violating route time windows. To avoid charging duration overlaps, a mixed randomization strategy is used: the charging position is randomly chosen from the candidate list (line 3), and a charger is selected based on a greedy strategy, prioritizing the one with the shortest total operation time (including access, egress, and charging time). Given the slack time for

recharging and the desired charging duration, a feasible starting time for recharging is randomly selected within a reduced interval (due to feasibility) without compromising the desired charging duration (lines 7-8 in Algorithm 2). If the bus cannot achieve the desired charge level, another charging stop is added at the next feasible location (line 10). If this fails, the algorithm restarts from the first feasible position (lines 15-16). If no feasible charging insertion is found, the route is deemed infeasible. If the charge schedule is successful, the output includes a sequence of charging operations specifying the insertion position, charger node, and start and end time of recharge. The charging conflict checks with other buses are performed when a solution is promising (line 13). To track charger occupancy, a binary matrix is used, with rows representing chargers and columns representing 10-second intervals. This allows conflict checks to be conducted in $O(1)$ time by referencing the matrix.

Algorithm 2. Bus charging scheduling algorithm

1. **Input**: an energy-infeasible temporary route $r = \{v_0, \dots, v_n\}$.
2. Set $success = false,\ repair = false$. Compute the cumulative energy consumption from the depot to each node of r. Identify the first infeasible node v^* where the state of charge is insufficient to reach that node (i.e., inferior to E^k_{min}).
3. Let $loc_r = \{(v_{i-1}, v_i)\}_{i=1,\dots,p}$ be a list of candidate positions for inserting a recharge operation where a possible recharge visit to a charger location is between v_{i-1} and v_i. Randomly select a position between 1 and p to insert the first charging visit.
4. Choose a charger v_s among the list of chargers with the least charging operation time (i.e., including access, egress, and charging times to connect v_{i-1} and v_i).
   Compute the amount of recharge energy $\Delta_{E_r} = E^k_{min} - (E^k_{init} - ec_r)$, where ec_r is the total energy consumption of r, including this charging visit. Compute the charge time $\delta_{\Delta_{E_r}}$ of Δ_{E_r} given the power of charger v_s.
5. Compute the allowed delays for service start time at v_{i-1} as $\delta^{delay}_{i-1} =$ $\min\{F_{i-1}, \sum_{i-1<p<n} W_p\}$ for all candidate positions, where F_{i-1} and W_i are the forward time slack and the waiting time at node v_{i-1}, respectively (Cordeau and Laporte, 2003).
6. Try inserting v_s after v_{i-1} as follows. Compute $\delta_{access} = t_{v_{i-1},v_s} + t_{v_s,v_i} - t_{v_{i-1},v_i}$.
7. **if** $\delta_{\Delta_{E_r}} + \delta_{access} \le \delta^{delay}_{i-1}$
8. Compute the recharge start time $B_{v_s} := D_{v_{i-1}} + t_{v_{i-1},v_s} + rand(\delta^{delay}_{i-1} - \delta_{\Delta_{E_r}})$, where $D_{v_{i-1}}$ is the departure time, and $rand()$ generates a random real number between 0 and its input. Compute the termination time of recharge $T^{end}_{v_s} = B_{v_s} + \delta_{\Delta_{E_r}}$. Set $success = true$.
9. **else**
10. Compute $B_{v_s} := D_{v_{i-1}} + T_{v_{i-1},v_s}$ and $T^{end}_{v_s} = B_{v_s} + (\delta^{delay}_{i-1} - \delta_{access})$. Update the energy mount to be recharged after v_s. Perform the same charging scheduling procedure for the remaining candidate positions. If successful, set $success = true$.
11. **end**
12. **if** $success = true$
13. **return** the charging schedule of r, i.e., $\{v_{i-1}, v_s, B_{v_s}, T^{end}_{v_s}\}$.
14. **else**
15. **if** $repair = false$
16. Set $repair = true$. Restart the charging schedule from the first candidate position, i.e., $v_{i-1} = v_0$, and go to line 6.
17. **else**
18. **return** $success = false$
19. **end**
20. **end**

2.3.4.2 Local search operators

The local search operators need to be designed by considering their complementarity in diversifying the searched neighborhood from the current solution (Arnold and Sörensen, 2019) and the specificity of the problem at hand. As we allow customer requests to be rejected, unserved customers are managed in a pool, which is regarded as a **virtual route**, allowing customers[3] to be removed from the buses. Given the layered graph structure, we can efficiently screen out infeasible insertion positions by checking whether the layer of a customer to be inserted is (in)compatible with the layer of the current inserted position of the route (based on the determination of compatible layers). This conflict check can be done in $O(1)$ (given constant number of transit stations) based on a lookup table of compatibility information for the layered graph. We propose eight local search operators as follows, where the last one adopts the destroy-repair operators in LNS (Pisinger and Ropke, 2019) to enlarge the neighborhood size.

- **Two-opt***: Two routes are randomly selected. Identify the candidate arcs to be removed from each route, i.e. the load of the bus is zero on these arcs (Parragh et al., 2010). Remove one candidate arc from each route and recombine the first part of the first route with the remaining part of the second route, and vice versa. The feasible one with the best cost improvement is retained.
- **Two-opt**: Reverses the order of visiting the nodes of a segment of the current route, in a sequential manner along the current route. First, the length of the segment is randomly determined between 2 and 4. A node is randomly selected and the end node of the segment can be identified. The feasible one with the best cost improvement is kept.
- **Exchange-segment**: Randomly select two routes. Identify candidate arcs with zero passenger load over the entire current route. The segments between two consecutive candidate arcs of a route are the swappable segments. Exchange two such segments, sequentially along the route, until a feasible and improved route is found (i.e., the bus travel time savings after the exchange is greater than the current threshold T).
- **Exchange-customer**: Swap two customers on two randomly selected routes. The pickup (drop-off) position of the customer on the first route needs to be reinserted at the pickup (drop-off) position of the customer on the second route (Braekers et al., 2014). If successful, the removed customer of the second route is reinserted into the first route at any feasible position. If unsuccessful, the customer is reinserted on another randomly selected route until a feasible insertion is found. This operator is applied sequentially along the routes until an improved and feasible exchange is found.
- **Four-opt**: Remove four successive arcs (three successive nodes) from the current route and find a feasible and improved one among all possible combinations of the removed segment in a sequential search along the route. The feasible and improved one with the least cost is retained.
- **Create**: if the number of used buses is smaller than the fleet size and the pool of unserved customers is not empty, create a new route by randomly inserting an unserved customer on an unused bus.
- **Relocate ensemble**: To avoid repetitive local search operations applied to the same neighborhood, we randomly select a relocation operator from the following two relocation operators: greedy relocation and worst relocation. The two relocation operators search different parts of a solution neighborhood. Greedy relocation randomly removes a customer from their current route and

[3] Note that a customer in the bus route optimization procedure denotes the requests assigned to the same dummy meeting point based on the customer-to-meeting-point assignment (Eqs. (3.36)-(3.46)).

reinserts the customer to the least cost position of the current solution. Worst relocation removes the worst customer (i.e., the most expensive with respect to the objective function) of a randomly selected route and reinserts the customer into the least cost position of the current solution.

- **Destroy-repair operators**: The destroy and repair operators are applied in pairs with each is randomly selected from below. The number $n^{destroy}$ is selected randomly as $1 \leq n^{destroy} \leq \min(n^{max}, \delta|R|)$, where n^{max} is a pre-defined maximum number of customers to remove. δ is a coefficient between 0.2 and 0.5. Based on our preliminary analysis, we set $n^{max} = 60$, and $\delta = 0.275$ (Lutz, 2014). Five destroy operators and two repair operators are used as follows.

  Destroy operators

  - ***Random destroy***: Randomly remove $n^{destroy}$ customers from the candidate list (i.e. all customers on the current solution s).
  - ***Worst removal***: Remove the first $n^{destroy}$ customers with the most contribution to the objective value from the existing served customers.

  For the following three related-destroy operators, the served customers in solution s are sorted based on their relatedness in decreasing order with a randomly selected customer from the unserved customer pool. The first $n^{remove} - 1$ customers in the sorted list are removed. The relatedness value is calculated using Eq.(2.45) – Eq.(2.49), where a lower value indicates a higher relatedness.

  Distance-related destroy: The distance relatedness for customers i and j at their origin and destinations is computed in equation (2.45) (adapted from Lutz, 2014), where c_{max} is the distance of the longest arc in $\mathcal{A}_B$.

$$r_{dist}(i,j) = \frac{c_{ij} + c_{n+i,n+j}}{c_{max}} \tag{2.45}$$

  - ***Time-window-related destroy***: The time-window relatedness for customers i and j is computed in equation (2.46).

$$r_{tw}(i,j) = |\frac{l_i - l_j}{T}| + |\frac{l_{n+i} - l_{n+j}}{T}| \tag{2.46}$$

  - ***Shaw removal***: Shaw relatedness (adapted from Ropke and Pisinger (2006)) integrates the distance-relatedness and time-window relatedness. The Shaw relatedness for customers i and j is computed in equation (2.47), where $\tilde{g}_i$ is the normalized capacity demand of customer i (with respect to $\max\{g_i\}$ for all $i \in G'$ where g_i is the number of passengers associated with i. We set (φ, χ, ψ) as (9, 3, 2) (Ropke and Pisinger, 2006).

$$r_{Shaw}(i,j) = \varphi r_{dist}(i,j) + \chi r_{dist}(i,j) + \psi|\tilde{g}_i - \tilde{g}_j| \tag{2.47}$$

  Repair operators

  Greedy repair: Iteratively select and insert a customer whose best insertion position has lowest cost among the customers in the unserved pool.

  Regret repair: Insert the customers of the unserved pool one by one based on the regret heuristic of Ropke and Pisinger (2006). Note that we implement Regret-2 and Regret-3 heuristics, which are selected randomly to achieve more diverse insertions.

Note that at the end of each local search operator, we update the bus charging schedule by applying the bus charging scheduling procedure (Algorithm 2). If the resulting routes satisfy all the constraints, the updated solution s' is kept (line 12 in Algorithm 1).

2.3.5 Post optimization

If the best solution obtained by the DA algorithm contains unserved customers, the customers on the layers containing unserved customers are reassigned and the sub-routes on the affected layers are reoptimized to satisfy the time windows, maximum ride time, and bus capacity constraints. The reassignment problem is solved by a matheuristic that formulates a conventional MILP problem for reassigning customers on the same layer that contains unserved customers. This matheuristic aims to find improved partial routes for each of such layers that minimize the total bus routing time and the total penalty of unserved customers without violating constraints. A short computation time limit is applied to find an optimal solution. If the solution reduces the initial number of unserved customers on the current layer, the corresponding sub-routes are updated. Finally, the charging schedules of the modified routes are updated and checked for charging conflicts. If the new solution is feasible and has an improved cost, the best solution is updated. The reader is referred to Ma et al. (2024) for the MILP formulation and solution algorithm for reassigning unserved customers.

2.4 Computational experiments

In this section, we present computational experiments for the proposed hybrid metaheuristic algorithm and compare its performance with exact solutions obtained by a state-of-the-art MILP solver. We first present the test instances and tune the algorithmic parameters. Then, the performance of the hybrid metaheuristic is evaluated, including analysis of bus charging scheduling. The performance of the layered graph is analyzed at the end of this section.

2.4.1 Test instance generation

To get a nuanced performance of the algorithm, we consider two scenarios corresponding to peak (P) and off-peak (OP) demand profiles. Scenario P simulates a peak-hour situation where customers' desired arrival times at transit stations are concentrated around 6:00-8:00, whereas scenario OP assumes a uniform distribution of arrival times over the operating period. 10 instances with the number of customers ranging from 10 to 100 customers are generated for each scenario. These test instances have a single bus depot, two train stations, and four chargers as shown in Figure 2.3. Half of the customers choose station 1 as their drop-off point, while the other half choose station 2. Meeting points are generated as a grid with a separation distance of 1 km, and the maximum walking distance of customers is 1.5 km. The two train stations offer punctuated services with three departures per hour, starting at 6:00 (6:10) and ending at 10:00 (9:50), resulting in 26 layers. In total, there are 26 layers with 25 to 49 activated meeting points per layer (meeting points within the maximum walking distance of the customers). Each customer can access up to 7 meeting points within their walking distance (Figure 2.3). Consequently, the possible customer-to-meeting-point assignment combinations are very large, providing non-trivial experiments to test the performance of the algorithm.

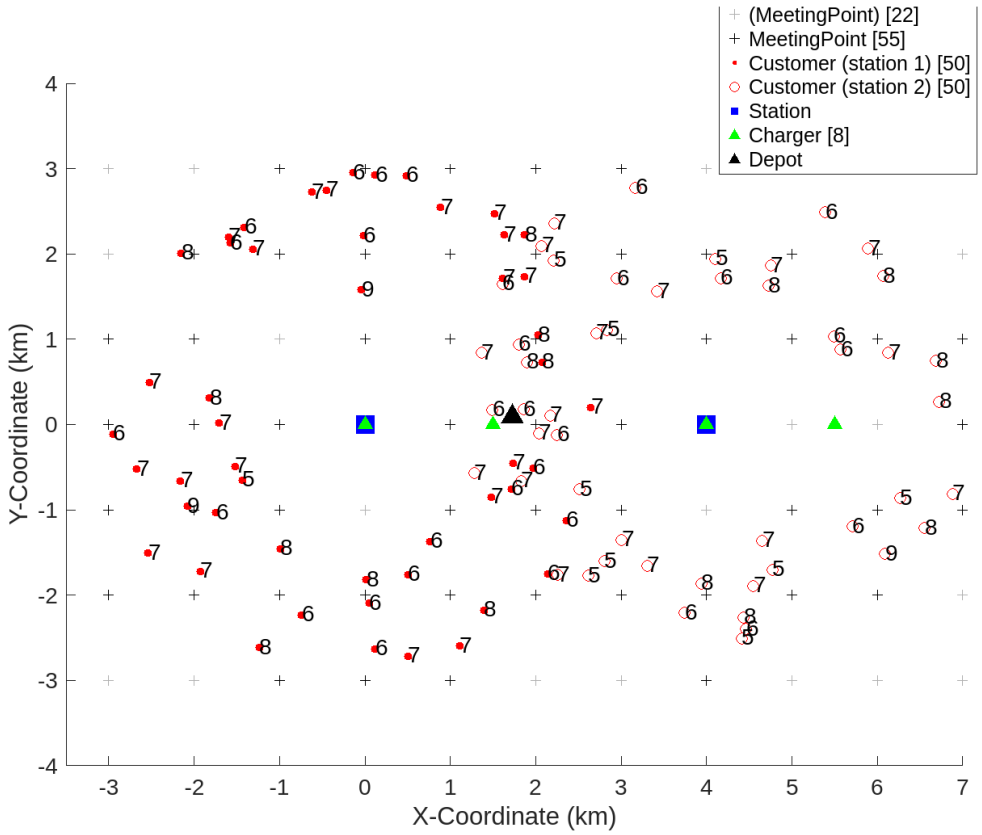


Figure 2.3: Example of a test instance with 100 customers[4].

We consider two types of buses, each with distinct passenger capacities, battery capacities, and energy consumption rates. Instances are labeled as cxx, where xx indicates the number of customers in the instance. In order to force bus charging operations, initial SoCs of buses are set as low as 20%, 30%,…, and 80% of the battery capacity. The number of buses is determined by considering approximately 70% of bus occupancy. The average initial battery state for both scenarios is 26.4%. We assume that, when returning at the depot, the minimum SoC is 10% of its capacity. An overview of the experiment is shown in Table 2.1.

[4] Number next to each customer shows desired transit departure. The greyed out (meeting point) are beyond the maximum walking distance of any customer.

Table 2.1: Overview of the experiment and algorithmic parameter settings

Parameter	Value	Parameter	Value
Number of type of buses	2	Charging power	0.83kWh/minute
Number of meeting points within customer's walking distance	25 to 42	E_{init}^{k}	20%, 30%,…,80% of $\bar{B}_k$
Number of punctuated service per transit station	12 or 13	Energy consumption rate of buses	0.24 kWh/km and 0.29 kWh/km
Number of transit stations	2	Number of customers	10,20,…,100
Number of chargers	4	Walking speed	0.085 km/minute
Number of buses	2 to 6	Detour factor φ	1.5
Passenger capacity of the buses	10 or 20	λ	1
Bus speed	0.83 km/min	ω (penalty of one unser customer)*	40
Battery capacity of buses ($\bar{B}_k$)	35.78 kWh and 53.70 kWh	Maximum walking distance of customers	1.5 km
$E_{max}^{k}(E_{min}^{k})$	$0.8\bar{B}_k$ ($0.1\bar{B}_k$)	u_i(service time)	0.5 minutes

*based on our preliminary experiments. The characteristics of electric buses is adapted from Volkswagen's 8-seat 100% electric Tribus with 35.8 kWh (https://www.tribus-group.com/zero-emission-volkswagen-e-crafter-electric-wheelchair-minibus/)

2.4.2 Algorithm parameter settings

The parameters used in Algorithm 1 include the customer-to-meeting-point assignment weight ρ, and several parameters used for the DA algorithm. We first present the sensitivity analysis of the parameters of the DA algorithm. Following Braekers et al. (2014), the parameters of the DA algorithm and the associated discrete values to be tested are listed as follows.

- t_{max}: A user-defined coefficient to determine the maximum threshold value T_{max} to accept worsen temporary solutions, i.e. $T_{max} = t_{max}\, T_{allbus}$ where T_{allbus} is the average bus travel time between all pick-up and drop-off nodes in the layered graph of a problem instance. 8 values of t_{max} are tested, i.e. $t_{max} \in \{0.3, 0.6, 0.9, 1.2, 1.5, 1.8\}$.
- T_{red} : Threshold reduction factor for reducing the threshold value ($T \coloneqq T - \frac{T_{max}}{T_{red}}$) (line 20 in Algorithm 1). Six values of T_{red} are tested, i.e. $T_{red} \in \{100, 200, 300, 400, 500, 1000\}$.
- n_{imp}: Restart parameter, set as $n_{imp} \in \{100, 200, 300, 400, 500, 600\}$.
- $Iter_{max}$: Maximum number of iterations (in thousands), set as $Iter_{max} \in \{25, 50, 100, 150, 200, 300, 400\}$.
- $n_{stagnant}$: Maximum stagnation multiplier, set as $n_{stagnant} \in$ {25,50,100,150,200,250, 300}

We generate 10 random test instances with random numbers of requests between 20 and 100 (see Table 2.2). We solve each instance 5 times with a constant ρ to get the average performance for each tested parameter setting. It takes a few days to complete the experiments on a high-performance machine.

Table 2.2: Best values of ρ on the test instances

Instance	c24	c28	c30	c34	c40	c48	c52	c72	c82	c88
Number of buses	2	2	3	3	3	3	3	5	5	6
Average E_{init}^{k} (% of $\bar{B}_k$)	20	20	23	23	23	23	23	28	28	32
Best ρ	0.7	1.2	0.7	0.1	0.6	1.1	0.2	0.4	0.5	0.1

Note: ρ is the same for each layer (constraint. (2.36)).

To evaluate the performance of a solution, a gap $(Z - Z^*)/Z^*$ is calculated with respect to the best solution found for each instance. Based on our preliminary analysis, t_{max}, T_{red} and n_{imp} are initialized as 0.6, 500 and 200, respectively. We first tune $Iter_{max}$ without considering $n_{stagnant}$. Given the tuned $Iter_{max}$, we vary $n_{stagnant}$ to analyze its sensitivity and a best value. Since the effects of t_{max}, T_{red}, and n_{imp} on the performance of the algorithm are not independent (Braekers et al., 2014), the best parameter setting needs to consider all possible combinations of the tested values of these parameters. Figure 2.4 (a) shows the results of $Iter_{max}$ on the average gaps over the 10 test instances, all other parameters remain unchanged. The best $Iter_{max}$ is found at 300k iterations with an average gap of 0.06%. However, the computation time is proportional to $Iter_{max}$. For the remaining experiments, we decide to use $Iter_{max} = 100k$ which is a good trade-off between computation time and solution quality (gap=0.17%). In Figure 2.4 (b), the best $n_{stagnant}$ is found at 200 with the average gap of less than 0.15%. Increasing $n_{stagnant}$ further to 300 significantly increases the computation time with little improvement in the gap value. The sensitivity analysis of t_{max}, T_{red}, and n_{imp} is shown in Table 2.3. We can observe that in general the gap increases with t_{max}, T_{red}, and n_{imp}. From the result of 512 combinations, the minimum gap 0.13% is found with $t_{max} = 2.1$, $T_{red} = 200$ and $n_{imp} = 100$. Therefore, we will use these values with $n_{iter} = 100$k and $n_{stagnant} = 200$ for the computational studies in the following sections.

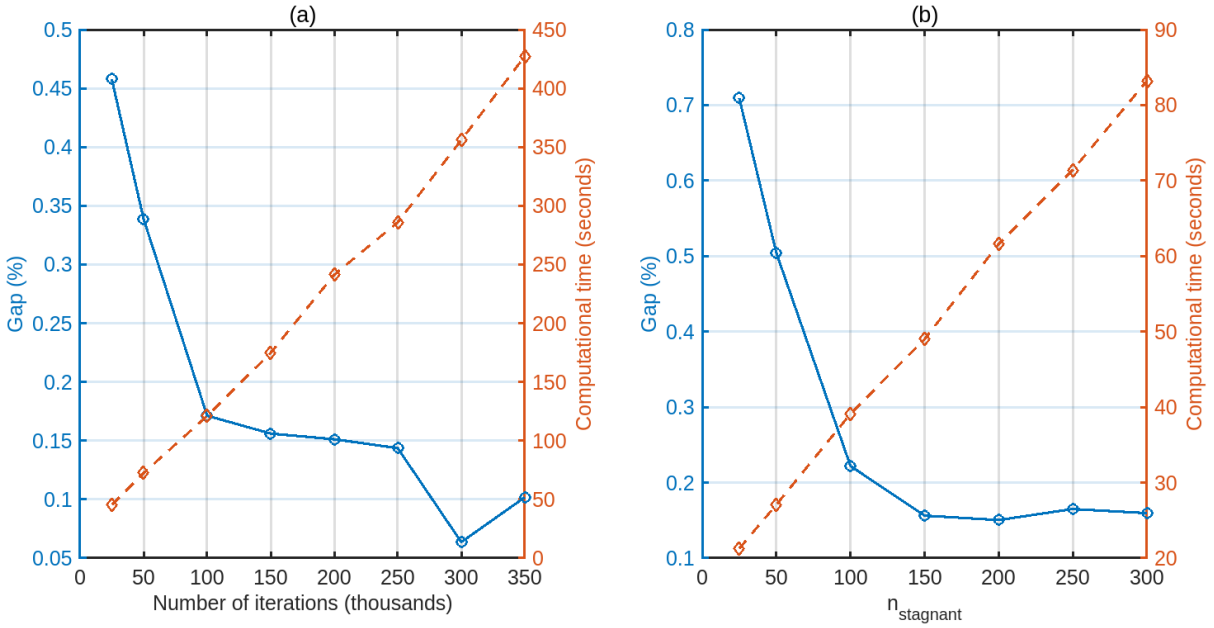


Figure 2.4: Impact of $Iter_{max}$ (a) and $n_{stagnant}$ (b) on the performance of the hybrid metaheuristic

Table 2.3: Sensitivity analysis on $t_{max}, T_{red}, n_{imp}$

t_{max}	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4
Avg. gap	0.54%	0.61%	0.60%	0.64%	0.62%	0.67%	0.69%	0.67%
T_{red}	100	200	300	400	500	1000	2000	3000
Avg. gap	0.58%	0.61%	0.62%	0.61%	0.61%	0.63%	0.68%	0.69%
n_{imp}	100	200	300	400	500	600	1000	2000
Avg. gap.	0.36%	0.49%	0.56%	0.65%	0.66%	0.68%	0.79%	0.84%

Regarding the impact of $\boldsymbol{\rho} = \{\rho_\ell\}_{\ell \in \mathcal{L}}$, it depends on the characteristics of each test case. Different factors such as demand intensity, spatial distribution of requests, maximum walking distance of customers, number of buses, capacity of buses, location of meeting points and transit stations may affect the best value of $\boldsymbol{\rho}$ for each test instance. Figure 2.5 shows the effect of $\boldsymbol{\rho}$ on the performance of the algorithm on three test instances with 28, 52, and 88 customers. It can be observed that the relationship between ρ and the algorithm performance is not unimodal. Thus, the classical golden section search method is not relevant to find the best value of ρ. The best $\boldsymbol{\rho}$ tends to be smaller (0.2) for larger instances to minimize the customer walking time in Eq. (2.36). We conduct a systematic ρ search experiment over the above 10 test instances using a 2-step approach as follows. First, set 10 test values of ρ (i.e. $\rho_\ell = \rho$ for all $\ell \in \mathcal{L}$) as $\rho \in \mathrm{P} = \{0.2, 0.4, .., 2.0\}$. Get the preliminary best $\tilde{\rho}$ (best solution found) over P for each test instance. Then run two tests to get the best ρ^* on the neighborhood of $\tilde{\rho}$, i.e. $\tilde{\rho} \pm 0.1$. For instances with unserved users, we tune the layer-specific ρ_ℓ with 5 additional runs on a second step. Initially set $\rho^0 = \rho^*$ for all $\ell \in \mathcal{L}$. For the subsequent iterations, increase the value of ρ_ℓ for $\ell \in \bar{\mathcal{L}}$ with δ (e.g. 1.5), where $\bar{\mathcal{L}}$ is the subset of layers with non-empty unserved users. We retain $\{\rho_\ell^*\}_{\ell \in \mathcal{L}}$ of the best of the 5 runs. It can be observed that for instances with more than 50 customers, the best $\rho^*$is no larger than 0.5, while for small instances, the best ρ^* can be larger than 1. This suggests that a narrowed search range between 0 and 0.5 is sufficient to find good ρ^* for the cases with more than 50 requests. In practice, it is not a problem to find the best ρ^* with more test values for small instances since the computation time is in the order of seconds. On the other hand, for larger instances, a budget of 5 runs can be allocated to find the best solution using 5 test values of ρ, i.e. $\rho \in \{0.2, 0.4, 0.6\}$ with two additional test values on the neighborhood (± 0.1) of $\tilde{\rho}$. If there are unserved users, additional 3 runs are necessary to adjust ρ for layers with unserved users.

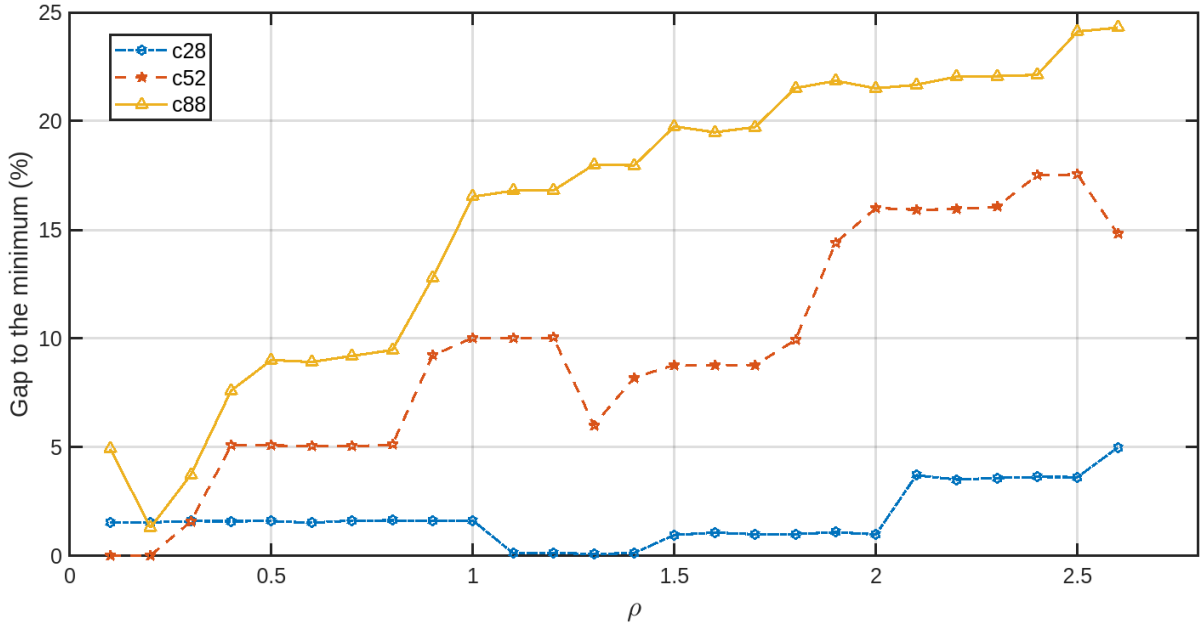


Figure 2.5: Impact of ρ on the performance of the hybrid metaheuristic algorithm

2.4.3 Computational results

We investigate the performance of the algorithm on the 20 test instances in P and OP scenarios using the tuned parameters for the DA algorithm. The algorithm performance is compared with the solutions obtained by the state-of-the-art MILP solver (Gurobi, version 10) with a computation time limit of 4 hours. Our algorithm and the MILP model are both implemented using the *Julia* programming language. The experiments are conducted on a laptop with an 11th generation Intel(R) Core(TM) i7-11800H, 16 logical processors and 64 GB of memory.

The results of MILP formulation and the hybrid metaheuristic are presented in Table 2.4. The Best Known Solution (BKS) in the MILP results is the incumbent solution after 4-hour CPU time. For the hybrid metaheuristic, the results are based on five runs. The best objective function value (Best obj.) represents the minimum objective function value over five runs. Gap avg. and Gap best are calculated as the gap between the average and minimum objective function value over five runs and the BKS, respectively. The results show the efficiency and good solution quality obtained by the proposed algorithm. Overall, the gaps for the 20 test instances are 1.80% (Gap avg.) and 1.69% (Gap best), with an average CPU time of only 55 seconds. The OP scenario shows a smaller average gap compared to the P scenario but requires longer CPU time due to the larger problem size caused by more activated layers. All customers are served in the OP scenario, while one customer is rejected in instances c30p and c50p. Both the MILP and hybrid metaheuristic results indicate less charging time in the OP scenario than in the P scenario. Note that the worst gap is found for the c50p in both P and OP scenarios. This also appears to be a difficult instance for the MILP in terms of its BKS and gap to the lower bound. These results indicate that for some difficult instances, more computational effort needed to explore other solutions involving other potential customer-to-meeting-point assignment alternatives.

Table 2.4: Computational results of the hybrid metaheuristic and MILP

Instance	MILP				Hybrid metaheuristic					
	Best know solution	Gap to lower bound	UC*	CT* (min)	Best obj.	Gap** (avg.)	Gap** (best)	UC*	CT* (min)	CPU (s)
c10op	107.91	4.29%	0	4.9	107.91	0.00%	0.00%	0	4.9	8
c20op	232.57	15.14%	0	17.9	234.15	0.69%	0.68%	0	18.2	12
c30op	327.76	13.90%	0	15.4	334.26	2.10%	1.98%	0	16.5	28
c40op	413.96	23.49%	0	29.2	416.88	0.77%	0.71%	0	31.1	26
c50op	566.74	31.74%	0	44.1	586.80	3.69%	3.54%	0	42.4	30
c60op	596.35	24.05%	0	22.8	602.63	1.24%	1.05%	0	26.1	84
c70op	702.64	25.09%	0	32.7	694.31	-0.83%	-1.18%	0	33.4	76
c80op	842.29	32.13%	0	40.0	815.56	-2.82%	-3.17%	0	43.1	85
c90op	816.47	29.05%	0	25.1	816.17	0.30%	-0.04%	0	31.5	149
c100op	967.48	27.79%	0	33.2	980.16	1.56%	1.31%	0	27.2	142
Average	557.41	22.67%	0	26.5	558.88	0.67%	0.49%	0	27.4	64
c10p	112.02	16.81%	0	2.8	112.55	0.47%	0.47%	0	4.9	6
c20p	283.40	40.72%	1	13.5	311.26	9.84%	9.83%	0	12.6	10
c30p	341.11	33.48%	1	10.7	361.15	5.88%	5.88%	1	8.9	76
c40p	421.79	39.10%	1	15.7	441.80	4.74%	4.74%	0	13.2	15
c50p	606.48	46.82%	1	31.9	674.19	11.19%	11.16%	1	27.3	66
c60p	584.07	34.47%	0	11.6	628.98	7.86%	7.69%	0	3.6	47
c70p	684.36	34.71%	1	7.7	689.05	0.75%	0.69%	0	11.4	45
c80p	872.50	43.21%	1	20.7	831.52	-4.65%	-4.70%	0	8.6	51
c90p	811.70	38.11%	0	6.9	761.15	-6.22%	-6.23%	0	18.7	85
c100p	943.47	34.25%	0	14.0	937.87	-0.53%	-0.59%	0	14.8	61
Average	566.09	36.17%	0.60	13.5	574.95	2.93%	2.89%	0.2	12.4	46
Overall average	561.75	29.42%	0.30	20.04	566.92	1.80%	1.69%	0.1	19.9	55

*UC: Number of unserved customers; CT: charging time

** Gap to the best solutions found by Gurobi at a computation time of 4 hours. Charging time of the obtained solutions is measured in minutes. CPU time is measured in seconds.

To test the performance of the post-optimization process, Table 2.5 compares the computational results with and without applying post-optimization. We only report the results where there are unserved customers without applying post-optimization to verify the effectiveness of this post-optimization. Note that Table 2.5 repots the average objective values based on 5 runs of the hybrid metaheuristic. Post-optimization is shown to both improve the solution quality (2/3) and reduce the number of unserved customers to half or zero. The average gap is significantly improved from 12.96% to 8.97%. In terms of computation time, the average time for the algorithm with post-optimization is significantly higher on average (+40 seconds on average), depending on the problem size of the MILP formulation for the post-optimization.

Table 2.5: Computational results for the post-optimization procedure of the hybrid metaheuristic

Instance	Algorithm **without** the post-optimization				Algorithm **with** the post-optimization			
	Avg. obj.	Gap*	# unserved customers	CPU (s)	Avg. obj.	Gap*	# unserved customers	CPU (s)
c20p	335.03	18.22%	3.0	6	311.29	9.84%	0.0	10
c30p	361.19	5.89%	1.0	14	361.16	5.88%	1.0	76
c50p	696.09	14.78%	2.0	14	674.34	11.19%	1.0	66
Average**	464.11	12.96%	2.0	11	448.93	8.97%	0.7	51

* Gap to the best solutions found by Gurobi with 4 hours of computation time

**Average over the instances with unserved customersS

To analyze the impact of charging operations on the hybrid metaheuristic, we conducted an experiment with a high (80%) initial SoC for all buses. Table 2.6 compares the algorithm's performance with high and low initial SoC levels across the 20 instances. The results for the low initial SoC are referenced from Table 2.4. No charging operations are observed when buses are 80% charged at the beginning of the service. With a high initial SoC, the CPU time for the hybrid metaheuristic in the OP scenario is lower than in the P scenario, which contrasts with the results for the low-SoC case. This may be because no customers are rejected in the high-SoC case for either the P or OP scenarios, whereas two instances in the low-SoC case rejected one customer in the P scenario. Additionally, the computation time for the high-SoC case averages less than 17 seconds, significantly lower than the average of 55 seconds for the low-SoC case.

Table 2.6: Impact of initial battery level of buses on the performance of the hybrid metaheuristic

Avg. E_{init}^{k}	MILP				Hybrid metaheuristic			
	Scenario	Avg. BKS	Avg. gap to the lower bound	Charging time (min)	Avg. obj. value	Avg. gap	Avg. gap	CPU (s)
Low*	Off-peak	557.41	22.67%	26.5	560.20	0.67%	0.49%	64
	Peak	566.09	36.17%	13.5	575.22	2.93%	2.89%	46
High**	Off-peak	549.21	19.34%	0.0	532.23	-1.70%	-1.81%	9
	Peak	573.78	35.88%	0.0	562.68	1.46%	1.46%	24

* E_{init}^{k} is set as 20%, 30%,… with an overall average of 26.4%.

** $E_{init}^{k} = 80\%$ for all buses. BKS is the best solution solved by Gurobi in 4 hours. The result of the hybrid metaheuristic is based on the average of 5 runs. Both sets of experiments use the same algorithmic parameters except ρ. Average gap is calculated as the incumbent solution found by Gurobi after 4 hours.

2.4.4 The performance of the layered graph

To demonstrate the effectiveness of the layered graph structure, we evaluate the performance of the hybrid metaheuristic algorithm with and without using this structure for the customer-to-meeting-point assignment. The MILP formulation for the assignment problem without the layered graph structure is presented below.

$$\text{Min } \lambda_2 \sum_{r\in R} \sum_{j\in G'} t_{rj}\, y_{rj} + \lambda_1 \rho \sum_{i\in G'} \sum_{j\in G'} t_{ij} z_{ij} \tag{2.48}$$

Subject to

$$c_{rj} y_{rj} \leq w_{max}, \forall r \in R, j \in G' \quad (2.49)$$

$$\sum_{j \in G'} y_{rj} = 1, \forall r \in R \quad (2.50)$$

$$\sum_{r \in R} y_{rj} \leq Q_{max}, \forall j \in G' \quad (2.51)$$

$$\sum_{r \in R} y_{rj} \leq M\theta_j, \forall j \in G' \quad (2.52)$$

$$z_{ij} \geq \theta_i + \theta_j - 1, \forall i \in G', j \in G' \quad (2.53)$$

$$z_{ij} \in \{0,1\}, \forall i \in G', j \in G' \quad (2.54)$$

$$\theta_j \in \{0,1\}, \forall j \in G' \quad (2.55)$$

$$y_{rj} \in \{0,1\}, \forall\, j \in G', \forall r \in R \quad (2.56)$$

For this analysis, we generate 18 new test instances with the number of customers ranging from 10 to 1600 for both P and OP scenarios. The maximum walking distance is set to 1.0 km, and the meeting-point separation distance is 1.2 km. The two MILP models, with (Eq. (2.36) – Eq.(2.46)) and without (Eq. (2.48) – Eq. (2.56)) layered graph are solved to optimality by Gurobi.

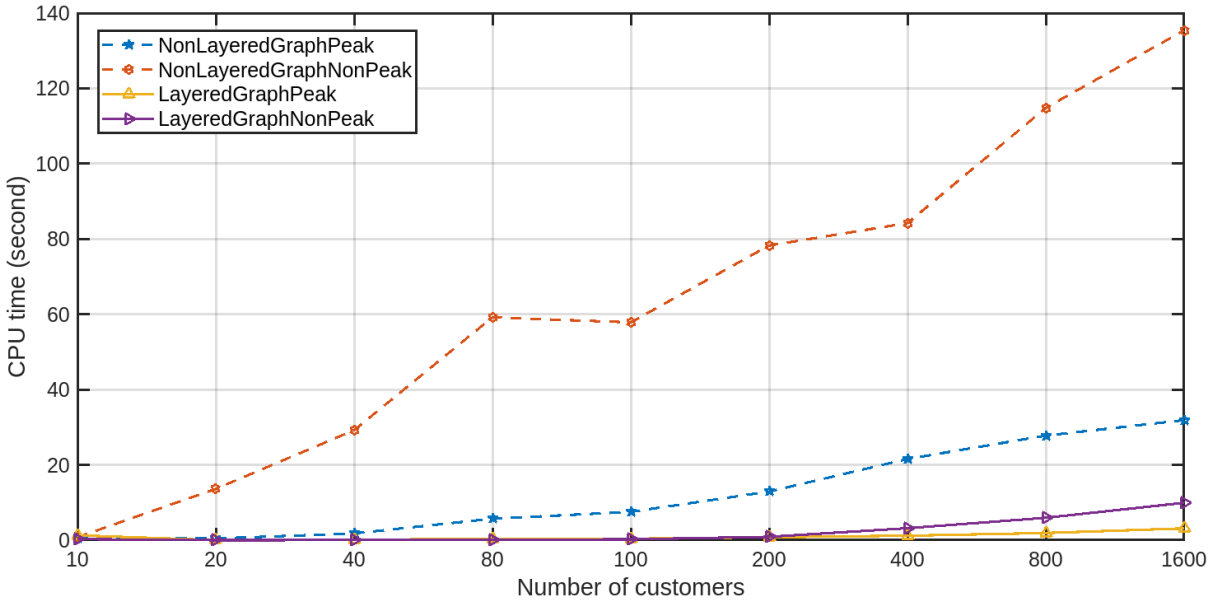


Figure 2.6: CPU time of the customer-to-meeting-point assignment models with and without using the layered graph

Figure 2.6 presents the results of the CPU time of the two approaches. As the number of customers increases, the CPU time for the model using the layered graph structure shows only a minor increase. For instances with 1600 customers, the CPU time remains under 10 seconds for both P and OP scenarios. In contrast, the CPU time for the model without the layered graph structure increases significantly, reaching 135 seconds for the 1600-customer instance in the OP scenario. These differences in CPU time clearly illustrate the effectiveness of the layered graph structure in enhancing the computational efficiency of the hybrid metaheuristic algorithm.

2.5 Discussion and conclusions

In this chapter, we addressed the problem of a meeting-point-based first-mile electric DRT service with realistic capacitated charging stations. We proposed a MILP formulation based on the layered graph structure and a hybrid DA-based algorithm framework that combines randomization and greedy strategies to solve this problem efficiently. The originality of this problem lies in the joint optimization of customer-to-meeting-point assignment, electric bus routing and charging scheduling. The layered graph structure enables elimination of unnecessary nodes and arcs to reduce the problem size. The DA algorithm is customized with two randomization methods, namely relocate ensemble and destroy and repair, to improve the exploration of the solution space. Moreover, for the customer-to-meeting-point assignment, we improved flexibility by allowing layer-specific ρ to balance users' walking time and bus travelling time. This approach enables the algorithm to adapt more effectively to demand intensity variations, especially when unserved customers exist in certain layers.

The algorithm was tested on 20 test instances with up to 100 customers and 49 meeting points under different initial battery levels and demand distributions (peak and off-peak scenarios). The results demonstrate that the proposed algorithm can efficiently solve the problem with good solution quality. When compared to an exact MILP solver with 4-hour time limit, the hybrid metaheuristic achieved an average and best gap of 1.80% and 1.69% compared to the 4-hour Gurobi solution, respectively, within an average computational time of less than 1 minute. The effectiveness of layered graph was also examined. The results show that implementing the layered graph significantly reduces the computational time for the customer-to-meeting-point assignment. When comparing scenarios with high initial SoC and low initial SoC of buses, the low-SoC cases require approximately twice the computational time of the high-SoC cases. This indicates that improving the efficiency of bus charging operations can lead to substantial computational savings.

The methodologies applied in MP-EFCS provide valuable insights for studying the entire EIDRT service. First, the departure-expanded approach, currently used only for first-mile services to synchronize with transit departures, can be extended to both first- and last-mile services in EIDRT. Second, the management of capacitated charging stations is integrated into both the MILP formulation and the metaheuristic for MP-EFCS. Further improvements in the efficiency and quality of these approaches can be explored in the following chapters for the EIDRT service.

Chapter 3 The electric integrated demand-responsive transport service

This chapter is based on the following working paper: *Fang, Y., Ma, T.-Y., Viti, F., 2024. Electric integrated dial-a-ride services with capacitated charging stations, multiple depots, and customer rejections.(under review for Public Transport)*

In Chapter 2, we developed a meeting-point-based first-mile DRT services using EVs. This service enables customers to reach their desired transit stations while ensuring their maximum waiting times. Building on this foundation, this chapter explores an EIDRT service that expands the scope by addressing many-to-many DRT service with all customer OD (many-to-many) trips. Furthermore, the service integrates the entire transit network within the service area to enhance efficiency while minimizing customer's inconvenience. The proposed system utilizes electric buses with capacitated charging stations, ensuring sustainability and operational feasibility.

This chapter focuses on modeling the EIDRT service, which is referred to as Electric Integrated Dial-A-Ride Problem (EIDARP). We present a novel MILP model for the EIDARP with capacitated charging stations. The proposed approach extends the existing IDARP by allowing customer rejections with multiple depots and coordinating on-demand buses' arrivals at transit stations to minimize customers' journey time on a timetabled transit network. Charging station capacitated constraints (CSCCs) are considered at each charger to avoid vehicles' charging conflicts. Different from previous work (Posada et al., 2017), we integrate timetabled transit network using a (scheduled) departure-expanded graph approach introduced in Chapter 2 to reduce redundant decision variables of the IDARP. A set of valid inequalities is proposed to strengthen the model, and this approach is tested with up to 50 requests, 3 transit lines and 13 transit stations. Results show that our model formulation can solve the problem optimally much faster and to a larger problem size compared to Posada et al. (2017). Moreover, compared to Chapter 2, we propose a more compact formulation to include capacitated charging station constraints. The numerical results show significant computational time savings when using our approach compared to existing charging station/charger replication-based approaches (Bruglieri et al., 2019; Ma et al., 2024).

The rest of this chapter is organized as follows. Section 3.1 reviews related studies with the focus on the modelling approach of integrated DRT. Section 3.2 presents the MILP formulation of EIDARP. Time window tightening, arc elimination, and valid inequalities are proposed for model strengthening in Section 3.3. Section 3.4 presents the computational studies on a set of test instances with alternative model formulation. The impact of key model parameters on the system performance is analyzed. Section 3.5 discusses the results and concludes this chapter.

3.1 Literature review

IDRT services have received increasing interest in recent years to improve the performance of fixed-route based public transport services and reduce the VKT of on-demand mobility services. Häll et al., (2009) were the first to introduce the IDRT service as IDARP by combining dial-a-ride services with fixed route transit to provide door-to-door multimodal trips. They formulated the IDARP by using a

MILP approach, assuming a high frequency of transit lines while neglecting customers' waiting time at transit stations. In this problem, requests can be served by a combination of on-demand buses and transit lines, where users' riding times are constrained to not exceed predefined maximum values. Besides, time window constraints are applied to users' origins and destinations, with the objective of minimizing total vehicle operational costs. Melis et al. (2024) proposed a MILP formulation for integrated on-demand bus services for minimizing total user riding times. Unlike the door-to-door service offered in IDARP, customers are required to walk to nearby bus stops, and the customer-to-bus stop assignment is solved alongside the bus routing problem. While the objective function focuses on minimizing total customer's travel time, no explicit constraints are placed on maximum customer travel time. Similar to the IDARP, the considered problem also assumes high-frequent transit service and neglects transfer synchronization and inter-modal transfer times.

However, in areas with low transit frequency, inter-modal transfer time at transit stations is crucial, as previous studies have highlighted transfer inconvenience (e.g., prolonged transfer times) as a significant factor affecting users' willingness to use integrated public transport systems (Chowdhury and Ceder, 2016; Gkiotsalitis, 2022). Kim and Schonfeld (2014) studied the coordination of fixed-route and flexible buses services under time-varying demand and assessed the system cost savings. They found that coordinating bus arrivals at transfer stations can significantly reduce passengers' waiting time and enhance the system's efficiency. These findings suggest that better synchronization at transfer stations is necessary to reduce users' transfer time and increase the reliability of the IDRT service.

In early studies, Liaw et al. (1996) considered the IDRT as a bimodal DARP, where DRT vehicles are coordinated with departures of fixed-route buses. Requests are characterized by their origins, destinations, and desired passengers' delivery time constraints that cannot be violated. Maximum allowable excess riding time and maximum allowable deviation from the desired delivery time are used to set up the time window constraints associated with origins and bus stops to synchronize DRT vehicles' arrival times with scheduled bus departures. Bus lines are not explicitly modeled, but instead associating a set of departures on bus stops for determining used bus departures for users' bimodal routes. The objective is to minimize total travel costs of vehicles. Posada et al. (2017) extended the model of Häll et al. (2009) by considering timetabled transit services allowing users to walk to/from transit stations. The arrival times of DRT vehicles at transit stations are coordinated with transit timetables to minimize the overall routing cost to serve customers' requests. The transport modes used for each leg of users' trips, vehicle routes and their connected transit stops are jointly optimized. Dragomir and Doerner (2020) addressed an inter-modal Pickup and Delivery Problem (PDP) between two cities. Long-haul (fixed) services operate between two cities with local short-haul (flexible) services running locally in each city. The authors split the problem into two sub-problems: long-haul assignment and short-haul routing. The first sub-problem decides which train departure of the long-haul service is used and their associated requests, while the second sub-problem deals with the routing optimization of short-haul pick-up and delivery for each city. Vehicles' arrivals at train stations are synchronized with scheduled departures of trains. In the field of logistics, the transfer synchronization is also considered. Ghilas et al., (2016) proposed a MILP formulation including transfer synchronization for a PDP with scheduled lines. DRT vehicles drop off/pick up requests at transit stations to meet arrival/departure times of vehicles of the scheduled lines.

Although these studies incorporate transfer synchronization, the constraints are typically designed to ensure that on-demand vehicles arrive at transfer points before the departure of fixed-route transit or that they depart only after the arrival of fixed-route transit when transfer requests exist. However, this

approach may result in increasing customer waiting times, as the constraints focus solely on ensuring mode compatibility without explicitly minimizing the waiting time at transfer stations.

Moreover, modelling transfer stations is a critical aspect of IDRT problems. Many studies address this issue by duplicating the transfer node based on the number of requests. For instance, Häll et al. (2009) duplicated each transfer nodes for every customer, leading to a dramatic increase in problem complexity as customer numbers grow. Their MILP formulation could be solved exactly using a commercial MILP solver with up to four customers and one transit line. Referring to the method of Häll et al. (2009), Ghilas et al. (2016) also replicated each transfer node for each customer, but they limited that each node is dedicated for each request. Similarly, Dragomir and Doerner (2020) introduced a dummy node for each customer at transfer stations if they transfer between short-haul and long-haul services. Posada et al. (2017) proposed a more efficient modelling approach by replicating the transfer stations based on the number of transit departures. In their model, the transit network is represented as a set of transit stations, with pairwise arcs connecting the duplicated (dummy) nodes of transit stations. The authors demonstrated that this method significantly reduced computational time compared to the customer-based replication method. However, the approach still requires a decision variable to determine the assignment between transit departures and the duplicated nodes.

Table 3.1: Related studies of IDRT modelling.

	Liaw et al. (1996)	Häll et al. (2009)	Ghilas et al. (2016)	Posada et al. (2017)	Melis et al. (2024)	This study
Objective function (minimize)	a	a	ab	ab	c	ac
Heterogeneous vehicles			✓	✓		✓
Multiple depots			✓			✓
Scheduled transit services	✓		✓	✓		✓
Maximum travel time/delivery time constraints	✓	✓		✓		✓
Maximum walking time constraints					✓	✓
Maximum inter-modal transfer time constraints	✓					✓
Door-to-door services	✓	✓	✓	✓		✓
Customer rejection	✓					✓
Electric vehicles						✓
Capacitated charging stations						✓

Notes: a. travel time (costs) of DRT vehicles; b. travel time (costs) of transit line; c. customers' journey/riding time.

Table 3.1 summarizes the characteristics of related studies for the IDRT. The new features of the proposed EIDARP include: (i) considering the trade-off of the operator's operating costs and customer's inconvenience in the objective function; (ii) reducing customer inconvenience by constraining

maximum intermodal transfer times and maximum walking time to/from transit stations; (iii) involving EVs and capacitated charging stations with partial recharge; (iv) considering multiple depots for on-demand buses and customer rejection. Moreover, we introduce a departure-expanded graph to efficiently model the inter-modal transfers.

3.2 Mathematical formulation

In this section, we first provide the problem description of the integrated dial-a-ride service. We then explain our modeling approach based on the departure-expanded transit network. The problem is formulated as a MILP by considering electric vehicle energy consumption and charging station capacity constraints at individual charger levels.

Notation

Sets	
K	Set of electric buses
$O, \bar{O}$	Set of origin and destination depots
P	Set of customers' origins, i.e. $P = \{1, \ldots, n\}$
R	Set of customers, i.e., $R = \{1, \ldots, n\}$
D	Set of customers' destinations, i.e., $D = \{n+1, \ldots, 2n\}$
$\mathcal{L}$	Set of transit lines
F	Set of physical transit stations, i.e., $F = \{F_l\}_{l \in \mathcal{L}}$
$\mathcal{D}$	Set of train departures, i.e., $\mathcal{D} = \{\mathcal{D}_l\}_{l \in \mathcal{L}}$
G	Set of transit (dummy) nodes, i.e., $G = \{g_d^f\}_{f \in F_l, d \in \mathcal{D}_l, l \in \mathcal{L}}$
C	Set of charging stations
S	Set of dummy charger nodes, i.e., $S = \{S_c\}_{c \in C}$
N	Set of all nodes. $N = P \cup D \cup G \cup S \cup \{0, \bar{O}\}$
A_B	Set of bus arcs
A_G	Set of transit arcs
A_R	Set of walking arcs between origins/destinations and transit nodes within a constant maximum walking distance
Parameters	
n	Number of customers
t_{ij}	Travel time from bus node i to bus node j, $\forall (i,j) \in A_B$
t_{ij}^w	Walking time from node i to node j, $\forall (i,j) \in A_R$
t_{ij}^m	Travel time from transit node i to j, $\forall (i,j) \in A_G$
c_{ij}	Distance from node i to node j
u_i	Service time at node i
e_i, l_i	Earliest and latest starting times of service at bus node i
$\bar{\theta}_i$, $\underline{\theta_i}$	Arrival and departure time of transit vehicle (train) at node i, $i \in G$
γ	Maximum transfer time for inter-modal transfer
φ	Detour factor
Q^k	Capacity of bus k
β^k	Energy consumption rate per kilometer traveled for bus k
α_s	Charging speed of charger s
$E_{min}^k, E_{max}^k, E_{init}^k$	Minimum, maximum, and initial SoC of bus k

L_r^{max}	Maximum travel time for customer r. $L_r^{max} = t_{r,n+r} \times \varphi$
t_{end}	End of service operation duration
$\lambda_1, \lambda_2, \lambda_3$	Weighted coefficient in the objective function
ω	Penalty associated with one customer rejection
M	Large positive number
Auxiliary variables	
A_i^k	Arrival time of bus k at a transit node i
E_i^k	Battery energy level of bus k when arriving at node i
$h_{kk'}^{ss'}$	1 if bus k' arrives at dummy charging node s' later than bus k's arrival time at dummy charging node s, 0 otherwise
v_r	1 if customer r is served, 0 otherwise
Decision variables	
x_{ij}^k	1 if arc (i,j) is traversed by bus k, 0 otherwise.
y_{ij}^r	if a bus serves customer r on arc (i,j), 0 otherwise
z_{ij}^r	1 if customer r uses transit arc (i,j), 0 otherwise
w_{ij}	1 if customer i walks to transit station j, 0 otherwise
B_i^k	Beginning time of service of bus k at node i
τ_s^k	Charging duration for bus k at charger $s \in S$

3.2.1 Problem description

We consider the IDRT service with three transport modes: walking, on-demand buses (called buses hereafter), and fixed-route transit (called transit hereafter), to serve a set of customer requests. Customers send their requests via a dedicated smartphone application, indicating their origins, destinations, and desired pickup or drop-off time windows. Before the service begins, the operator communicates whether customers' requests are accepted and provides information to the passengers regarding modes used, arrival times of buses/trains, and the sequential trip legs of their paths that will connect them from their origins to their destinations. A customer's trip from their origin to destination can be one of the following five travel options:

1) Walk + transit + walk
2) Bus + transit + walk
3) Walk + transit + bus
4) Bus + transit + bus
5) Bus only

The last four options include first/last- mile bus service or walking. When it is a first-/last- mile service, buses need to arrive at transit stations within a pre-defined buffer time before the departure of trains/after the arrival of trains, ensuring a reasonable inter-modal transfer time for customers. The fleet is heterogeneous in terms of number of passengers and battery capacity. Buses are charged exclusively at operator-owned charging stations with partial recharge policy. For vehicles' charging operations, no overlaps are allowed at any charger, i.e. buses do not have to wait to be recharged. Transit services are operated based on fixed-schedule timetables. We assume the capacity of trains is sufficiently large compared to the number of ride requests. No train delays are assumed with a constant dwelling time at

each station. The objective is to minimize the weighted sum of bus routing time, customers' journey time, and the penalty associated with customer rejections.

The EIDARP is modeled on a directed graph $\mathcal{G} = (N, A)$, where N is the set of nodes and A is the set of arcs. Considering the nature of different transport modes involved in the IDARP, N is decomposed into five subsets related to different node types: customer origins, customer destinations, transit stations (stops), chargers, and depots. Note that chargers are associated with different charging stations. Let R be the set of customer requests, i.e., $R = \{1, \ldots, n\}$. Let $P = \{1, \ldots, n\}$ be the set of customers' origin nodes, and $D = \{n+1, \ldots, 2n\}$ be the set of customers' destination nodes. Each transit station is duplicated with a set of dummy transit nodes for modeling timetabled transit networks (described in Section 3.2.2), i.e., $G = \{2n+1, \ldots, 2n+|G|\}$, where $|G|$ is the number of transit dummy nodes for timetabled transit service modeling. We model the charging stations capacity constraints based on the replication-based approach. The set of charger nodes S is composed of a set of charger dummy nodes where each charger $c \in C$ is duplicated with m dummies to allow multiple visits of buses, i.e., $S = \{2n+|G|+1, \ldots, 2n+|G|+m|C|\}$. We denote $O = \{2n+|G|+m|C|+1, \ldots, 2n+|G|+m|C|+|O|\}$ be the set of origin depots, and $\bar{O} = \{2n+|G|+m|C|+|O|+1, \ldots, 2n+|G|+m|C|+2|O|\}$ be the set of destination depots. The origin and destination depots for vehicle k are denoted as o^k and $\bar{o}^k$, respectively.

The directed graph $\mathcal{G}$ comprises three subgraphs to model passenger/vehicle flows on walk, bus, and transit network. Bus subgraph $\mathcal{G}_B$ is defined as $\mathcal{G}_B = (P \cup D \cup G \cup S \cup O \cup \bar{O}, A_B)$, transit subgraph as $\mathcal{G}_T = (G, A_G)$, and walking subgraph as $\mathcal{G}_R = (P \cup D \cup G, A_R)$, where A_B is the set of bus arcs containing directed arcs between any pair of nodes in $\mathcal{G}_B$. A_G is the set of transit arcs connecting feasible transit nodes, which is further described in Section 3.2.2. A_R is the set of walking arcs connecting customers' origins/destinations from/to transit nodes within customers' maximum walking distance. Let $\sigma_m^-(i)$ and $\sigma_m^+(i)$ denote the set of incoming and outgoing nodes of node i of mode m, respectively. To facilitate the reading, we present the timetabled transit service modelling for IDARP in Section 3.2.2, followed by the MILP formulation of the EIDARP in Section 3.2.3.

3.2.2 Departure-expanded graph for timetabled transit service modelling

The timetabled transit services are modeled on a directed graph $\mathcal{G}_T$ composed of a set of transit nodes and transit arcs. Given a set of physical transit stations, we define a set of transit lines for which associated transit services are specified by the timetables. For clarity, the notation related to the transit subgraph are defined as follows.

3.2.2.1 Definitions of the departure-expanded network

I. ***Transit line***: A transit line l is the sequence of undirected transit stations $F_l = \{f_1, \ldots, f_m\}$, where f_i is a physical transit station node. Each transit line operates in both directions. If a transit station is traversed by multiple lines, this transit node is duplicated as many as the number of transit lines.
II. ***Transfer station***: A transit station where customers can transfer between transit lines.
III. ***Transit node***: An ordered set of duplicates of transit stations based on the timetables of transit lines.
IV. ***Transfer arc***: A set of directed arcs connecting pairs of transfer nodes of different transit stations.
V. ***(Scheduled) route***: A scheduled route is a sequence of transit nodes visited by a transit vehicle defined by its scheduled timetables (see an example in Figure 3.1 and Figure 3.2).
VI. ***Transit arc***: A directed arc connecting two different transit nodes. Two transit nodes are connected if there exists a feasible shortest route that satisfies predefined transfer time constraints between

different scheduled departures of trains at transfer stations. The arc weight corresponds to the shortest travel time between them, given the timetables of the transit network.

Let $\mathcal{D}$ denote the set of departures (schedules) of trains, i.e., $\mathcal{D} = \{\mathcal{D}_l\}_{l\in\mathcal{L}}$, and F denote the set of physical transit stations, i.e., $F = \{F_l\}_{l\in\mathcal{L}}$. We denote p_{dl} as a scheduled route, i.e., $p_{dl} = \{g_1, g_2, \dots, g_m\}, \forall d \in \mathcal{D}_l, l \in \mathcal{L}$. The ordered index d corresponds to the d-th scheduled vehicle in the service timetable of line l. A timetable associated with line l is a table of departure times of transit nodes on the set of scheduled routes $\{p_{dl}\}_{d\in\mathcal{D}_l}$ of line l. Given transit nodes and the timetables, we model the transit subgraph based on the departure-expanded graph approach where the directed arcs connect two transit nodes if there exists a feasible shortest path between them. This idea extends our previous layered graph approach for first-mile service modeling by considering timetabled services on a transit network with integrated bus-transit services (Fang and Ma, 2023; Ma et al., 2024).

3.2.2.2 Construction of the departure-expanded network

Consider a timetabled transit service operating in a region. The transit (e.g., train) service is characterized by a set of physical transit stations, transit lines and timetables (which defines a timetabled transit network). For EIDARP problem modeling, a set of transit nodes $G = \{g_d^f\}_{f\in F_l, d\in\mathcal{D}_l, l\in\mathcal{L}}$ are generated to characterize the scheduled transit services. The structure of G is illustrated in Figure 3.1 and Figure 3.2. A scheduled route of line l at departure d is then represented by $\{g_d^f\}_{f\in F_l}$(rows in Figure 3.2 (a)), and all transit nodes of a physical transit station f at line l is $\{g_d^f\}_{d\in\mathcal{D}_l}$(columns in Figure 3.2 (a)). Each transit node i is associated with a departure time $\underline{\theta_i}$ of its corresponding transit vehicle. The transit vehicle arrival time at i is $\bar{\theta}_i = \underline{\theta_i} - \pi$ with a constant dwelling time π. To ensure an efficient transfer, an allowable transfer time constraint is fixed and defined by minimum and maximum allowable transfer times, denoted as τ^{min} and τ^{max}, respectively, to walk from platform i to platform j and wait for the departure of transit vehicles at j. If $\underline{\theta_j} - \underline{\theta_i} < \tau^{min}$ or $\underline{\theta_j} - \underline{\theta_i} > \tau^{max}$, a transit path traversing from transit node i to j is not feasible. Note that τ^{min} and τ^{max} are determined by the operator to ensure the quality of service with allowed maximum transfer times in the transit network. Given an entry-exit pair of transit nodes $(i, j) \in G$, there might be multiple feasible paths connecting them. Our interest is to compute (transfer) feasible shortest transit paths between each pair (i, j) of transit nodes. If there exist, directed arcs are generated for (i, j) with the arc weight being the shortest travel time between them. Then, the departure-expanded graph is constructed with transit nodes and directed arcs associated with feasible shortest travel times. Infeasible arcs are trimmed off [5].

We present an illustrative example to further explain our approach in Figure 3.1 and Figure 3.2. Figure 3.1 presents a scheduled transit service with six physical transit stations and two transit lines. Line 1 (A-B-C-D) and Line 2 (D-F-G-H) are connected at two transfer stations (B/F and D/H). The scheduled timetables for each line are presented on the left of Figure 3.1 with both directions indicated. The corresponding transit nodes $G = \{1,2,\dots,20\}$ are depicted in Figure 3.2, where the numbers within the filled points represent the ordered indexes of the transit nodes. Each transit node corresponds to a transit station and a specific departure time as defined by the timetable in Figure 3.1. Direct arcs represent

[5] Note that in Posada et al. (2017) the authors did not explicitly model the timetabled transit network. Instead, each pair of transit station dummy nodes is associated with a set of scheduled services resulting in numerous redundant arcs (binary decision variables) in their MILP formulation.

direct routes between two stations on the same departure, requiring no transfers, while transfer arcs connect transit nodes on different transit lines, provided the transfer time is within a predefined maximum transfer time limit. For example, the direct arc from node 1 to node 3 in Figure 3.2 means that a transit vehicle traverses stations 1, 2, and 3, consecutively. The weights on the arcs correspond to the travel times between two connected stations. There are 6 direct arcs for the first departure of each line. The generation of transfer arcs at transfer stations is explained in (b) of Figure 3.2. For the first transfer station, three transit nodes (2, 10, and 18) are associated with the station node B (Line 1), and two transit nodes (6 and 14) are associated with the station node F (Line 2). The transfer from node 1 (departure time is 7:20) to node 6 (departure time is 7:23) takes 3 minutes, while the transfer from node 1 (departure time is 7:20) to node 10 (departure time is 7:46) takes 26 minutes. By limiting the maximum allowed transfer time (e.g., τ^{max}=10 minutes) within the same transit station, infeasible transfer arcs are trimmed off. As a result, only two feasible transfer arcs are created given the 10-minute maximum transfer time limit (Figure 3.2).

Timetable of Line 1

Departure	A		B		C		D
1	20	→	26	→	32	→	38
2	43	←	37	←	31	←	25
3	50	→	56	→	62	→	68

Timetable of Line 2

Departure	E		F		G		H
1	23	→	29	→	35	→	41
2	46	←	40	←	34	←	28

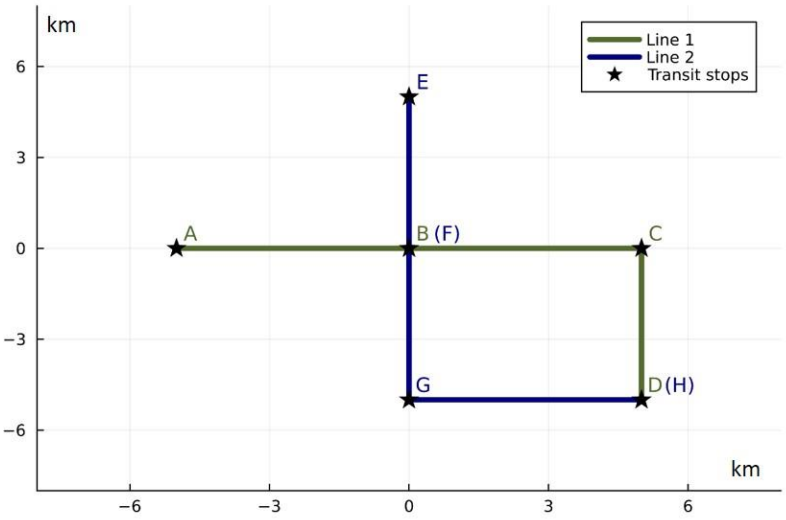
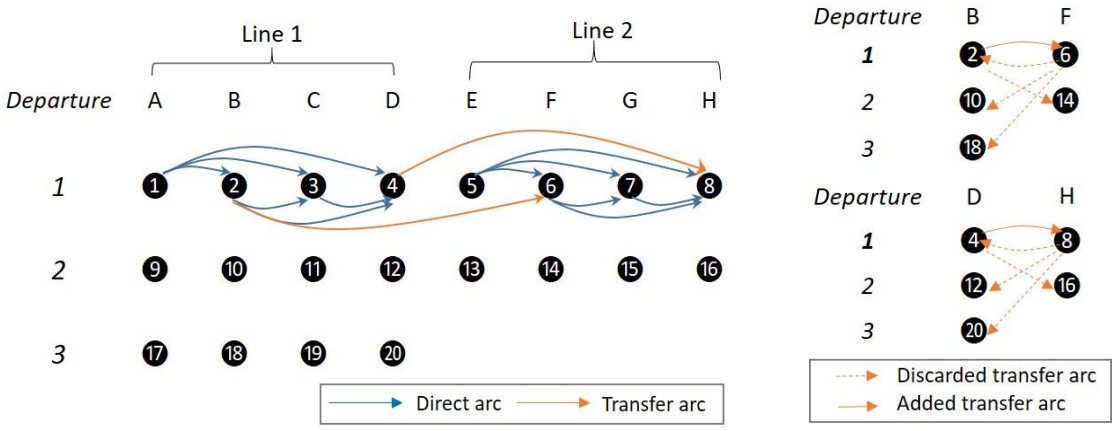


Figure 3.1: An illustrative example of transit service timetables with two transit lines



(a) Transit nodes (filled black points) and transit arcs (b) Transfer arcs generation

Figure 3.2: Transit arc generation for the departure-expanded graph of the example in Figure 3.1

Algorithm 3 describes the details to generate the departure-expanded graph $\mathcal{G}_T$ for the EIDARP. The algorithm includes mainly two steps: the first step is to generate a temporary departure-expanded transit graph $\mathcal{G}'_T = (G, A'_G)$ with temporary transit arcs A'_G connecting direct transit and feasible transfer arcs for feasible shortest path travel time calculation between pairs of transit nodes. The inputs are the sets of transit nodes G, transit lines $\mathcal{L}$, departures $\mathcal{D}$, arrival times $\bar{\theta}$ and departure times $\underline{\theta}$ of vehicles at transit nodes given the timetables, and minimum and maximum allowable transfer times τ^{min} and τ^{max}, respectively. The output is the departure-expanded graph and the feasible shortest travel times on transit arcs, i.e., $t_{ij}^m, \forall i,j \in G$. For transit line and departure pair, the direct arcs and transfer arcs are inserted to A'_G. Then we apply the conventional Dijkstra Algorithm (Dijkstra, 1959) to get the shortest travel time between pairs of transit nodes on $\mathcal{G}'_T$. Finally, transfer arcs are removed.

Algorithm 3 Transit arcs generation

Input: $G, \mathcal{L}, \mathcal{D}, \bar{\theta}, \underline{\theta}, \tau^{min}, \tau^{max}$
Output: $A_G,\ t_{ij}^m \quad \forall i,j \in G$

Set $A'_G = \emptyset, A_G = \emptyset,\ t_{ij}^m = \infty, \forall j \in G$
for $l \in \mathcal{L}$
    for $d \in \mathcal{D}_l$
        Insert direct arcs (i,j) to A'_G and set $t_{ij}^m = \underline{\theta}_j - \underline{\theta}_i$
        for $i \in$ transfer nodes of line l
            for $(i,j) \in$ outgoing transfer arcs from i
                if $\tau^{min} \leq \underline{\theta}_j - \bar{\theta}_i \leq \tau^{max}$
                    Insert transfer arcs (i,j) to A'_G and set $t_{ij}^m = \underline{\theta}_j - \bar{\theta}_i$
                end if
            end for
        end for
    end for
end for
$\mathcal{G}'_T = (G, A'_G)$
for $i \in G$
    Calculate the shortest travel time $\tilde{t}_{ij}^m$ for $\forall j \in G, j \neq i$ using Dijkstra Algorithm
    Insert (i,j) to A_G and update $t_{ij}^m = \tilde{t}_{ij}^m$ if $t_{ij}^m < \infty$
end for
Remove transfer arcs from A_G

The advantage of using this departure-expanded graph representation for the MILP formulation of IDARP compared to the formulation of Posada et al. (2017) is shown in Figure 3.3 (a) and Figure 3.3 (b) respectively. Posada et al. (2017) duplicated each transit station $\bar{M}_i$ times, where $\bar{M}_i$ is an upper bound on the number of visits to a physical transit station i for transit line l, e.g., $\bar{M}_i = min(|\mathcal{D}_l|, n)$, where n is the number of customer requests. In general, the number of customers is greater than that of train departures, resulting in $\bar{M}_i = |\mathcal{D}_l|$ (same as our approach). Our departure-expanded network associates each transit node with a time window corresponding to the departure/arrival time of trains and a fixed buffer time. This enables a compact scheduled transit service graph generated using Algorithm 1. Figure 3.3 (a) (our approach) keeps the feasible transit arcs only, while Figure 3.3 (b) (Posada's approach) needs all transit nodes from station i to j to be connected. As the arrival and

departure times of trains are integrated with each transit node in our approach, the departure-expanded graph allows to trim off the bus arcs connecting transit nodes by time-window tightening (described in Section 3.3) (see Figure 3.3 (a)).

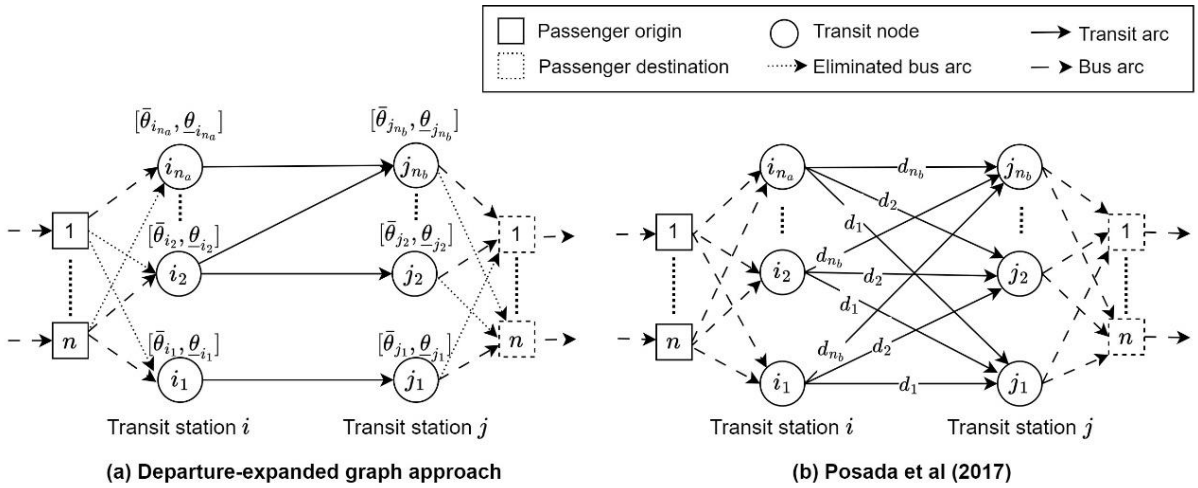


Figure 3.3: Comparison of the departure-expanded graph approach and the state-of-the-art approach (Posada et al. 2017)

3.2.3 MILP formulation

We formulated the EIDARP problem as a MILP as follows. We use a three-index formulation to track on-demand buses' routes and energy consumption with a fleet of heterogeneous buses. Given a set of requests, the objective function Eq. (3.1) minimizes the weighted sum of bus routing time, customers' journey time, and the penalty for unserved customers.

$$Min\ \lambda_1 \sum_{(i,j)\in A_B} \sum_{k\in K} t_{ij} x_{ij}^k + \lambda_2 \sum_{r\in R} L_r + \omega \sum_{r\in R} (1 - v_r) \tag{3.1}$$

subject to

$$\sum_{j\in \sigma_B^+(o^k)} x_{o^k j}^k = 1,\ \forall k \in K \tag{3.2}$$

$$\sum_{i\in \sigma_B^-(\bar{o}^k)} x_{i,\bar{o}^k}^k = 1,\ \forall k \in K \tag{3.3}$$

$$\sum_{j\in \sigma_B^+(o)} x_{oj}^k = 0,\ \forall k \in K, o \in O, o \neq o^k \tag{3.4}$$

$$\sum_{j\in \sigma_B^-(i)} x_{ji}^k - \sum_{j\in \sigma_B^+(i)} x_{ij}^k = 0,\ \forall k \in K, i \in N \setminus \{O, \bar{O}\} \tag{3.5}$$

$$y_{ij}^r \leq \sum_{k\in K} x_{ij}^k,\ \forall r \in R, (i,j) \in A_B \tag{3.6}$$

Constraints (3.2)-(3.3) ensure that buses need to leave and return to their respective depots. Constraint (3.4) forbids each bus from visiting other depots except its own. Constraint (3.5) ensures bus flow conservation. Constraint (3.6) ensures the consistency between x_{ij}^k and y_{ij}^r.

$$w_{ij} = 0,\ \forall i \in P, j \in G, (i,j) \notin A_R \tag{3.7}$$

$$v_i = \sum_{j \in \sigma_R^+(i)} w_{ij} + \sum_{j \in \sigma_B^+(i)} y_{ij}^i,\ \forall i \in P \tag{3.8}$$

$$v_i = \sum_{j \in \sigma_R^-(n+i)} w_{j,n+i} + \sum_{j \in \sigma_B^-(n+i)} y_{j,n+i}^i,\ \forall i \in P \tag{3.9}$$

$$\sum_{j \in \sigma_B^+(i)} y_{ij}^r - \sum_{j \in \sigma_B^-(i)} y_{ji}^r = 0,\ \forall i \in P \cup D, r \in R, i \neq r, i \neq n + r \tag{3.10}$$

$$w_{i,n+r} - w_{ri} + \sum_{j \in \sigma_B^+(i)} y_{ij}^r - \sum_{j \in \sigma_B^-(i)} y_{ji}^r + \sum_{j \in \sigma_T^+(i)} z_{ij}^r - \sum_{j \in \sigma_T^-(i)} z_{ji}^r = 0, \forall r \in R, i \in G \tag{3.11}$$

Constraint (3.7) enforces that customers will not walk from/to transit stations if the distance between their origins/destinations and transit stations exceeds the maximum walking distance. A_R is the set of directed walking arcs to connect transit station nodes within the maximum constant walking distance. Constraints (3.8) and (3.9) state that each customer can be either served or not served. If served, the customer must be served by a bus or walk to/from a transit station at their origin and destination. $\sigma_R^+(i)$ and $\sigma_R^-(i)$ are the sets of outgoing and incoming nodes of i on walking arcs A_R, respectively. $\sigma_B^+(i)$ and $\sigma_B^-(i)$ are the sets of outgoing and incoming nodes of i on bus arcs, respectively. Note that given customers' time windows constraints at origins/destinations, their maximum ride time, and inter-modal allowable transfer buffer (i.e., 15 minutes before the departure of trains) associated with transit nodes, a preprocessing procedure is applied to trim off infeasible arcs by verifying these time-window constraints. $\sigma_B^+(i)$ and $\sigma_B^-(i)$ contains only a small fraction of relevant outgoing/incoming bus arcs. Customer flow consistency is ensured at bus nodes (Constraint (3.10)) and transit nodes (Constraint (3.11)).

$$\sum_{(i,j) \in A_G} z_{ij}^r \leq 1, \forall r \in R \tag{3.12}$$

$$w_{ri} \leq \sum_{j \in \sigma_T^+(i)} z_{ij}^r, \forall r \in R, i \in G \tag{3.13}$$

$$\sum_{j \in \sigma_B^+(i)} x_{ij}^k \leq \sum_{j \in \sigma_B^+(n+i)} x_{n+i,j}^k + M \sum_{(i,j) \in A_G} z_{ij}^i,\ \forall k \in K, i \in P \tag{3.14}$$

Constraint (3.12) states that each customer can use timetabled transit service at most once with entry transit node i and exit transit node j. If customers walk to a transit station, they must be served by transit services (Eq. 3.13). Constraint (3.14) states that if customers do not use transit service, they must be served by the same buses visiting their origins and destinations.

$$y_{ij}^r = 0,\ \forall r \in R, i \in S \cup 0, j \in \sigma_B^+(i) \tag{3.15}$$

$$\sum_{r \in P} y_{ij}^r \leq Q^k + M\left(1 - x_{ij}^k\right),\ \forall k \in K, (i,j) \in A_B \tag{3.16}$$

$$B_j^k \geq B_i^k + t_{ij} + u_i - M\left(1 - x_{ij}^k\right),\ \forall k \in K, (i,j) \in A_B, i \notin S \tag{3.17}$$

$$B_j^k \geq B_i^k + t_{ij} + \tau_i^k + u_i - M\left(1 - x_{ij}^k\right),\ \forall k \in K, (i,j) \in A_B, i \in S \tag{3.18}$$

$$e_i \leq B_i^k \leq l_i,\ \forall k \in K, i \in P \cup D \cup O \cup \bar{O} \tag{3.19}$$

Constraint (3.15) ensures that no customer can be on board at the depot and charging stations, while constraint (3.16) states that passenger load cannot exceed bus capacity. Constraints (3.17) and (3.18)

ensure the consistency of the beginning time of service at bus nodes and charger nodes, respectively. Constraint (3.19) states the time-window constraints at pickups, drop-offs and depots.

$$\underline{\theta}_i - \gamma \le A_i^k \le \bar{\theta}_i + \gamma,\ \forall k \in K, i \in G \tag{3.20}$$

$$A_j^k \ge B_i^k + t_{ij} + u_i - M\left(1 - x_{ij}^k\right),\ \forall k \in K, j \in G, i \in \sigma_B^-(j) \tag{3.21}$$

$$A_j^k \le B_i^k + t_{ij} + u_i + M\left(1 - x_{ij}^k\right),\ \forall k \in K, j \in G, i \in \sigma_B^-(j) \tag{3.22}$$

$$A_i^k \ge \underline{\theta}_i - \gamma - M\left(1 - \sum_{j \in \sigma_T^+(i)} z_{ij}^r\right),\ \forall r \in R, i \in G, k \in K \tag{3.23}$$

$$A_i^k \le \underline{\theta}_i + M\left(1 - \sum_{j \in \sigma_T^+(i)} z_{ij}^r\right),\ \forall r \in R, i \in G, k \in K \tag{3.24}$$

$$B_i^k \ge \bar{\theta}_i - M\left(1 - \sum_{j \in \sigma_T^-(i)} z_{ji}^r\right),\ \forall r \in R, i \in G, k \in K \tag{3.25}$$

$$B_i^k \le \bar{\theta}_i + \gamma + M\left(1 - \sum_{j \in \sigma_T^-(i)} z_{ji}^r\right),\ \forall r \in R, i \in G, k \in K \tag{3.26}$$

To ensure a maximum waiting time at transit stations, an arrival time is introduced by constraints (3.21) and (3.22) with its upper bound and lower bound (Eq. 3.20). As buses can visit a transit node for both dropping off and picking up customers, constraint (3.20) defines the lower and upper bounds as the departure time minus and plus a maximum waiting time. Constraints (3.23)-(3.26) are inter-modal waiting (transfer) time constraints at transit stations for customers transfer between buses and transit stations. Constraints (3.23) and (3.24) coordinate buses arrival at transit node i for the first-mile service within the predefined time window $[\underline{\theta}_i - \gamma]$ with a maximum waiting time γ. Constraints (3.25) and (3.26) are for the last-mile service to ensure that buses pick up customers from transit nodes (stations) within the buffer time γ after transit vehicle's arrival at time $\bar{\theta}_i$.

$$\underline{\theta}_j \le l_i + t_{ij}^w + M\left(1 - w_{ij}\right),\ \forall i \in P, j \in \sigma_B^+(i) \tag{3.27}$$

$$\underline{\theta}_j \ge e_i + t_{ij}^w - M\left(1 - w_{ij}\right),\ \forall i \in P, j \in \sigma_B^+(i) \tag{3.28}$$

$$\bar{\theta}_j \le l_i - t_{ji}^w + M\left(1 - w_{ji}\right),\ \forall i \in D, j \in \sigma_B^-(i) \tag{3.29}$$

$$\bar{\theta}_j \ge e_i - t_{ji}^w - M\left(1 - w_{ji}\right),\ \forall i \in D, j \in \sigma_B^-(i) \tag{3.30}$$

$$L_r \le L_r^{\max},\ \forall r \in R \tag{3.31}$$

$$L_r = \sum_{(i,j) \in A_B} t_{ij} y_{ij}^r + \sum_{(i,j) \in A_G} t_{ij}^m z_{ij}^r + \sum_{i \in G} t_{ri}^w w_{ri} + \sum_{i \in G} t_{n+r,i}^w w_{n+r,i}\ \forall r \in R \tag{3.32}$$

Constraints (3.27)-(3.30) ensures time-window constraints if customers walk to/from a transit station. It is worth to mention that we assume that if customers walk from/to transit stations, their waiting time is neglected. Constraint (3.32) gets the journey time of customers. Constraint (3.31) ensures that the maximum journey times of customers cannot be violated. Note that the customers' maximum journey times are determined by multiplying a detour factor φ and the direct travel time by buses between their origins and destinations.

$$E_0^k = E_{\text{init}}^k,\ \forall k \in K \tag{3.33}$$

$$E_{\min}^k \le E_i^k \le E_{\max}^k,\ \forall i \in N \setminus O, k \in K \tag{3.34}$$

$$E_j^k \geq E_i^k + \alpha_i \tau_i^k - c_{ij}\beta^k - M\left(1 - x_{ij}^k\right),\ \forall k \in K, (i,j) \in A_B, i \in S \quad (3.35)$$

$$E_j^k \leq E_i^k + \alpha_i \tau_i^k - c_{ij}\beta^k + M\left(1 - x_{ij}^k\right),\ \forall k \in K, (i,j) \in A_B, i \in S \quad (3.36)$$

$$E_j^k \geq E_i^k - c_{ij}\beta^k - M\left(1 - x_{ij}^k\right),\ \forall k \in K, (i,j) \in A_B, i \notin S \quad (3.37)$$

$$E_j^k \leq E_i^k - c_{ij}\beta^k + M\left(1 - x_{ij}^k\right),\ \forall k \in K, (i,j) \in A_B, i \notin S \quad (3.38)$$

Constraint (3.33) sets the initial SoC of buses. Constraint (3.34) ensures the upper bound and lower bound of buses' SoC. Constraints (3.35) and (3.36) track buses' SoC changes at bus nodes, while constraints (3.37) and (3.38) track buses' SoC changes at charger nodes.

$$\sum_{j \in \sigma_B^+(s)} x_{sj}^k \leq 1,\ \forall k \in K, s \in S \quad (3.39)$$

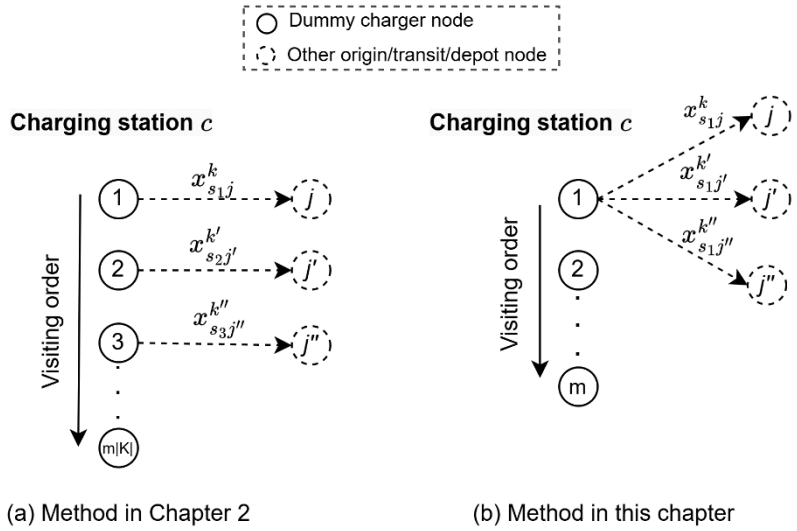
$$\sum_{j \in \sigma_B^+(l)} x_{lj}^k \leq \sum_{j \in \sigma_B^+(h)} x_{hj}^k,\ \forall k \in K, h, l \in S_c, c \in C, h < l \quad (3.40)$$

$$h_{kk'}^{ss'} + h_{k'k}^{s's} = 1, \forall\ s, s' \in\ S_c, c \in\ C, k, k' \in\ K, k \neq\ k' \quad (3.41)$$

$$B_{s'}^{k'} \geq\ B_s^k + \tau_s^k - M(1 - h_{kk'}^{ss'}), \forall\ s, s' \in\ S_c, c \in\ C, k, k' \in\ K, k \neq\ k' \quad (3.42)$$

Constraints (3.39)-(3.40) are related to bus charging operation constraints. Constraint (3.39) states that each vehicle k can connect each dummy charger node s at most once, but a dummy charging node s can be visited by different buses. With multiple charger dummy nodes associated with a charger, a bus is allowed to charge multiple times at the same charger by visiting different charger dummy nodes. The symmetry issue is avoided by constraint (3.40), which ensures that the used dummy charger nodes need to follow their ordered indexes. $h_{kk'}^{ss'}$ is an indicator being 1 if bus k' arrives later than bus k at the same physical charger. s' and s are the visited dummy charging nodes for bus k' and k, respectively (Eq.41). Constraint (3.42) ensures if bus k' arrives later than bus k, bus k' can start its charging session after the end of the charging session of bus k.

Figure 3.4 illustrates the difference in the new CSCCs modeling approach proposed in this chapter when there are multiple bus visits at a CS. This new formulation enables multiple buses to connect the same charger dummy node, whereas the previous method in Chapter 2 restricts each dummy to a single visit by one bus. Moreover, when the maximum number of recharge operations per charger is set to m, the method in Chapter 2 requires $m|K|$ dummies to accommodate the maximum operations for all buses. In contrast, the new method only requires m dummies. In the case of $|C|$ chargers, the new method reduces $m|K||C| - m|C|$ dummy nodes. Since each dummy node involves numerous arcs connecting to other nodes in the graph, reducing the number of unnecessary nodes can significantly decrease the overall problem size.



(a) Method in Chapter 2 (b) Method in this chapter

Figure 3.4: Illustration of two methods handling multiple visits at a charging station

Lastly, note that if customer rejection is not allowed, constraint (3.43) can be added.

$$v_r = 1,\ \forall r \in R \tag{43}$$

3.3 Strengthening the model

The problem size of Eq. (3.1) – Eq. (3.43) increases quickly with the number of customer requests. To address this issue, we propose time-window tightening and arc elimination methods and a set of valid inequalities to trim off unnecessary decision variables and tighten the feasible solution space.

3.3.1 Preprocessing

We apply the following time window tightening to strengthen the MILP model. First, the time windows associated with the origins of outbound requests are tightened as $e_i = \max\{0, e_{n+i} - L_i^{max} - \mu_i\}$ and $l_i = \min\{l_{n+i} - t_{i,n+i} - \mu_i, t_{end}\}$. For inbound requests, the time-windows associated with their destinations are tightened as $e_{n+i} = \max\{0, e_i + \mu_i + t_{i,n+i}\}$ and $l_{n+i} = \min\{l_i + \mu_i + L_i^{max}, t_{end}\}$ (Cordeau, 2006). Since transit nodes could serve as departure and arrival nodes for train service, the above time-window tightening method is not applicable to those. Nevertheless, their time windows are determined by constraints (3.23) - (3.26).

If buses depart at their earliest time from node i and directly arrive at node j, and the arrival times exceed the end of time window l_j, these bus arcs are trimmed, i.e., $e_i + t_{ij} + \mu_i > l_j, \forall (i,j) \in A_B$ and $i \neq j$. For the remaining bus arcs, we retain the following relevant arcs in $\mathcal{G}_B$. The other bus arcs are trimmed off.

(1) $(o,j),\ \forall o \in O, j \in P \cup S \cup G \cup \bar{O}$

(2) $(i,j),\ \forall i \in P, \forall j \in P \cup D \cup G$

(3) $(i,j),\ \forall i \in D, \forall j \in P \cup D \cup G \cup S \cup \bar{O}$

(4) $(i,j),\ \forall i \in G, \forall j \in P \cup D \cup S \cup \bar{O}$, and $\forall j \in G$, if i, j are not at the same departure

(5) $(i,j),\ \forall i \in S, \forall j \in P \cup G \cup S \cup \bar{O}$

Rule (1) retains arcs connecting the origin and destination depot, allowing buses to stay in the depot if unused. Rule (2) retains arcs from customer origins to all nodes except charger nodes and the depots. Rule (3) states that customers' destinations can connect to all the nodes except origin depots. Rule (4) states that buses can travel from transit station nodes to all nodes except the origin depots and the bus arcs at different departures within transit services. We assume that bus speed is inferior to train speed so the bus arcs within the same departures are removed. Rule (5) retains arcs from charger nodes to other nodes except customer destinations and the depot.

3.3.2 Valid inequalities

We propose a set of valid inequalities to tighten the solution space as follows. Constraints (3.44) and (3.45) prohibit arcs from arriving at any pickup node and arcs from leaving from any drop-off node. As each bus has its dedicated depot, constraint (3.46) prohibits buses from returning to irrelevant destination depots. We further restrict the feasible range for the continuous variables related to the beginning time of service. Constraints (3.47) and (3.48) set the beginning time of service of buses at transit nodes to zero if the related transit nodes are not used by customers.

$$\sum_{j \in \sigma_B^-(i)} y_{ji}^i = 0,\ \forall i \in P \tag{3.44}$$

$$\sum_{j \in \sigma_B^+(n+i)} y_{n+i,j}^i = 0,\ \forall i \in P \tag{3.45}$$

$$\sum_{i \in \sigma_B^-(\bar{o}^k)} x_{i,j}^k = 0,\ \forall k \in K, j \in \bar{O}, j \neq \bar{o}^k \tag{3.46}$$

$$B_i^k \leq M \sum_{j \in \sigma_T^+(i)} \sum_{r \in R} z_{ij}^r + M \sum_{j \in \sigma_T^-(i)} \sum_{r \in R} z_{ji}^r,\ \forall i \in G, k \in K \tag{3.47}$$

$$B_i^k \geq -M \sum_{j \in \sigma_T^+(i)} \sum_{r \in R} z_{ij}^r - M \sum_{j \in \sigma_T^-(i)} \sum_{r \in R} z_{ji}^r,\ \forall i \in G, k \in K \tag{3.48}$$

If a customer is rejected ($v_r = 0$), constraints (3.49) – (3.51) ensure that no arc connects to/from this customer.

$$y_{ij}^r \leq v_r,\ \forall r \in R, (i,j) \in A_B \tag{3.49}$$

$$w_{ij} \leq v_i,\ \forall i \in P, j \in G \tag{3.50}$$

$$\sum_{(i,j) \in A_G} z_{ij}^r \leq v_r,\ \forall r \in R \tag{3.51}$$

Lastly, we also add constraints (3.52) and (3.53) referring to Posada et al. (2017). Constraint (3.52) ensures no cycle between any two bus nodes. If no customers use transit arcs, constraint (3.53) ensures no buses visit the corresponding transit nodes.

$$x_{ij}^k + x_{ji}^k <= 1,\ \forall k \in K, (i,j) \in A_B \tag{3.52}$$

$$\sum_{j \in \sigma_B^+(i)} x_{ij}^k \leq \sum_{r \in P} \sum_{j \in \sigma_T^+(i)} z_{ij}^r + \sum_{r \in P} \sum_{j \in \sigma_T^-(i)} z_{ji}^r,\ \forall i \in G, k \in K \tag{3.53}$$

3.4 Computational studies

In this section, we test the proposed EIDARP model on a set of test instances. First, we describe the test instance generation. Then we solve the test instances with a maximum computational time limit of 4 hours. The performance of our departure-expanded graph approach is compared with standard IDARP MILP formulation (Posada et al., 2017). We solve the test instances with low and full initial SoC of vehicles to analyze the impact of charging operations on the optimality gaps. The effectiveness of the new CSCC formulation is compared to the standard replication-based CSCC formulation with duplicates of single-visit dummy charger nodes (Bruglieri et al., 2019; Ma et al., 2024). The test instances are solved by Gurobi MIP solver V11.0 on a laptop with an 11th Gen Intel(R) Core(TM) i5-1135G7 CPU and 64GB RAM. The test instances are available at https://github.com/YMF2022/EIDARP-instances.git.

3.4.1 Test instances

We generate two sets of EIDARP test instances for the scenarios with two crossed transit lines and three transit lines in a square area of 16 km width. The two scenarios are illustrated in Figure 3.5 and Figure 3.7. In Scenario 1, there are two transit lines operating both directions. Line 1 serves stations A, B, and C, while line 2 serves stations D, E, and F. Transfer stations are located at B and E, enabling passengers to switch between the two lines. The IDARP service planning horizon is 2 hours from t_0 during which there are 4 departures (two for each direction) operating for each line. For Line 1, two trains depart at t_0+20 and t_0+50 from A to C, while two trains depart at t_0+25 and t_0+55 from C to A. Time is measured in minutes. For Line 2, two trains depart from D to F at time t_0+23 and t_0+53, and two trains depart from F to D at time t_0+28 and t_0+58. The average operating speed of the train is 50 km/h, while the bus speed is set as 25 km/h. Customers' origins and destinations are randomly located in the operational area. We assume an unlimited fleet size to serve customers. The bus operator uses two types of buses with two different depots. Two operator-owned DC fast chargers are located at different locations. We assume a service time (i.e. access and egress time) for using any of the charging stations. Table 3.2 presents the parameter setting for the numerical study. Note that the penalty cost ω for customer rejection is set to a large value so that the service will serve as many customers as possible.

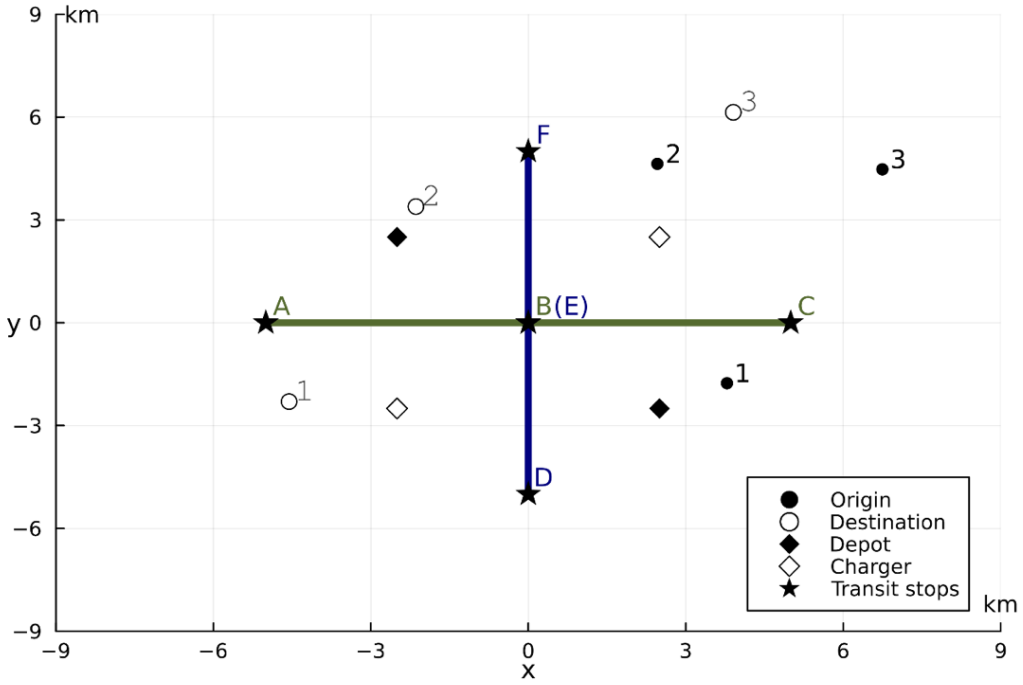


Figure 3.5: Example of a test instance in scenario 1 with two transit lines

Table 3.2: The parameter settings of the numerical study

Parameter	Value
Operation duration	2 hours
Number of customers	6 to 20
Number of transit lines and stations	2 lines and 5 stations (Scenario 1), 3 lines and 13 stations (Scenario 2)
Number of departures	4 per line (2 per direction)
Number of depots	2
Number of chargers	2
Number of bus types	2 types
Bus capacity Q^k	15 and 22 passengers/bus
Bus battery capacity	69 and 103.5kWh
Bus energy consumption rate	0.552 and0.828kWh/h
Bus average speed	25km/h for both types
$E^k_{\max}$ $(E^k_{\min})$	80%(10%) of the battery capacity
Charging power	0.83kWh/minute (DC fast charger)
Maximum waiting time at transit stations γ	10 minutes
Maximum walking distance	1.5km
Walking speed	0.085 km/minute
Detour factor φ	1.5
λ_1, λ_2	1, 1
Penalty cost ω	100
Service time μ	0.5 minute
Access Service time at charging station	1 minute

3.4.2 Comparison to the standard MILP formulation of IDARP (Posada et al., 2017)

We implement the strengthened MILP formulation (Model 2) of Posada et al. (2017) and compare it with our formulation. To ensure an impartial comparison, both models are revised to solve the same problem with a fleet of homogeneous vehicles, single resource (passenger seats), and unique depot located at station B. EVs and customer rejections are not considered. The objective function (1) is revised as Eq. (3.54) to minimize bus and train travel time, which is consistent with Posada et al. (2017). We apply the preprocessing procedures described in Section 3.3.1 to both models. Valid inequalities (3.52) and (3.53) are applied to the model of Posada et al. (2017), while valid inequalities (3.44)-(3.51) are applied to our model.

$$Min\ \lambda_1 \sum_{(i,j)\in A_B} \sum_{k\in K} t_{ij} x_{ij}^k + \lambda_2 \sum_{r\in R} \sum_{(i,j)\in A_G} t_{ij}^T z_{ij}^r \tag{3.54}$$

Scenario 1 with two transit lines is used (Figure 3.5) to generate test instances. Table 3.3 shows the computational results compared to Posada's approach. Within a 4-hour computational time, the model of Posada et al. (2017) finds 6 optimal solutions, while our approach finds 8 optimal solutions with up to 12 customers. Furthermore, Posada's approach cannot find feasible solutions for instances with 16 and more than 20 customers. When comparing the CPU time for the 6 instances that are optimally solved, our approach shows more than 95% CPU time reduction. For larger instances with more than 40 requests, feasible solutions cannot be found for both models by Gurobi given a four-hour computational time. Figure 3.6 compares the number of decision variables for both approaches, showing that around 50% of decision variables are trimmed off. This is due to the departure-expanded transit graph $\mathcal{G}_T$ introduced in Section 3.2.2 and the valid inequalities (3.44)–(3.51), which trim off unnecessary arcs and reduce the solution space significantly.

Table 3.3: Computational results compared to Posada et al. (2017)

# of cus.	Posada et al. (2017)			Departure-expanded graph approach			
	Obj. value[1] (min)	Gap to lower bound	CPU (s)	Obj. value[1] (min)	Gap to lower bound	CPU (s)	CPU time reduction[2]
5	**169.47**	**0.0%**	212	**169.47**	**0.0%**	1	-99.8%
6	**232.26**	**0.0%**	172	**232.26**	**0.0%**	0	-99.8%
7	**236.71**	**0.0%**	5232	**236.71**	**0.0%**	1	-100.0%
8	**269.59**	**0.0%**	926	**269.59**	**0.0%**	3	-99.7%
9	**258.96**	**0.0%**	4532	**258.96**	**0.0%**	1	-100.7%
10	346.61	26.0%	14000	**344.85**	**0.0%**	263	-[3]
11	**327.06**	**0.0%**	5753	**327.06**	**0.0%**	103	-98.2%.
12	381.56	28.6%	14400	**372.60**	**0.0%**	5274	-
13	389.54	25.1%	14400	371.28	2.2%	14400	-
14	418.28	43.2%	14400	414.27	27.2%	14400	-
15	425.76	44.1%	14400	418.86	26.0%	14400	-
16	*[4]	*	14400	424.37	28.0%	14400	-
18	492.73	57.1%	14400	446.72	43.2%	14400	-
20	*	*	14400	878.00	67.1%	14400	-
30	*	*	14400	946.27	64.9%	14400	-

Notes: 1. Based on a maximum computational time of 4 hours. Optimal solution is in bold. 2. CPU time reduction compared to the original model. 3. Not available if both models are not optimally solved. 4. Gap to upper bound was not found after 4 hours; the same was true for the test instances with 40 and 50 customers (not reported in this Table).

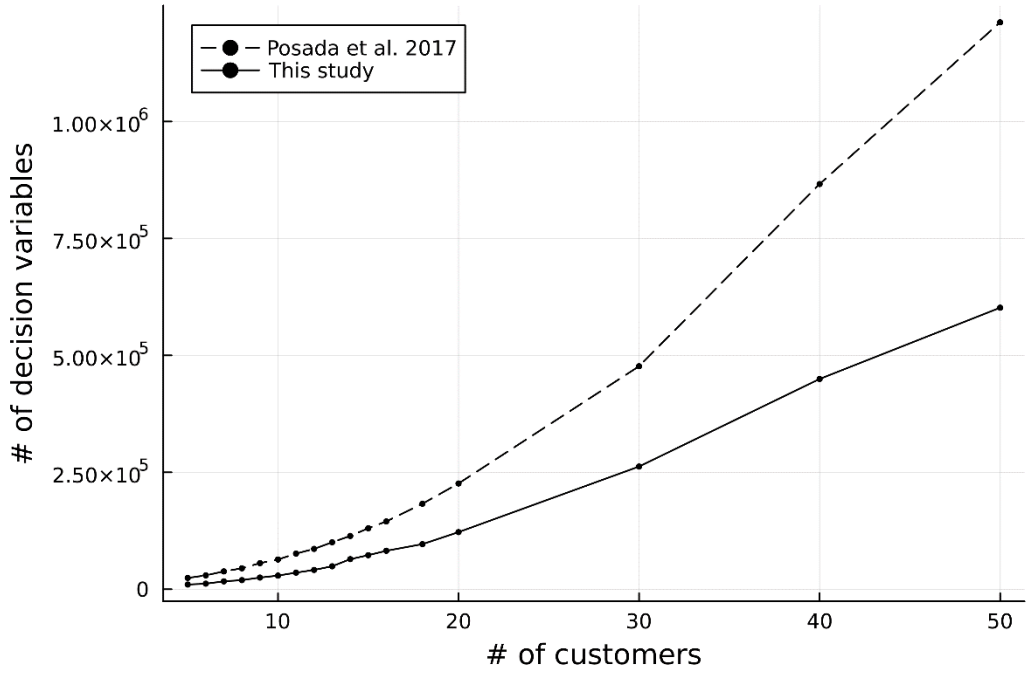


Figure 3.6: Comparison of the number of decision variables between Posada et al. (2017) (strengthened model) and our MILP formulation based on the departure-expanded graph approach

3.4.3 Results for EIDARP

We solve the EIDARP test instances with up to 50 customers for Scenario 1. The initial SoC of buses is set to full. The computational results for the models with and without applying the strengthening preprocessing and constraints are shown in Table 3.4. As all buses have full initial SoC, no charging operations are observed in the obtained solutions. Within a 4-hour computational time, the strengthened model solves the instances of up to 18 customers, while the original model can only find optimal solutions for up to 12 customers. For instances solved optimally for both cases, the strengthened model saves around 70% computational time. We use the strengthened model for the remaining numerical studies.

Table 3.4: Computational results for the original and strengthened models

# of cus.	Original model			Strengthened model			
	Obj. value[1] (min)	Gap to lower bound	CPU (s)	Obj. value[1] (min)	Gap to lower bound	CPU (s)	CPU time reduction[2]
6	**335.48**	**0.0%**	58	**335.48**	**0.0%**	2	-95.5%
8	**363.03**	**0.0%**	1727	**363.03**	**0.0%**	65	-96.2%
10	**461.54**	**0.0%**	2482	**461.54**	**0.0%**	464	-81.3%
12	**564.65**	**0.0%**	6060	**564.65**	**0.0%**	4136	-31.7%
14	594.31	2.1%	14400	**594.31**	**0.0%**	1077	-[3]
16	627.23	4.3%	14400	**627.23**	**0.0%**	11458	-
18	637.51	3.0%	14400	**637.51**	**0.0%**	11789	-
20	898.27	22.8%	14400	890.17	12.3%	14400	-
30	1412.56	29.5%	14400	1324.13	17.9%	14400	-
40	3767.87	76.2%	14400	2434.30	48.2%	14400	-
50	*[4]	*	14400	5702.89	71.4%	14400	-

Notes: 1. Based on a maximum computational time of 4 hours. Optimal solution is in bold. 2. CPU time reduction compared to the original model. 3. Not available if both models are not optimally solved. 4. Upper bound not found after 4 hours.

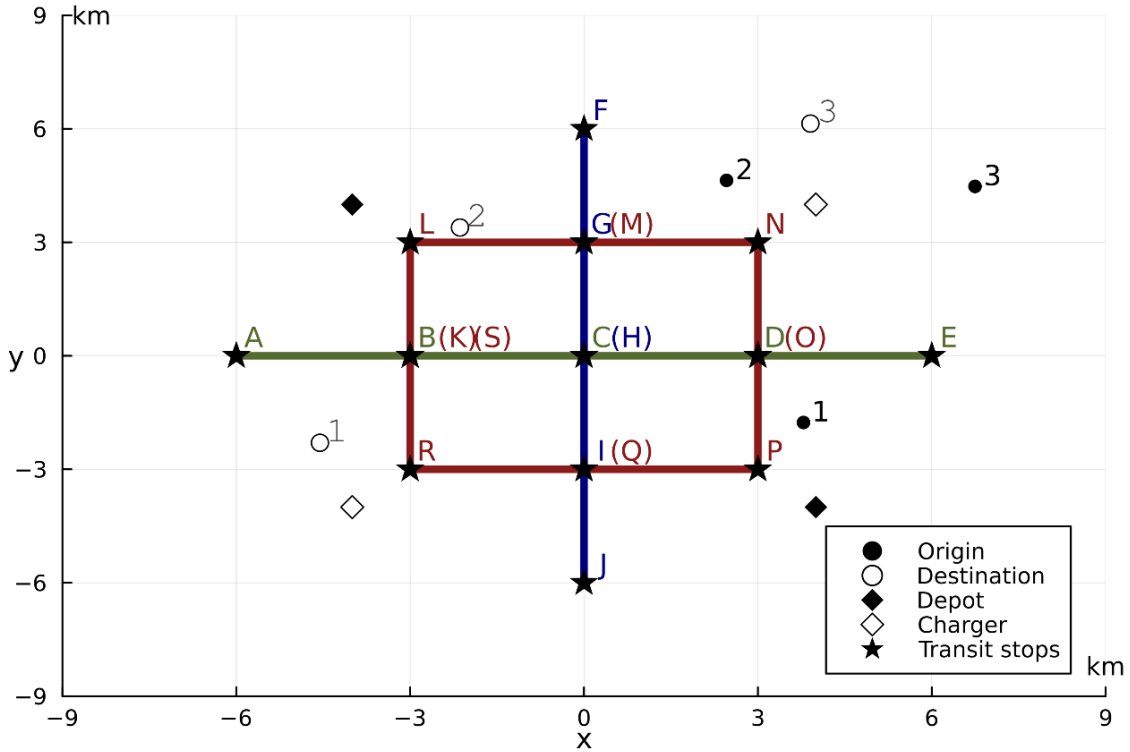


Figure 3.7: Example of a test instance in Scenario 2 with a three-line transit network

We further conduct the computational study for scenario 2 and compare the results with Scenario 1 with up to 50 customers. Scenario 2 represents a more generalized transit network with a ring-shaped line compared to the simpler one in Scenario 1. Scenario 2 involves three transit lines operating in both directions. Line 1 connects stations A through E, line 2 serves stations F through J, and line 3 links stations L through S. Figure 3.7 shows the configuration of three transit lines, each with four departures

of trains similar to Scenario 1. Transfer stations are located at the intersections of these transit lines. There are two depots and two DC fast chargers. The number of transit arcs A_G in Scenario 1 is 32, while there are 413 transit arcs in Scenario 2. The computational results are reported in Table 3.5. The test instances with up to 8 customers can be optimally solved. For larger instances with up to 30 customers, the optimality gaps range from 4.1% to 84.6%, and no feasible solutions are found for 40 and 50 customers. Compared with the results obtained from Scenario 1 on the right part of Table 5, it shows that the transit network complexity has significant impact on the computational time and the optimality gaps.

Table 3.5: Computational results for three-line transit network (Scenario 2) compared to two-line transit network (Scenario 1)

# of cus.	Scenario 2			Scenario 1		
	Obj. value (min)	Gap to lower bound	CPU (s)	Obj. value (min)	Gap to lower bound	CPU (s)
6	**310.16**	**0.0%**	1583	**335.48**	**0.0%**	3
8	**385.77**	**0.0%**	8362	**363.03**	**0.0%**	66
10	442.07	7.1%	14400	**461.54**	**0.0%**	465
12	552.69	4.1%	14400	**564.65**	**0.0%**	4137
14	617.73	5.7%	14400	**594.31**	**0.0%**	1078
16	619.64	12.0%	14400	**627.23**	**0.0%**	1149
18	648.63	8.7%	14400	**637.51**	**0.0%**	11789
20	892.22	16.3%	14400	890.17	12.3%	14400
30	3000	84.6%	14400	1324.13	17.9%	14400
40	*	*	*	2434.30	48.2%	14400
50	*	*	*	5702.89	71.4%	14400

Note: Based on a maximum computational time of 4 hours. The optimal solution is in bold. * means that the upper bound cannot be found after 4 hours.

3.4.4 Experiments for charging operations

As no charging operations are observed up to 50 customers when setting buses' initial SoC to full battery capacity (see Table 3.6), we conduct some experiments with low initial SoC of buses. Furthermore, we test the efficiency of the proposed CSCC modeling approach. Table 3.6 shows the impacts of the low initial SoC of buses on the optimality gaps for the test instances of Scenario 1 with up to 50 customers. We set the initial SoC of buses as 30% of the battery capacity (Low SoC) to activate charging operations. The results are compared with the case with fully recharged buses before starting the services. Within a 4-hour computational time, the test instances for low initial SoC setting can be optimally solved with up to 8 customers, while no feasible solution found for 40 and 50 customers. In contrast, test instances with full initial SoC settings can be solved optimally up to 20 customers. Customer rejection occurs only in the 30-customer instance with low-SoC, resulting in lower bus travel time and customer travel time compared to the full-SoC case. No customers are rejected in instances with 8 to 20 customers in either case. However, to serve all customers, more buses need to be used for the low SoC scenario. We can observe that charging operations are performed at the test instances with 6, 10, 12, 20 and 30 customers.

Table 3.6: Results of the low SoC and full SoC for Scenario 1

# of cus	Low SoC							Full SoC[2]			
	Obj. value (min)[1]	Gap to lower bound	CPU (s)	# used bus	BTT[3] (min)	CTT[3] (min)	CT[3] (min)	Obj. value (min)[1]	# used bus	BTT compare[4]	CTT compare[4]
6	**337.56**	**0.0%**	240	4	207.26	130.29	32.10	**335.48**	4	-2.9%	3.2%
8	**363.04**	**0.0%**	1125	4	198.89	164.14	0.00	**363.03**	4	0.0%	0.0%
10	473.24	4.9%	14400	6	285.22	188.01	3.49	**461.54**	6	-4.0%	-0.2%
12	614.19	16.5%	14400	8	399.53	214.65	6.06	**564.65**	6	-15.3%	5.4%
14	596.76	4.0%	14400	7	318.22	287.53	0.00	**594.31**	6	-0.8%	0.0%
16	649.55	11.5%	14400	7	369.34	280.20	0.00	**627.24**	7	-4.2%	-2.5%
18	657.13	3.0%	14400	9	377.56	279.56	0.00	**637.51**	6	-9.6%	6.0%
20	1081.21	31.0%	14400	14	694.55	386.66	7.45	890.17	9	-32.3%	8.7%
30	2198.05[5]	71.6%[5]	14400[5]	11[5]	481.62[5]	216.43[5]	2.61[5]	1324.13	11	51.1%	175.6%

Notes: 1. The computational time limit is 4 hours. The optimal solutions are in bold.

2. Results retrieved from the strengthened model in Table 3.4.

3. BTT: Total bus travel time; CTT: Total customers' travel time; CT: Total charging time.

4. BTT and CTT changes compared to that of Low SoC.

5. 15 customers are rejected for this instance in low SOC. Note that the upper bounds are not found after 4 hours for 40 and 50 customers for the low SoC case.

Another numerical study is conducted to demonstrate the efficiency of the proposed CSCCs (Eq. (3.39) – (3.42)). The results are compared with the standard replication-based CSCC formulation (Bruglieri et al., 2019; Ma et al., 2024). To activate multiple bus charging operations, we design a very simple scenario in which buses move between two locations (origin and destination) with low initial SoCs at their origin and need to derivate their routes for en-route recharge. Buses do not have customer service, but each bus needs to be recharged multiple times with one charger. To simulate a more realistic situation, a visit to the nearest and unvisited not-charger nodes is enforced between every two recharge operations (see Figure 3.8). The number of buses varies from 5 to 25 with the same initial SoCs at the beginning of service (13% battery capacity). To evaluate the impact of the two different CSCC formulations, we set up three scenarios with 1, 2, and 3 mandatory charging operations per bus, respectively. As there is only one charger, the standard replication-based CSCC formulation needs to create $m|K|$ dummy charger nodes (m is the number of charging operations per bus) compared to m for our new formulation. The objective function is revised as to minimize the total bus travel time. Each bus ends up with the same route with different beginning time of service at the charger due to the capacity constraint.

The results are shown in Table 3.7. When the number of visits and number of buses increase, using the standard replication-based CSCC formulation would significantly increase the computational time. All test cases with up to 25 buses and 3 charging operations per bus can be optimally solved within less than 6 minutes using Eqs. (3.39) – (3.42), while standard replication-based CSCC approach can solve to optimal for 2 charging operations. Figure 3.9 compares the number of decision variables for the two CSCC modelling approaches. We can observe that the number of decision variables increases dramatically with high charging demand. The results confirm the benefits of this new CSCC formulation when charging demand is high.

Table 3.7: One CS with different number of visits

# of buses	The proposed CSCCs		Standard replication-based approach	
	Optimal obj. value (min)	CPU (s)	Optimal obj. value (min)	CPU (s)
One charging operation / bus				
5	72.00	1.0	72.00	1
10	144.00	1.8	144.00	3
15	216.00	2.9	216.00	7
20	288.00	4.9	288.00	8
25	360.00	6.1	360.00	14
Two charging operations / bus				
5	129.08	0.8	129.08	1
10	258.16	4.8	258.16	43
15	387.24	2.8	387.24	113
20	516.32	6.8	516.32	2912
25	645.41	8.7	645.41	567
Three charging operations / bus				
5	209.30	5.3	209.30	274
10	418.61	28.5	*[1]	7200
15	627.91	42.1	*	7200
20	837.22	382.5	*	7200
25	1046.52	337.7	*	7200

Note: The computational time limit is 2 hours. 1. No feasible solution found after the computational time limit.

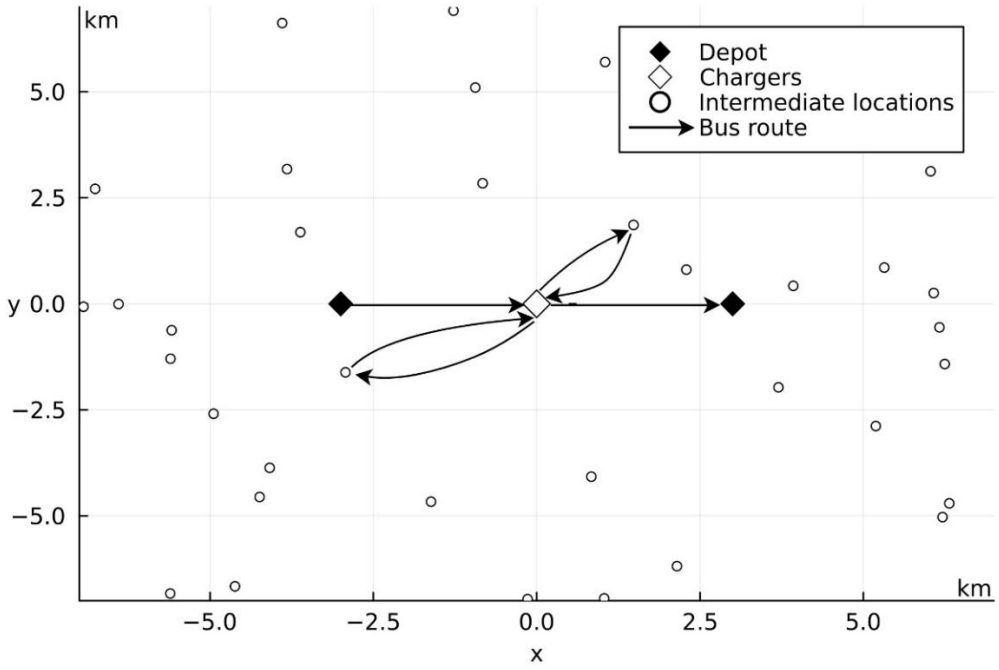


Figure 3.8: Illustrative example of buses' three charging operations with two intermediate stop visits

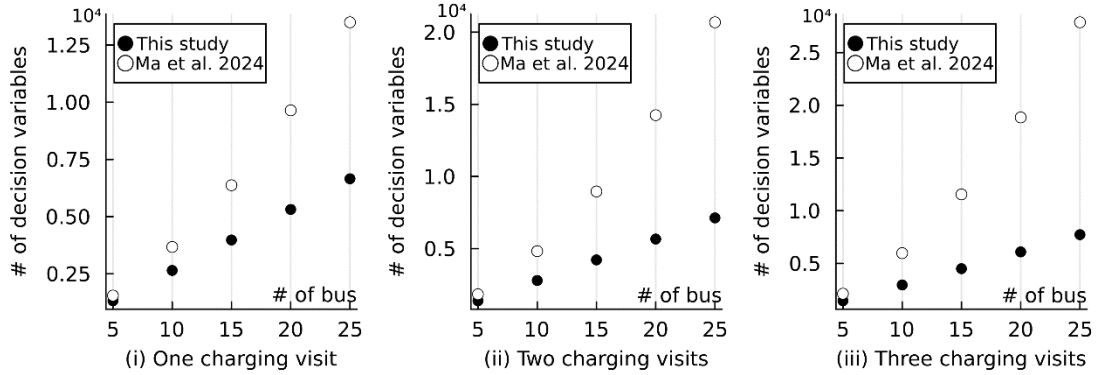


Figure 3.9: Number of decision variables compared to the standard replication-based CSCC formulation

3.4.5 Comparison to a standard door-to-door DARP

To understand whether the integrated service is beneficial for operators and customers, this section compares the results of EIDARP of Scenario 1 with full initial SoCs and standard door-to-door E-DARP. We consider the test instances with up to 18 customers within a 4-hour computational time that EIDARP can be solved optimally. The results in Table 3.8 shows that the solutions of EIDARP for all test instances have no customer rejections, while for E-DARP, one customer rejection is observed for two test instances. Thanks to the integrated service, part of buses' route is replaced by transit services. The bus service rate of EIDARP becomes higher (shorter trip duration) and more customers can be served to meet their time window or maximum ride time constraints. On average, IDARP reduces by 12.3% bus travel time and increases 3.2% customer travel time compared to DARP under Scenario 1. For the instance with 12 customers, EIDARP provides 6.6% more bus travel time. This is because E-DARP service rejects 1 customer, while the EIDARP serves all 12 requests. It is reasonable to observe higher customer journey time with an integrated service compared to that of door-to-door service. However, the total customer travel time of EIDARP does not increase too much due to maximum ride time constraint. Compared to EDARP, the objective function values (overall system costs) of EIDARP are much lower for all test instances.

Table 3.8: Comparison results for E-DARP and EIDARP.

# of cus	EDARP					EIDARP			
	Obj. value (min)	BTT[1] (min)	CTT[2] (min)	# rej[3]	CPU (s)	Obj. value (min)	BTT Compare[4]	CJT Compare[4]	# cus transit[4]
6	**340.08**	220	120	0	1	**335.48**	-8.7%	12.3%	2
8	**452.49**	261	191	0	1	**363.03**	-23.8%	-14.2%	4
10	**539.15**	279	161	1	1	**461.54**	-1.7%	16.9%	6
12	**618.31**	317	201	1	71	**564.65**	6.6%	12.5%	6
14	**720.91**	438	283	0	321	**594.31**	-27.8%	-1.7%	9
16	**679.61**	405	275	0	262	**627.24**	-12.6%	-0.5%	5
18	**721.41**	417	305	0	1367	**637.51**	-18.1%	-2.7%	6

Notes: The computational time limit is 4 hours. All test instances are optimally solved. 1. BTT: Total bus travel time. 2. CTT: Total customer travel time. 3. # rej: Number of rejected customers. 4. BTT and CTT changes compared to that of EDARP. 5. CT: Number of customers using transit services.

3.5 Discussion and conclusions

In this chapter, we developed a new methodology based on the departure-expanded graph for modeling a multi-depot IDARP using EVs. This methodology extends the existing IDARP modeling approach (Posada et al., 2017) by allowing customer rejections and operating with a set of EVs. Different from prior works, our objective function considers the trade-off between operational costs and customers' inconvenience. Furthermore, customer travel times are minimized with a maximum inter-modal transfer time at transit stations. By exploiting timely transfer requirement, the departure-expanded graph approach allows the reduction of the problem size with time window tightening. Furthermore, an efficient formulation is proposed for EVs' charging schedule modelling under capacitated charging station constraints and partial recharge.

The computational studies are conducted on a set of test instances with two-line and three-line transit networks and with customer demand of up to 20 requests. With a computational time limit of 4 hours, the proposed approach finds optimal solutions for EIDARP with up to 18 customers (two-line transit network, full initial SoCs of buses). For IDARP, our numerical result shows that the proposed departure-expanded graph approach can solve up to 12 requests with a fractional computational time compared to Posada's modelling approach (Posada et al. (2017). For electric charging operation modeling, our CSCC formulation allows a charger to be visited by multiple vehicles and reduce significantly the problem size compared to the standard replication-based CSCC formulation. Our numerical study on a high demand charging operation scenario shows that the proposed method can reduce substantial computational time. When comparing the performance of EIDARP and the standard DARP services, the results show that the integrated service can reduce the number of used buses and operational costs.

Although the proposed modelling approach outperforms the state-of-the-art, the instances it can solve remain limited, making it impractical for evaluating the application of EIDARP in real-world scenarios. To address this limitation, efficient solution approaches, such as heuristic or metaheuristic approaches, will be developed to handle larger problem sizes effectively in the next Chapter.

Chapter 4 The hybrid large neighborhood search for the electric integrated dial-a-ride problem

This chapter is based on the following working paper: *Fang, Y., Ma, T.-Y., 2025. A hybrid large neighborhood search algorithm for the integrated dial-a-ride problem using electric vehicles. (under review for Transportation Research Part E: Logistics and Transportation Review).*

Chapter 3 proposed an efficient MILP formulation of the EIDARP which can be solved exactly with the problem size up to 20 customers. More efficient solution approach is necessary to tackle real-world problem size. In this chapter, we develop a hybrid LNS metaheuristic for EIDARP, building on the MILP formulation of Chapter 3, which aims to minimize both customer's and operator's costs (time). To further reduce customer inconvenience, the maximum travel time constraint per customer is defined as the difference between the customer's arrival time at destination and departure times at origin. It ensures that the overall waiting time for each customer is also bounded within this constraint to improve customers' service experiences.

The developed hybrid LNS introduces several key features tailored to the considered problem: 1) a departure-expanded graph is employed to manage the synchronization between on-demand buses and transit departures; 2) a set of LNS operators hybridized with the problem-specific local search operators is designed specifically for the EIDARP; 3) the solution evaluation scheme extends the eight-step scheme proposed for DARP (Parragh, 2011) to a nine-step scheme, adapted for the integrated problem using electric buses; 4) a specific approach is implemented to address the capacitated charging station when optimizing bus routing and charging operations. The hybrid LNS is benchmarked against the solutions obtained by Gurobi MILP solver, demonstrating significant advantages in efficiency and solution quality.

The remainder of this paper is organized as follows. A literature review is presented in Section 4.1. The hybrid LNS algorithm is detailed in Section 4.2. Section 4.3 focuses on the algorithm's parameters tuning and computational experiments, and Section 4.4 draws the conclusions.

4.1 Literature review

The literature review begins with an exploration of related IDRT studies, with a focus on the solution methodologies. Given that the challenges of the (E)IDARP, compared with DARP, primarily involve managing multi-vehicle customer journeys and synchronizing busses' arrivals with transit departures at transit stations, Section 4.1.2 reviews the feasibility evaluation procedures for both DARP and IDARP. Section 4.1.3 provides a short review of LNS metaheuristics and related hybridization techniques to improve the algorithm's intensification.

4.1.1 IDRT and solution methods

Early studies focus on integrating paratransit services with fixed bus lines as a first-mile/last-mile services. Liaw et al. (1996) formulated this integration as a bi-modal DARP on a MILP formulation, in which an on-line and off-line decision support system was developed to automate the routing and scheduling process. Hickman and Blume (2001) investigated a similar problem but further emphasized on customer inconvenience including constraints for maximum travel time and number of transfers. They proposed a two-stage heuristic: the first stage identifies whether customer itineraries can be assigned to fixed-route services, and the second schedules paratransit legs for the customers identified in the first stage. A case study involving thousands of one-way passenger trips demonstrated that more than half of the passenger trips could be accommodated using the proposed heuristic. The heuristic was extended by Aldaihani and Dessouky (2003) by incorporating a Tabu search procedure to improve vehicle routing without increasing customer travel time. The algorithm was tested on a real paratransit service provider with up to 155 requests. The results demonstrated a 16.6% reduction in VKT, albeit with an 8.7% increase in customer travel time.

The MILP formulation of IDARP was first brought up by Häll et al. (2009), and Posada et al. (2017) extended this formulation by incorporating timetabled transit, which synchronizes transit departures and arrivals with on-demand vehicles. The authors later proposed an Adaptive Large Neighborhood Search (ALNS) method to solve this problem, with test instances up to 145 requests on one transit line (Posada and Häll, 2020). Molenbruch et al. (2021) developed an LNS for a similar problem, though it excluded walking behavior. They introduced a scheduling procedure to manage the synchronization with public transport and to address the maximum ride time constraints effectively. The algorithm was benchmarked using the DARP instances of Braekers et al. (2014), and demonstrated quality improvements. Moreover, the study conducted a comprehensive analysis on the performance of IDARP from across different demand distribution and the operators' characteristics, including the frequency and speed of public transportation, and maximum detour factor for ride time. Melis et al. (2024) investigated an on-demand bus routing problem with the objective of minimizing customer travel time, in which maximum travel time was excluded. The problem assumes high-frequency transit services, eliminating the need for transit synchronization. The authors developed an insertion-based solution approach that includes a speed-up procedure for assigning metro segments. Fang et al. (2024) were the first to incorporate EVs into IDARP and developed a MILP for the EIDARP, accounting for capacitated charging station constraints. A departure-expanded transit network was employed to reduce the problem size, and a set of valid inequalities was proposed to tighten the search space. However, this approach is limited to solve the test instances with up to 20 customers and two transit lines.

Ghilas et al. (2016) adapted the IDARP for freight transportation, formulating it as a PDP with scheduled lines. The authors proposed an ALNS approach, incorporating a repair step for transfer nodes during the insertion stage, where replicated transfer nodes are greedily inserted. This method was tested on instances with up to 100 requests and one transit line. Subsequently, Ghilas et al. (2018) introduced a branch-and-price algorithm for the same problem, enabling the computation of optimal solutions for instances with up to 50 requests and two transit lines.

4.1.2 Feasibility evaluation for DARP and IDARP

The eight-step scheme proposed by Cordeau and Laporte (2003), and later modified by Parragh et al. (2010), has been widely applied in the PDP and DARP. This method efficiently evaluates each position in a route for potential constraint violation in terms of vehicle capacity, time windows and maximum

ride time of customers. If a time-related violation is detected, a time delay process is initiated by calculating forward slack times (maximum potential delay time) for subsequent positions of the route to repair its infeasibility. However, Braekers et al. (2014) pointed out that this approach increases computational complexity and suggested a preliminary check on these constraints to reduce unnecessary computations. Gschwind and Drexl (2019) proposed a constant-time feasibility check that evaluates only the pickup and delivery nodes when a request is inserted. The approach achieves an average speedup factor of 3.8 compared with the eight-step scheme by Cordeau and Laporte (2003).

Feasibility evaluation for IDARP presents unique challenges due to the need for synchronization with transit departures and the possibility of splitting customer trips across two on-demand vehicles for first-mile and last-mile connections, complicating the enforcement of maximum ride time constraints of customers. Hickman and Blume (2001) addressed these challenges by identifying potential transfer points at each customer's pickup and drop-off location, treating the customer journey as a shortest path problem with time windows constraints. Aldaihani and Dessouky (2003) tackled the transfer synchronization by predefining each customer's feasible path, including a transit origin-destination pair. However, the problem did not include the maximum ride time constraint, as the objective of this problem minimizes total customer travel time. Posada and Häll (2020) applied the constant-time feasibility check of Gschwind and Drexl (2019), but did not clarify how their approach was adapted to account for transit synchronization and split trips for IDARP. Molenbruch et al. (2021) formulated an exact scheduling problem for IDARP as a simple temporal problem, a framework used for modeling and solving problems that involve temporal constraints by representing events and their relationships as time intervals. This approach, applied in the solution feasibility evaluation step, integrates constraints such as time windows, maximum ride time, and transit synchronization. The scheduling problem is solved using an extended Bellman-Ford algorithm. This approach builds on the DARP with transfers, where customer's trips are also served by different vehicles (Masson et al., 2014).

As we consider an IDRT service using EVs, recharging operations must be nevertheless examined at each position of a bus route. The forward slack time introduced by Cordeau and Laporte (2003) can be utilized to determine the potential recharging positions, an approach successfully implemented by Ma et al. (2024). Consequently, this chapter extends the eight-step evaluation scheme to accommodate the requirements of EIDARP.

4.1.3 Large neighborhood search and its hybridization

LNS was initially introduced by Shaw (1998) for solving the VRP, where neighborhoods are explicitly defined by "destroy" (remove parts of the solution) and "repair" (reinsert the removed part of the solution) operators. The Adaptive LNS (ALNS) extends this approach by dynamically selecting destroy and repair operators based on a weighting system (Ropke and Pisinger, 2006). ALNS has become widely popular and successfully applied across various routing problems, such as vehicle routing, location routing, PDP and DARP (Mara et al., 2022). When comparing ALNS and LNS, Turkeš et al. (2021) investigated the benefit of the adaptive layer on solution quality and found it to improve results only by approximately 0.14% on average.

Unlike Variable Neighborhood Search (VNS), which systematically alternates between diversification and intensification, (A)LNS is a metaheuristic that primarily leverages diversification. Hybrid approaches that integrate intensification strategies with LNS have been explored in the literature. For instance, Alinaghian and Shokouhi (2018) incorporated improvement phases using VNS within the framework of ALNS to solve the multi-depot multi-compartment VRP. Şatir Akpunar and Akpinar

(2021) also hybridized ALNS via VNS in the capacitated location routing problem, achieving enhancements both in computational time and quality compared with the standalone ALNS. Another approach to balance diversification and intensification of LNS is through an acceptance criteria such as simulated annealing or DA (also known as threshold accepting). These methods increase the probability of accepting worse solutions at higher temperatures (diversification) and decrease it at lower temperatures (intensification). Santini et al. (2018) compared nine different acceptance criteria for ALNS and found simulated annealing, DA and record-to-record travel (acceptance criteria based on the difference between new and the best solution) outperformed the other methods.

(A)LNS has also been applied to integrated on-demand mobility services in both freight transportation (Ghilas et al., 2016) and to passenger transportation (Molenbruch et al., 2021; Posada and Häll, 2020). These studies introduced specific destroy and repair operators tailored for transit integration. For instance, Ghilas et al. (2016) and Molenbruch et al. (2021) involved customers' assigned transit departure times to evaluate the similarity between pairs of customers during the destroy stage, and considered transit stations at both first-miles and last- miles during the repair stage. Posada and Häll (2020) proposed two IDARP-specific operators: the first repair operator examines the possibility of introducing a fixed-line leg into a customer trip, while the second destroy operator removes the integrated customer routes in descending order of cost. These studies tested their algorithms on real-world problem instances and benchmarked with instances of DARP or PDP. However, none of these studies incorporate separate local search procedures for further improve the solution quality.

Table 4.1 summarizes the related IDRT services discussed in this section. Among the studies reviewed, the proposed EIDARP is the only one to date to consider a weighted sum of both operator and customer costs and to include EVs in the integrated on-demand services. To address multi-vehicle customer trips and synchronization between demand-responsive buses and transit, the conventional eight-step evaluation scheme (Parragh et al., 2010) is extended to a nine-step process, through an additional step for recharge scheduling at capacitated charging stations. In terms of solution algorithms, three previous studies applied (A)LNS to their developed integrated on-demand services, utilizing tailored destroy and repair operators for the specific characteristics of transit integration. This chapter also employs LNS as the solving method. However, given that Turkeš et al. (2021) reported only 0.14% improvement in solution quality with the adaptive layer of ALNS, it is not incorporated here. Instead, a hybrid LNS is developed to enhance intensification, incorporating several local search operators and a DA criterion to improve solution quality. Dedicated LNS operators and local search operators are also developed for the EIDARP.

Table 4.1: Related IDRT studies and solution methods

	P/ F[1]	Obj.[2]	ST[3]	MD[4]	HV[5]	MR[6]	Exact/ heuristic	Solution method	Instances
Liaw et al. (1996)	P	a	✓				Heuristic	On-line and off-line decision support system	Up to 120 requests
Hickman and Blume (2001)	P	d	✓			✓	Heuristic	Two-stage heuristic	Multiple transit lines, 3588 on-way requests
Aldaihani and Dessouky (2003)	P	a or b	✓				Heuristic	Tabu search	Multiple transit lines, up to 155 requests
Häll et al. (2009)	P	a				✓	Exact	MILP solver	One transit line, 4 requests
Ghilas et al. (2016)	F	ab	✓	✓	✓		Metaheuristic	ALNS	One fixed line, up to 100 requests
Posada et al. (2017)	P	ab	✓		✓	✓	Exact	MILP solver	One transit line, up to 6 requests
Ghilas et al. (2018)	F	ab	✓	✓	✓		Exact	Branch-and-price algorithm	Two transit lines, up to 50 requests
Posada and Häll (2020)	P	ab	✓		✓	✓	Metaheuristic	ALNS	One transit line, up to 145 requests
Molenbruch et al. (2021)	P	ab	✓	✓	✓	✓	Metaheuristic	LNS	Multiple transit lines, 200 requests
Melis et al. (2024)	P	c					Heuristic	Insertion-based heuristic	Multiple transit lines, up to 2000 requests
This study	P	ac	✓	✓	✓	✓	Metaheuristic	Hybrid LNS	Multiple transit line, up to 100 requests

Notes: 1. P: passenger transport; F: freight transport. 2. Objective function (minimize): a. travel time (costs) of demand-responsive vehicles, b. travel time (costs) of transit line, c. customers' journey/travel time, d. generalized cost of customer convenience. 3. Synchronized transfer. 4. Multiple depots. 5. Heterogeneous vehicles. 6. Maximum ride (travel) time constraint.

4.2 Solution algorithm

In this section, we propose a hybrid LNS to solve the EIDARP as described in Algorithm 4. The hybrid LNS integrates problem-specific local operators and DA acceptance criteria to balance its diversification and intensification. An initial solution s_{init} is constructed and set as both the incumbent solution s and the current best solution s_{best}. In the next step, a destroy (i.e. remove customers from bus routes) and repair (i.e. insert unserved customers to bus routes) operator is randomly selected and applied to s, yielding s'. Subsequently, several local search operators are applied to s' to obtain the local optimal solution s''. The local search procedure is only activated when s' is promising, i.e. $c(s') \leq \alpha c(s_{best})$, where a smaller α ($\alpha > 1$) suggests a less-frequent local search. A DA acceptance criteria decides whether s'' should be accepted as the next incumbent solution. At each iteration, s_{best} is updated if objective function value $c(s'') \leq c(s_{best})$ (line 16-18). The algorithm terminates when the maximum iteration is reached, and s_{best} is returned as the final solution.

In this algorithm, only feasible solutions determined by the solution evaluation scheme are accepted when a new solution is constructed. The feasibility evaluation is particularly challenging due to the incorporation of multiple travel options for customers' journeys optimization. The five travel options of customers defined in Section 3.2.1 introduce a significant difference from DARP, impacting on both the solution evaluation process and the design of destroy, repair, and local search operators. Furthermore, as EIDARP manages recharging scheduling, an additional step is necessary to tackle recharging effectively. Algorithm 4 provides detailed descriptions of these steps.

Algorithm 4. Hybrid LNS

Construct s_{init}
$s, s_{best} \leftarrow s_{init};\ i = 1$
repeat
  randomly select a destroy operator $d_i \in \{d_{random}, d_{worst}, d_{relate}, d_{route}\}$
  randomly select a repair operator $r_i \in \{r_{random}, r_{greedy}, r_{regret}, r_{Tpriority}, r_{TP}\}$
  $n^{destroy} = rand(1, \lceil n \times \xi_{max} \rceil)$
  s, destroyed customers $\leftarrow d_i(s, n^{destroy})$
  $s' \leftarrow r_i(s,$ unserved customers ∪ destroyed customers))
  if $c(s') < \alpha \times c(s_{best})$
    $s'' \leftarrow LocalSearch(s')$
  end if
  $T = T - T_{max}/T_{red}$; if $T \leq 0$, set $T = T_{max}$
  if $c(s'') < c(s_{best}) + T$
    $s \leftarrow s''$
  end if
  if $c(s'') < c(s_{best})$
    $s_{best} \leftarrow s''$
  end if
until $i = n_{iter}$
return s_{best}

4.2.1 Preprocessing and initial solution

As discussed in Section 3.2.2, each transit node in the departure-expanded transit graph $\mathcal{G}_T$ is associated with specific departure and arrival times. Recall that a transit arc (TA) (i, j) is defined as a feasible shortest route connecting two transit station nodes from i to j, provided it satisfies predefined maximum transfer time constraints for transfers between different scheduled train departures at transfer stations. The arc weight represents the shortest travel time between the nodes, based on the timetables of the transit network. To accelerate the algorithm's search for customer trips, a set of potential TAs is identified for customers who can use transit as a part of their journeys. The TAs of customer $r \in R$, denoted as $\mathcal{T}_r$, are determined based on the following criteria, considering traveling by bus and walking:

- At customer r's **origin** with time window $[e_r, l_r]$, a TA (i, j) can be added to customer r travelling by **bus** to transit node i (travel option 2 and 4) if $e_r + t_{ri} \leq \underline{\theta}_i$ and $l_r + t_{ri} \geq \underline{\theta}_i - \gamma$
- At customer r's **destination** with time window $[e_{2n+r}, l_{2n+r}]$, a TA (i, j) can be added to customer r traveling by **bus** from transit node j (travel option 3 and 4) if $\bar{\theta}_j + t_{j,2n+r} \leq l_{2n+r}$ and $\bar{\theta}_j +$

$t_{j,2n+r} + \gamma \geq e_{2n+r}$.

- At customer r's **origin** with time window $[e_r, l_r]$, a TA (i, j) can be added to customer r **walking** to transit node i (travel option 1 and 3) if $e_r + t_{ri}^w \leq \underline{\theta}_i, l_r + t_{ri}^w \geq \underline{\theta}_o - \gamma$, and t_{io}^w is within the maximum walking time of customers (identical for each customer).
- At customer r's **destination** with time window $[e_{2n+r}, l_{2n+r}]$, a TA (i, j) can be added to customer r **walking** from transit node j (travel option 1 and 2), if $\bar{\theta}_j + t_{j,2n+r}^w \leq l_{2n+r}$, $\bar{\theta}_j + t_{j,2n+r}^w + \gamma \geq e_{2n+r}$ and $t_{j,2n+r}^w$ is within the maximum walking time.

The time-window at a transit node i for bus $k \in K$ depends on three different situations. These time-windows are different in terms of arrival time A_i^k and beginning time of service B_i^k:

- If transit node i has customers transferring from bus to transit services on bus k, $\underline{\theta}_i - \gamma \leq A_i^k \leq \underline{\theta}_i$.
- If transit node i has customers transferring from transit to bus services on bus k, $\overline{\theta}_i \leq B_i^k \leq \overline{\theta}_i + \gamma$.
- If transit node i has both customers transferring between bus and transit service on bus k, it needs to satisfy $\underline{\theta}_i - \gamma \leq A_i^k \leq \underline{\theta}_i$ and $\overline{\theta}_i \leq B_i^k \leq \overline{\theta}_i + \gamma$.

As these time windows change dynamically during the solution search process, the time window of transit node i in the preprocessing step is defined as the union of the three situations mentioned above, i.e., $[\underline{\theta}_i - \gamma, \overline{\theta}_i + \gamma]$. Additional steps to handle the dynamic time windows is discussed in Section 4.2.4. The remaining procedures regarding time-window tightening and arc elimination procedures for bus subgraph $\mathcal{G}_B$ are consistent with the method proposed in Section 3.3.1.

An initial solution is constructed by prioritizing customers with a positive number of TAs, aiming to reduce bus travel costs in the objective function and ensure the quality of the initial solution. The process begins by serving customers within walking distance of transit stations at both their origins and destinations (travel option 1). Next, customers whose origin or destination is within the maximum walking distance (travel options 2 and 3) are served. Customers whose origin and destination are not within walking distance but has a positive number of TAs are service afterwards, with both their first- and last- mile served by bus. Lastly, the remaining customers are served exclusively by bus.

4.2.2 Destroy and repair operators

The shaking step consists of destroy and repair procedures, with most operators having already been extensively applied in existing studies to different variants of VRPs (Mara et al., 2022). However, for solving the EIDARP problem, three tailored operators, i.e., $d_{relate}, r_{Tpriority}$ and r_{TP}, are proposed. In the destroy procedures, $\xi_{max} \in (0,1)$ is a parameter to setup the degree of destroy of a current solution which determines the size of the search neighborhood. The number of destroyed customers $n^{destroy}$ is randomly selected between 1 and $\lceil n \times \xi_{max} \rceil$ to have variable search neighborhoods during the search. The set of proposed destroy operators includes:

- *Random removal* – d_{random}: randomly remove $n^{destroy}$ customers from the served customers.
- *Worst removal* – d_{worst}: remove the most expensive $n^{destroy}$ customers in terms of the objective function value from the served customers.
- *Route removal* – d_{route}: randomly remove a bus route along with its associated nodes.
- *Related removal* – $d_{related}$: remove the $n^{destroy}$ most related customers to a random selected customer from the unserved customer list in solution s. Eq. (4.1) defines the relatedness of customer

i and j, where c_{max} and t_{max} represents the distance of the longest arc in $\mathcal{G}$, and t_{max} is the planning horizon. The first term represents the spatial similarities of origins and destinations of customer i and j, and the second term computes their temporal similarities. Note that $abs(x)$ represents the absolute value of x. For customers' journeys involving transit legs, similarities of customer i's and j's TAs are incorporated as the third term by calculating the number of shared TAs, i.e. $|\mathcal{T}_i \cap \mathcal{T}_j|$ divided by the total number of TAs, i.e. $|\cup_{r\in R}\mathcal{T}_r|$. Note that $abs(A_i - A_j)$ represented the absolute value of $A_i - A_j$, while $|\mathcal{T}_i \cap \mathcal{T}_j|$ represent the number of elements in set $\mathcal{T}_i \cap \mathcal{T}_j$.

$$Relatedness_{ij} = \left(\frac{c_{ij}+c_{2n+i,2n+j}}{2c_{max}} + \frac{abs(B_i-B_j)+abs(A_i-A_j)}{2t_{max}} + \frac{|\mathcal{T}_i \cap \mathcal{T}_j|}{|\cup_{r\in R}\mathcal{T}_r|}\right)^{-1} \quad (4.1)$$

As the considered EIDARP problem allows customer rejection, the repairing procedure considers reinserting the union of customers removed by the destroy operator and the existing unserved customers of solution s'. Repairing a customer in the algorithm aims to find the best position with the lowest cost, defined as the insertion position that minimally increases the objective function value. In DARP, this process only involves iterating over all routes to determine the best position for the customer's origin and destination (travel option 5). However, for EIDARP the complexity is significantly greater due to the need to evaluate all five different travel options for each customer. In the case of EIDARP, searching for the best position includes identifying the best travel option along with its associated best insertion positions.

- *Random repair* – r_{random}: repair unserved customers in random order at the lowest-cost insertion positions.
- *Greedy repair* – r_{greedy}: iteratively select and insert a customer whose best insertion position has the lowest cost among the unserved customers.
- *Regret repair* – $r_{k-regret}$: iteratively select and insert a customer who has most regret value, which is calculated as the accumulative difference between the objective function value at the best insert position up to the k^{th}-best insertion position. In this algorithm, k is chosen as 2 and 3.
- *Transit priority repair* – $r_{Tpriority}$: repairs unserved customers by prioritizing those who have options to use trains (positive number of TAs of customers), thereby reducing the bus travel costs in the objective function.
- *TA priority repair* – r_{TP}: repairs multiple customers simultaneously if they share the same TA(s) as they might have higher likelihood to share the same first-mile/last-mile bus trips, leading to the reduction in objective function value.

4.2.3 Local search operators

Local search operators (e.g., *2-opt, 4-opt*, *relocate*) are widely used in solving DARP, PDP, and VRPs. These operators are typically designed for minimizing total vehicle travel costs to serve all requests. However, our EIDARP aims to minimize both customer and bus travel costs as a weighted sum in the objective function. The developed local search operators proposed in this algorithm place particular emphasis on the customers' trips. Five local search operators are applied as described as follows:

- *Bus exchange*: exchange a bus with another bus starting from a different depot if such exchange results in lower travel costs while keeping the visited sequences of customers unchanged.
- *Exchange TAs of customers*: for customers served by transit service, replace customers' TA with

other available options. Update the solutions if the objective function value is reduced.

- *Replace TA by bus*: for customers served by transit, evaluate whether switching them to door-to-door bus service reduces the objective function value. If so, retain the obtained new solution.
- *Re-assign first-/last- mile bus service*: for customers served by transit, explore alternative bus routes for first- or last- mile connections. Retain the new route if the objective function value is improved.
- *Replace walk with bus service*: for customers' first-mile/last-mile by walking, replace their walking arcs by bus. Retain such change if it reduces the objective function value.

4.2.4 Feasibility evaluation for the integrated bus routing

To address the specific requirements of IDARP, we extend the eight-step evaluation scheme of Cordeau and Laporte (2003) as a nine-step evaluation scheme to check the feasibility of a bus routes and customer's journeys. Furthermore, the EIDARP necessitates the incorporation of an additional energy repair procedure (Section 0). To assess potential violations of time-window, maximum travel time and bus capacity constraints, the following auxiliary variables are computed:

- A_i: Arrival time at node i, $A_i = D_{i-1} + t_{i-1,i}$
- B_i: Beginning time of service at node i, $B_i = \max(A_i, e_i)$
- W_i: Waiting time at node i, $W_i = B_i - A_i$
- D_i: Departure time at node i, $D_i = B_i + \mu_i$
- q_i: Passenger load at node i
- L_i: Customer travel time at pick up or destination node i, calculated as the difference between customer's arrival time at destinations and departure time at origins.
- F_i: Forward slack time at node i
- ρ_i: Waiting time violation at transit node i if customers are transferring from a bus to a transit service, $\rho_i = (\bar{\theta}_i - \gamma - A_i)^+$[6]; $\rho_i = 0$, others.

It should be noted that as each bus route is evaluated individually, the index of bus k is dropped for the above-mentioned variables. Cordeau and Laporte (2003) introduced F_i to minimize customers' travel time for DARP. However, in the context of IDARP, the calculation of F_i must consider transit departure synchronization and maximum waiting time constraints at transit stations, as defined in Eq. (4.2). The first term of Eq. (4.2) is the cumulative waiting time, where q denotes the last node of the route (destination depot). The second term F_j' of IDARP ensures that the arrival time at the transit node will not exceed transit's departure time if node j is a transit node with customers transferring from a bus to a transit service. The calculation of F_j' in other situations remains identical to DARP.

$$F_i = \min_{i \le j \le q} \left\{ \sum_{i<p\le j} W_p + F_j' \right\} \tag{4.2}$$

where F_j' equals:

1) $\left(\bar{\theta}_j - \max(A_j, \bar{\theta}_j - \gamma) \right)^+$, if j is a transit node with customers transferring from bus to transit service.
2) $\left(min\{ l_j - B_j, L_r^{max} - L_j \} \right)^+$, if j is a destination node of customer r.
3) $\left(l_j - B_j \right)^+$, otherwise.

[6] $(x)^+ = \max\{0, x\}$

For evaluating maximum waiting time constraint at the transit stations, ρ_i is introduced. If there are customers transferring from a bus route to a transit service, buses' arrival time should not be earlier than $\bar{\theta}_i - \gamma$. ρ_i is calculated as the earliest time window violation and serves as an indicator to decide whether it is necessary to apply a repair procedure (Step 7 in Algorithm 5).

Algorithm 5. Nine-step evaluation scheme

1: Set $D_0 = e_0, f_{delay} = false$.

2: Compute A_i, W_i, B_i, D_i and q_i for each node i on the route. If $B_i > l_i$ or $q_i > Q$ or $A_i > \overline{\theta}_i$ if i is a transit node and has customers using transit service, GO TO STEP 9.

3: Compute F_0.

4: Set $D_0 = e_0 + \min\{F_0, \sum_{0<p<q} W_p\}$.

5: Update A_i, W_i, B_i, D_i and ρ_i for each node i on the route.

6: Compute L_i for each request on the route. If all $L_i \leq L^{max}$ and $\sum_{0<i<q} \rho_i$ if i is a transit node and has customers using transit service, GO TO SETP 9.

7: For every node j that is an origin node or a transit node with customers transferring from bus to transit service:
   (a) Compute F_j. If $F_j < \rho_j$, GO TO STEP 9.
   (b) Set $W_j = W_j + F_j$; $B_j = A_j + W_j$; $D_j = B_j + \mu_j$.
   (c) Update A_i, W_i, B_i, D_i, and ρ_i for each node i that comes after j in the route and set $f_{delay} = true$.
   (d) Update L_i for each request i whose destination is after j. If all $L_i < L^{max}$ of requests whose destinations lie after j GO TO SETP 9; otherwise Go TO STEP 8.

8: If i is a destination node and $L_i > L^{max}$, repair the route where the origin node is located at by following STEP 7 if this route has not been repaired.

9: Compute changes in violations of bus load, duration, time window, travel time and energy constraints.

The nine-step evaluation (Algorithm 5) begins with setting the departure time from the depot to the earliest time window (Step 1). In Step 2, A_i, B_i, W_i, D_i, and q_i are updated, and potential violations of time window, bus capacity, and transit departure synchronization are checked. Step (3) – (6) delay departure time without violating route duration. If neither the maximum travel time nor the transit departure synchronization is not violated, the process advances directly to Step 9; otherwise Step 7 is applied to delay the departure time at origin and transit nodes. Step 7 (a) computes F_j. If $F_j < \rho_j$, the route is deemed infeasible. Cordeau and Laporte, (2003) noted that delaying the departure time at node i within $\sum_{i<p\leq j} W_p$ does not impact the arrival time of any node, but further delays would increase the arrival time of the subsequent nodes after node i as much. In DARP, delaying the departure time at an origin node is unnecessary when the corresponding destination node is on the same route, which is the reason DARP sets $W_j = W_j + \min\{F_j, \sum_{j<p\leq q} W_p\}$ at step 7 (b). However, for IDARP, customers' origins and destinations node can be served by different buses (routes). Consequently, delaying the departure time more than $\sum_{j<p\leq q} W_p$ can still help to minimize customers' travel times. Step 7 (b) therefore delays the departure time as much as possible to minimize customer travel time when j is an origin node. This step also ensures minimal waiting time at transit station if customer changes from bus to transit service at node j. Step 7 (c) updates all the indicators after node j, with ρ_i recalculated as

$\rho_i = (\rho_j - F_j)^+$. Additionally, a flag f_{delay} is set to $true$, indicating that delay procedures have already been applied to this route. If Step 7 (d) detects that $L_i > L_r^{max}$ at destination node i of customer r, Step 8 checks whether customer r's first-mile route is on another route and has not undergone delay procedures (as indicated by f_{delay}). If so, the delaying procedures are applied to minimize customer r's travel time at their first mile. Otherwise, it goes directly to Step 9.

4.2.5 Bus recharge scheduling

The energy recharging scheduling procedure is initiated immediately on detecting an energy violation in Step 9 of the nine-step evaluation scheme. This proposed procedure builds on the method outlined in Chapter 2 with adaptions to handle capacitated charging stations and to compute the available recharging time for the EIDARP context.

The amount of energy to be charged (energy violation) of the current route Δ_E is computed as $\Delta_E = E_{min} - (E_{init} - ec)$, where E_{init} and E_{min} is the initial and minimum SoC, and ec is the total energy consumption for the route examined. If $\Delta_E \leq 0$, the recharging scheduling is skipped. Otherwise, the first position i_{low} with the bus's SoC below E_{min} is identified, meaning that a recharge event must be inserted before i_{low} with the amount $\Delta_{E_{low}} = E_{min} - E_{i_{low}}$, where $E_{i_{low}}$ is the SoC at position i_{low}. Δ_E and $\Delta_{E_{low}}$ also represent the total required recharging amount for the route and minimum required recharging amount before position i_{low}, respectively. Each charging node $s \in S$ maintains a record of its charging events including visited buses, the start and end time, and the charging duration. From this information, the available time intervals of s can be easily calculated as the gaps between the recorded charging events within the service operational duration. The recharging scheduling aims to find common time intervals that are available for both the route and charging station.

The energy recharging scheduling process is described in Algorithm 6, starting with seeking a feasible recharge between the first position and i_{low}. Let I denote an ordered set of nodes of the route with $q_i = 0$. The recharging scheduling is based on a greedy insertion strategy, starting by checking the nearest recharging station (line 4). Line 5 computes the maximum potential recharging time $\delta_{v_i s}$ at position i if recharge at charging node s, which is determined by deducting F_i from the travel time difference to s and the service time at s (i.e. μ_s). Δ_E^s and $\Delta_{E_{low}}^s$ are updated by including extra energy consumption travelling to s, with the total (maximum) required charging time $\delta_{v_i s}^{max}$ and the minimum required charging time $\delta_{v_i s}^{min}$ calculated based on the charging power of s. If $\delta_{v_i s} < \delta_{v_i s}^{min}$, the position is moved to the next one; otherwise, the maximum available recharging time interval ε_{route}^i at charging position $i \in I$ is calculated (line 9). This interval is then compared with the available time intervals ε_s at charger s. If a common time interval exists with a duration δ more than $\delta_{v_i s}^{max}$, the recharging is considered successful with charging time $\delta_{v_i s}^{max}$, and this charging event is added to s (line 13-14). If a common time interval δ is between $\delta_{v_i s}^{min}$ and $\delta_{v_i s}^{max}$, the recharging event is inserted with the charging time δ, but the scheduling process continues from the next recharge position with updated Δ_E, $\Delta_{E_{min}}$ and i_{low} (line 16-18). If Δ_E is still greater than 0 at the end of the algorithm, the recharge scheduling is considered unsuccessful.

Algorithm 6. Recharging scheduling with capacitated charging stations

Input: $r = \{v_0, \dots, v_q\}, S, \Delta_E, i_{low}, \Delta_{E_{low}}$

Output: success = true or false

$i_{init} = 0$
for i in I
  Compute F_i as Eq.(3).
  for $s \in S$, where S is sorted by the distance to v_i in ascending order
    Compute $\delta_{v_i s}$ as $\delta_{v_i s} = F_i - \left(t_{v_i s} + t_{s v_{i+1}} - t_{v_i v_{i+1}}\right) - \mu_s$.
    Update the total required recharge amount and minimum required recharge amount as $\Delta_E^s = \Delta_E + \left(c_{v_i s} + c_{s v_{i+1}} - c_{v_i v_{i+1}}\right) \times \beta, \Delta_{E_{low}}^s = \Delta_E + (c_{v_i s} + c_{s v_{i+1}} - c_{v_i v_{i+1}}) \times \beta$.
    Compute the total required and minimum required charging time if recharge at s, $\delta_{v_i s}^{max} = \frac{\Delta_E^s}{\alpha_s}, \delta_{v_i s}^{min} = \frac{\Delta_{E_{low}}^s}{\alpha_s}$.
    GO TO line 2 if $\delta_{v_i s} < \delta_{v_i s}^{min}$.
    Define $\varepsilon_{route}^i = [A_i + t_{v_i s} + \mu_s, A_i + t_{v_i s} + \mu_s + \delta_{v_i s}]$.
    for $\varepsilon_{route}^i \in$ available time intervals of s, with time intervals sorted by their starting time
    Compute the duration δ of $\varepsilon_{route}^i \cap \varepsilon_s$ if $\varepsilon_{route}^i \cap \varepsilon_s \neq \emptyset$; otherwise GO TO line 2.
      if $\delta \geq \delta_{v_i s}^{max}$
        Insert recharge at i with charging time $\delta_{v_i s}^{max}$ and add this charging event to s.
        Set $\Delta_E = 0$ and **return true**.
      else if $\delta_{v_i s}^{min} \leq \delta < \delta_{v_i s}^{max}$
        Insert recharge at i with charging time δ and add this charging event to s.
        Update $\Delta_E = \Delta_E^s - \delta \times \alpha_s$ and update the E_j with position j after i.
        If $\Delta_E > 0$, update i_{low} and $\Delta_{E_{min}}$ to the next position with its SoC below E_{min}, and GO TO line 2; otherwise, **return true**.
      else
        GO TO line 10.
      end if
    end for
  end for
end for
return false and remove the inserted charging events above if $\Delta_E > 0$.

4.2.6 Acceptance criteria

The stopping criteria is based on the DA approach. s'' is accepted at the next iteration if $c(s'') \leq c(s_{best}) + T$, where the temperature T is updated at every iteration as $T = T - T_{max}/T_{red}$. The maximum temperature T_{max} depends on the average distance $\bar{c}$ of the arcs of the graph, i.e., $T_{max} = t_{max} \times \bar{c}$ (Braekers et al., 2014). T_{red} is the cooling temperature that decides the cooling speed of T. T is set to T_{max} again when $T \leq 0$ (line 16 in Algorithm 4).

4.3 Computational experiments

To evaluate the proposed algorithm, we generate a set of instances for the computation experiments. The hybrid LNS algorithm is implemented in *Julia*. All experiments are conducted on a laptop with an 11th Gen Intel(R) Core(TM) i5-1135G7 CPU and 64GB RAM. The test instances are first described along with the test scenario, followed by a sensitivity analysis on the algorithm parameters. The performance of destroy, repair and local search operators designed specifically for the EIDARP are analyzed. At the end, the proposed hybrid LNS are also compared with the 8-hour exact solution found by the commercial solver Gurobi V11.0.

4.3.1 Test instances

Test instances are generated based on a scenario with two crossed transit lines in a 16 by 16 km square area (see Figure 4.1). The two transit lines operate in a bidirectional manner, and each has three transit stations (A, B, C for line1 and D, E, F for line2) with one transfer station in the middle, allowing customers to transfer from one transit line to another. The EIDARP service planning horizon is 2 hours, with a total of four departures (two for each direction) operating at each line. The average operating speed of the transit is 50 km/h, while the bus speed is set as 25 km/h. The bus operator utilizes two types of buses at two depots, with sufficient large fleet size to serve customers. Each bus must start and end its service at its designated depot. Each depot is equipped with an operator-owned DC fast charger. We assume a service time (i.e. access and egress) for each charging operation. The parameter setting is presented in Table 2. Note that the penalty cost ω for customer rejection is set to a large value, so that the system will serve as many customers as possible. Five instances are generated for the algorithm parameter tuning, each consisting of 50 customers with origins and destinations randomly located in the service area. Each instance consists of an equal number of outbound and inbound customers. We use the Gurobi solver within an 8-hour computational time limit to validate the performance of the hybrid LNS algorithm. The performance of the algorithm is also evaluated based on a set of test instances with the number of customer ranging from 10 to 100 customers.

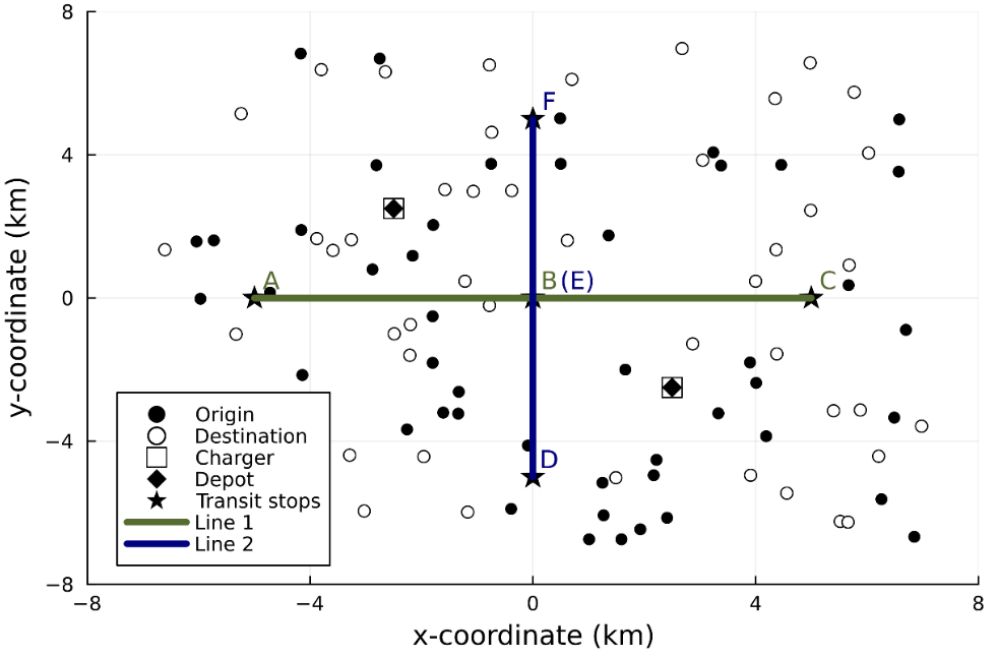


Figure 4.1: Example of the test scenario with two bi-directional transit lines and 50 customers

Table 4.2: The parameter settings of the numerical study

Parameter	Value
Planning horizon T	2 hours
Number of transit lines and stations	2 lines and 5 stations
Number of transit departures	4 per line (2 per direction)
Number of depots	2
Number of chargers	2
Number of bus types	2 types
Bus capacity Q^k	15 and 22 passengers/bus
Bus battery capacity	69 and 103.5 kWh
Bus energy consumption rate	0.552 and 0.828 kWh/km
Bus average speed	25km/h for both types
Maximum and minimum SoCs	80% (10%) of the battery capacity
Charging power	0.83 kWh/minute (DC fast charger)
Time window width at customer's origin and destination	15 minutes
Maximum waiting time at transit stations γ	10 minutes
Maximum walking distance	1.5 km
Walking speed	0.085 km/minute
Detour factor φ	1.5
The weights at objective function for bus travel cost $\lambda_1$and customer travel costs λ_2	1, 1
Penalty cost ω	200
Service time μ	0.5 minute
Service time for each charging operation	1 minute

4.3.2 Algorithm parameter tuning

In this section, we tune five parameters $n_{iter}, t_{max}, T_{red}, \xi_{max}$ and α in the proposed hybrid LNS to identify a good configuration. Based on preliminary experiments, these parameters are initially set to $(n_{iter}, t_{max}, T_{red}, \ \xi_{max}, \alpha) = (500, 1.1, 100, 0.2, 1.02)$. Each parameter is then varied within a range around its initial value to assess its impact on performance. The experiments are conducted sequentially as the parameter are independent of each other, except for t_{max} and T_{red}. Each time a new value for the analyzed parameter is identified, it is updated in subsequent analyses. Note that more advanced parameter tuning approaches can also be used (Lindauer et al., 2022; López-Ibáñez et al., 2016). Five instances with 50 customers are applied for different parameter values, and the CPU time and objective values are recorded after five runs. The BKS for each instance is recorded from all the experiments conducted. The average gap reported in the results is calculated as the average gap of the hybrid LNS solutions compared to the 8-hour BKS obtained by Gurobi. The results are shown in Table 4.3 – Table 4.5.

First, n_{iter} is analyzed with the results presented in Table 4.3. The solution quality increases with the number of iterations at the costs of higher CPU time. After 600 iterations, the gap reduces slightly, but CPU time increases more than 20%. Therefore, n_{iter} is set to 600 by considering the trade-offs of additional computational time and the reduction in the objective function.

Table 4.3: Results of sensitivity analysis on n_{iter}

n_{iter}	300	400	500	600	700	800	900	1000
Avg. Gap (%)	2.36	2.13	2.03	1.73	1.68	1.67	1.65	1.61
Avg. CPU time (s)	166	258	327	397	481	492	548	623

As t_{max} and T_{red} are used to jointly determine the DA acceptance criteria, the experiments are conducted by setting 100 combinations with 10 values each for both t_{max} and T_{red}. Table 4.4 presents the average gap of each value for these two parameters. The minimum average gap occurs at $t_{max} = 1.1$ and $T_{red} = 700$. As the CPU time does not exhibit a clear trend across the tested combinations, we set $t_{max} = 1.1$ and $T_{red} = 700$.

Table 4.4: Results of sensitivity analysis on t_{max} and T_{red}

t_{max}	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
Avg. gap (%)	2.05	1.86	2.16	1.63	1.76	1.81	1.93	1.82	1.89	2.06
T_{red}	100	200	300	400	500	600	700	800	900	1000
Avg. gap (%)	1.96	2.10	2.00	1.90	1.83	1.81	1.73	1.85	2.02	1.77

Table 4.5 presents the results for ξ_{max} , α and β. ξ_{max} is the degree of destroy. We can observe that when ξ_{max} increases, the CPU time increases but the average gap does not increase accordingly. The minimum average gap is 1.73% with $\xi_{max} = 0.25$. α controls the frequency of executing the local search procedure. The higher the α is, the longer the CPU time and the better the average gap. We set α to 1.06 at the minimum average gap.

Table 4.5: Results of sensitivity analysis on ξ_{max} and α

ξ_{max}	0.15	0.2	0.25	0.3	0.35
Avg. gap (%)	1.94	1.77	1.73	1.77	1.92
Avg. CPU time (s)	253	343	459	555	681
α	1.02	1.03	1.04	1.05	1.06
Avg. gap (%)	2.11	1.91	1.94	1.64	1.43
Avg. CPU time (s)	398	467	488	509	525

In summary, the best configuration for these parameters is $(n_{iter}, t_{max}, T_{red}, \xi_{max}, \alpha) = (600, 1.0, 700, 0.25, 1.06)$, which is utilized in the following experiments.

4.3.3 Performance of the algorithm operators

This section evaluates the impacts of the six operators that are exclusively developed for EIDARP as described in Section 4.2.2 and 4.2.3, which are $r_{TS}, r_{Tpriority}, d_{related}$ and three local search operators: *exchange TAs of customers*, *replace TA by bus*, and *re-assign first-/last-mile bus service.* To assess their contribution to the solution quality, each is removed from the hybrid LNS and compared with the case in which all operators are included. The experiment is conducted on the five 50-customer test instances, each solved with five runs.

Table 4.6 presents the average gap to BKS and CPU time for different cases. The average gap excluding one of these operators leads to 0.41% — 0.84% higher gap than when all operators are included. In the case without $r_{Tpriority}$, the average gap increases up to 2.09% with a slight CPU time reduction. This

indicates that $r_{Tpriority}$ is an effective and efficient operator. By contrast, removing the local search operator *exchange TAs of customers* results in the smallest improvement to the average gap, but not the greatest reduction to the CPU time. This suggests that this operator has a relatively lower impact compared to other five operators. Despite these variations, the six operators demonstrate their influence on the hybrid LNS algorithm for EIDARP. The exclusion of any operator negatively impacts on solution quality.

Table 4.6: Performance of the EIDARP-excusive operators

Experiments	Avg. Gap	Avg. CPU time (s)
All operators	**1.43%**	**525**
Without r_{TS}	2.14%	346
Without $r_{Tpriority}$	2.09%	501
Without $d_{related}$	1.95%	418
Without local search operator *exchange TAs of customers*	1.84%	459
Without local search operator *replace TA by bus*	1.85%	522
Without local search operator *re-assign first-/last-mile bus service*	2.27%	444

4.3.4 Algorithm performance

The solution quality of the proposed hybrid LNS is compared with the strengthened MILP formulation developed in Chapter 3 solved by Gurobi. Test instances are generated using the scenario described in Section 4.3.1 with the number of customers between 10 and 50. The results from Gurobi are obtained after 8-hour CPU time, and the results of the hybrid LNS are based on five runs. The test instances are available in github.com/YMF2022/EIDARP-instances.

Table 4.7: Performance of the hybrid LNS and its comparison with MILP (full initial SoCs of buses)

n	MILP		Hybrid LNS			
	Obj. value[1]	Gap to LB[2]	Best obj.	Best gap[3]	Avg. gap[3]	Avg. CPU (s)
10	490.82	0.0%[4]	490.82	0.0%	0.0%	6
15	688.48	9.5%	683.52	-0.7%	0.0%	12
20	815.54	2.4%	811.69	-0.5%	0.5%	11
25	1203.21	22.3%	1072.10	-10.9%	-10.8%	33
30	1419.42	24.7%	1197.97	-15.6%	-14.3%	76
35	2640.80	51.8%	1522.13	-42.4%	-42.1%	137
40	3351.43	63.5%	1759.03	-47.5%	-47.0%	162
45	3457.61	57.9%	1792.33	-48.2%	-47.8%	365
50	3727.94	56.9%	1917.23	-48.6%	-48.3%	417
Average	1977.25	32.1%	1249.65	-23.8%	-23.3%	136

Notes: 1. Objective function value of the incumbent solution after an 8-hour CPU time. 2. Gap to the lower bound. 3. Gap compared wtih the objective function value found by Gurobi. 4. Optimal solution with the CPU time of 1239 seconds. Note that no customers are rejected.

Table 4.7 summarizes the results when buses are fully charged at the beginning of the service. Among the instances, Gurobi obtained the optimal solution only for the 10-customer case with 1239 seconds, and the average gap to lower bound is 32.1% (MILP column). The hybrid LNS obtained the same solution for 10-custromer test instance with an average of 6-second CPU time. For the remaining instances, the best objective function value obtained by the hybrid LNS across five runs consistently outperform (i.e. negative gaps) the MILP solutions obtained by Gurobi, with an average improvement of -23.8% (best) and -23.3% (average). The 8-hour Gurobi solution has an average gap of 32.1% to the lower bound, while the hybrid LNS's average gap to the 8-hour Gurobi solution is -23.3%. This suggests the good quality of hybrid LNS solution. Moreover, the hybrid LNS obtains the solutions significantly faster with an average of 136 seconds only.

In the low initial SoC (30% of the battery capacity) scenario, the gaps to the lower bound of MILP are significantly higher than the fully initial SoC scenario. This experiment limits the maximum number of customers to 20 as Gurobi was not able to obtain feasible solutions or the obtained gaps to lower bound is relatively high for larger problem size (Gap to lower bound is 78.0% for 30-customer scenario). The results are provided in Table 4.8. The instance with 10 customers is the only one for which MILP finds the optimal solution. However, due to recharging scheduling in the low SoC case, the CPU time is 7.3 hours, which is 20 times longer than the fully charged scenario presented in Table 4.7. By contrast, the hybrid LNS shows a 1-second difference for the same instance. MILP's average gap to the lower bound is 12.5% for all instances, while the hybrid LNS's average gap to MILP's solution are -5.5% (best) and -4.8% (average). When buses have low initial SoCs, the variance of the hybrid LNS performance is slightly higher than the fully charged case. In terms of charging duration, the hybrid LNS achieved an average charging time of 7.9 minutes, compared with MILP's average of 17.4 minutes. This difference aligns with the hybrid LNS strategy of recharging only the minimum required amount, as described in Section 0.

Table 4.8: Performance of the hybrid LNS and its comparison with MILP (low initial SoCs of buses)

	MILP			Hybrid LNS				
n	Obj. value[1]	Gap to LB[2]	Charging time (min)	Best obj.	Best gap[3]	Avg. gap[3]	Avg. charging time (min)	Avg. CPU (s)
10	490.82	0.0%[4]	5.0	490.82	0.0%	1.6%	5.0	7
11	502.37	4.1%	7.8	502.37	0.0%	0.2%	6.1	11
12	584.67	15.1%	11.0	573.31	-1.9%	-0.5%	13.7	8
13	486.89	2.8%	16.3	486.89	0.0%	1.6%	1.7	6
14	653.80	11.7%	0.1	610.56	-6.6%	-6.5%	0.0	15
15	790.14	27.2%	4.7	683.22	-13.5%	-12.9%	2.9	16
16	685.22	14.8%	8.2	641.75	-6.3%	-5.6%	7.8	12
17	889.40	20.2%	36.1	803.48	-9.7%	-9.3%	10.4	25
18	686.44	8.7%	4.1	675.83	-1.5%	-1.5%	4.3	15
19	998.44	23.6%	1.8	851.03	-14.8%	-14.7%	6.0	19
20	862.16	10.4%	96.0	811.84	-5.8%	-4.8%	29.3	13
Average	693.67	12.5%	17.4	653.04	-5.5%	-4.8%	7.9	13

Notes: 1. Objective function value of the incumbent solution after an 8-hour CPU time. 2. Gap to the lower bound. 3. Gap compared with the objective function value found by Gurobi. 4. Optimal solution with the CPU time of 26118 seconds. Note that no customers are rejected.

Table 4.9 presents the results of the hybrid LNS for the test instances up to 100 customers with full and low initial SoCs of buses. In both scenarios, the CPU time increases significantly with the number of customers. No recharging events are observed in the full SoC scenario, whereas the low SoC scenario shows an overall increase in charging times as the number of customers grows. Although the low SoC scenario consistently exhibits longer charging times than the full SoC scenario, the difference in their CPU times is minor, suggesting that the recharging scheduling approach is efficient and does not significantly impact the CPU time for these experiments.

Table 4.9: Performance of the hybrid LNS on full and low initial SoC

n	Full (100%) initial SoC			Low (30%) initial SoC			
	Best obj.	Avg. obj.	Avg. CPU (s)	Best obj.	Avg. obj	Avg. charging time (min)	Avg. CPU (s)
10	490.82	490.82	10	490.82	498.90	5.0	7
20	811.69	819.58	11	811.84	820.62	7.9	13
30	1197.97	1216.87	76	1211.10	1219.12	10.6	134
40	1759.03	1777.56	162	1787.43	1801.12	13.2	251
50	1917.23	1928.69	417	1953.05	1976.15	22.7	513
60	2467.26	2492.48	1031	2499.64	2525.25	26.0	1125
70	2630.24	2638.84	1078	2697.17	2715.59	14.8	1361
80	3060.91	3122.02	1821	3137.84	3158.55	20.7	2891
90	3256.01	3267.10	3722	3312.57	3350.95	21.9	4299
100	3864.78	3908.28	6163	4007.95	4029.80	27.4	7054
Average	2145.59	2166.22	1449	2575.00	2595.03	17.0	1765

Note: No recharging for all instances in the full SoC scenario.

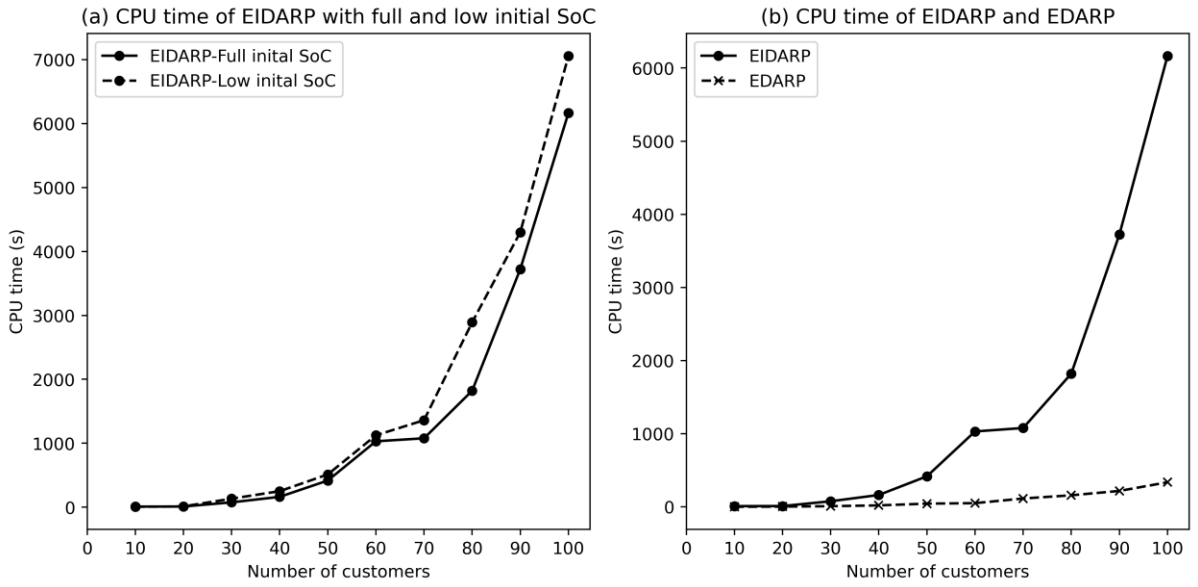


Figure 4.2: Computational time of EIDARP at different initial SoC levels and comparison with EDARP

Figure 4.2(a) illustrates the average CPU time for the full and low SoC scenarios, confirming that CPU time increases exponentially with the increase of the number of customers. Charging scheduling part for the low SoC scenario does not have a significantly impact on its CPU time. Figure 4.2 (b) also compares the CPU time between EDARP and EIDARP under full SoC scenario, where no recharging is observed for either case. A significantly higher CPU time is observed for solving the EIDARP compared with that of EDARP. This increase in CPU time of EIDARP is attributed to the additional complexity when considering different possible travel options involving walk, bus and transit (travel option 1-4). While EDARP handles only one travel option (direct bus trips from customer origins to destinations).

4.4 Discussion and conclusions

This chapter developed a hybrid LNS metaheuristic to solve the EIDARP introduced in Chapter 3 with consideration of transit synchronization and utilization of EVs. Several numerical studies demonstrated that the difficulties of solving the EIDARP primarily stem from the utilization of different vehicles within a customer journey (different travel options). The proposed algorithm employs the departure-expanded network graph to tighten time-windows associated at the nodes of the graph. By identifying corresponding feasible transit stations (departure and arrival stations) for customers (i.e. TAs), the proposed hybrid LNS solve the EIDARP efficiently with partial recharge policy and capacitated charging stations.

A nine-step feasibility evaluation scheme for the EIDARP is proposed to efficiently check the feasibility of both bus routes and customers' journeys in terms of customers' maximum travel time, bus arrival time coordination at transit stations, and maximum waiting time at transit stations. The hybrid LNS incorporates a set of local search operators and a DA acceptance criteria to balance the diversification and intensification in better solution search. Several problem-specific LNS operators are also developed for the integrated problems.

To demonstrate the performance of the hybrid LNS, the metaheuristic is benchmarked against exact solutions found by the Gurobi solver. The hybrid LNS demonstrates near-optimal solutions for up to 50 customers with two transit lines when buses are fully charged, and up to 20 when the initial SoCs of buses is 30%. Computational experiments with up to 100 customers were conducted with full and low initial SoCs of buses. The results indicate that the proposed recharging scheduling method does not have significant impact on computational time, suggesting an efficient charging scheduling procedure. When compared with the non-integrated problem (EDARP), the computational complexity of EIDARP increases significantly faster due to the integration with transit, which introduces four additional travel options, thereby adding to the computational challenges.

Chapter 5 Performance analysis of the electric integrated demand-responsive system

This chapter analyzes the performance of the developed EIDRT service. When comparing integrated and non-integrated DRT, most existing studies primarily focus on the reduction in VKT by the on-demand mobility service, often overlooking the significant role played by the fixed-route service within the system. The performance of IDRT depends not only on the operational policy of on-demand bus operators, but also on the supply of the implied transit network. Analyzing the performance of the IDRP with respect to different system parameters is essential to improve its effectiveness and increase the ridership of the system.

Molenbruch et al. (2021) investigated the service performance by examining various system configurations with respect to on-demand bus service operations (e.g., different maximum detour factors), transit service operations (e.g., transit speed and frequencies), and demand distributions. Their findings suggest that the intensity of transit service frequency is less critical due to complementary services provided by more flexible on-demand bus operators. Similarly, Melis et al. (2024) examined the IDRT with respect to transit network configuration, fleet size and the speeds of transit and on-demand vehicles. The performance metrics are mainly customers' service rate and travel time. Inspired by these two studies, Section 5.1 investigates the service by exploring various system parameter configurations and analyzing their impact on the performance of the EIDRT service. The experiments are designed to reflect real-world applications. Our analysis focuses on evaluating the performance of EIDRT from both the customer's and operator's perspectives.

Moreover, it is crucial to assess whether EIDRT can effectively compete with existing transportation options, including private cars, PT and current DRT services. This evaluation provides insights into EIDRT's viability and attractiveness as a transportation alternative. For this purpose, Section 5.2 presents a case study comparing the EIDRT service with these existing options. The analysis leverages realistic ride data obtained from a DRT service operator in Luxembourg, ensuring the comparison reflects practical, real-world conditions. By examining factors such as customer service level, and operational costs, this case study aims to highlight EIDRT's potential advantages and identify areas for its improvement. Section 5.3 discusses the key findings of the experiments.

5.1 Performance analysis under different EIDRT system configurations

To analyze the performance of the EIDRT, we identify eight key parameters from the modelled problem, EIDARP, for the analysis as follows.

(1) Detour factor φ
(2) Weights on customer travel time in the objective function (λ_2)
(3) Maximum waiting time (γ)
(4) Bus fleet size
(5) Bus operational speed
(6) Initial SoCs of buses
(7) Transit frequency
(8) Transit network layout

The analysis is based on a baseline scenario of 100 customers with the problem configuration described in Section 4.3.1. The reference values of the EIDRT system parameters are set as $\varphi = 1.5$, $\lambda_2 = 1.0$, unlimited bus fleet size, 25km/h for the average bus speed, full (100%) initial SoC of buses, transit frequency is every 30 minutes, and two crossed transit lines (Figure 4.1). The hybrid LNS developed in Chapter 4 is applied as the solution approach. In each of the following sections, one of the seven parameters is varied while the others remain unchanged to observe its impact on the results. The following Key Performance Indicators (KPIs) are defined for the evaluation.

- BTT: Total bus travel time in minutes.
- RT: Total recharging time in minutes.
- CTT: Average customer travel time: including in-transit travel time, in-bus travel time plus walking time in minutes.
- CTT-transit: Average customer in-transit travel time in minutes.
- CTT-bus: Average customer in-bus travel time in minutes.
- CTT-walk: Average customer walking time in minutes.
- WT: Average customer experienced waiting time in minutes, including waiting times at transit stations for transfers at both first mile and last mile along with waiting times during bus services.
- # cus-transit: The number of customers using the transit service.
- # used buses: The number of used buses.
- # cus/bus: Average number of served customers per bus.
- # reject: The number of rejected customers.

Note that we use the average value for time-related KPIs of customer travel experience because the total value cannot straightforwardly describe their travel experience in the case of customer rejection.

5.1.1 Detour factor φ

This section analyzes the effects of detour factor, φ, within the range of 1.3 to 2.5. The results is summarized in Table 5.1. The result shows that an increase in φ leads to higher CTT and reduced BTT. The increase in CTT can be attributed to higher utilization of the transit service and longer waiting time, as indicated by the rise in CTT-transit and WT. However, the values of CTT-bus increases slightly with larger φ due to a higher degree of ride-sharing, evidenced by a decrease in the number of used buses and an increase in the average number of customers served per bus. WT also increases accordingly, reaching up to 6.4 minutes when $\varphi = 2.5$, while no consistent trend is observed for CTT-walk.

Table 5.1: Impacts of φ on the performance of EIDARP

φ	BTT	# used buses	# cus/bus	# cus-transit	CTT	CTT-transit	CTT-bus	CTT-walk	WT
1.3	2232.0	36	3.5	31	20.5	1.9	18.6	0.00	1.9
1.5[1]	1902.9	29	4.6	39	21.4	2.8	18.4	0.15	2.7
1.7	1864.0	28	5.1	52	22.3	4.0	18.1	0.24	3.9
1.9	1829.0	29	5.0	53	23.0	3.7	19.2	0.14	4.5
2.1	1770.5	26	5.7	53	23.6	3.9	19.6	0.17	4.8
2.3	1783.4	26	5.4	47	24.2	3.9	20.3	0.07	5.0
2.5	1775.1	25	6.0	59	24.3	4.6	19.5	0.25	6.4

Note: 1. The baseline scenario. No customers rejected and no recharging operations observed for all the cases.

Figure 5.1 depicts customers' experienced detour. Figure 5.1 (a) shows a more dispersed distribution of customers' experienced detours when larger φ is used. For each value of φ, some customers experience the upper bound of maximum travel time, whereas the minimum customers' experienced detours are keeping at the similar level despite different values of φ. When customers' experienced detours are smaller than 1.0, they benefit from using transit service (operating with a higher speed (50 km/h) compared with the bus (25 km/h)), as shown in Figure 5.1 (b). Figure 5.1 (b) classifies customers into two groups: customers (partially or exclusively) served by transit (travel options 1-4) and customers served exclusive by buses (travel option 5). As presented in Figure 5.1 (b), customers served by transit could eventually experience a detour less than 1.0, while customers exclusively served by bus always experience detour equal or greater than 1.0.

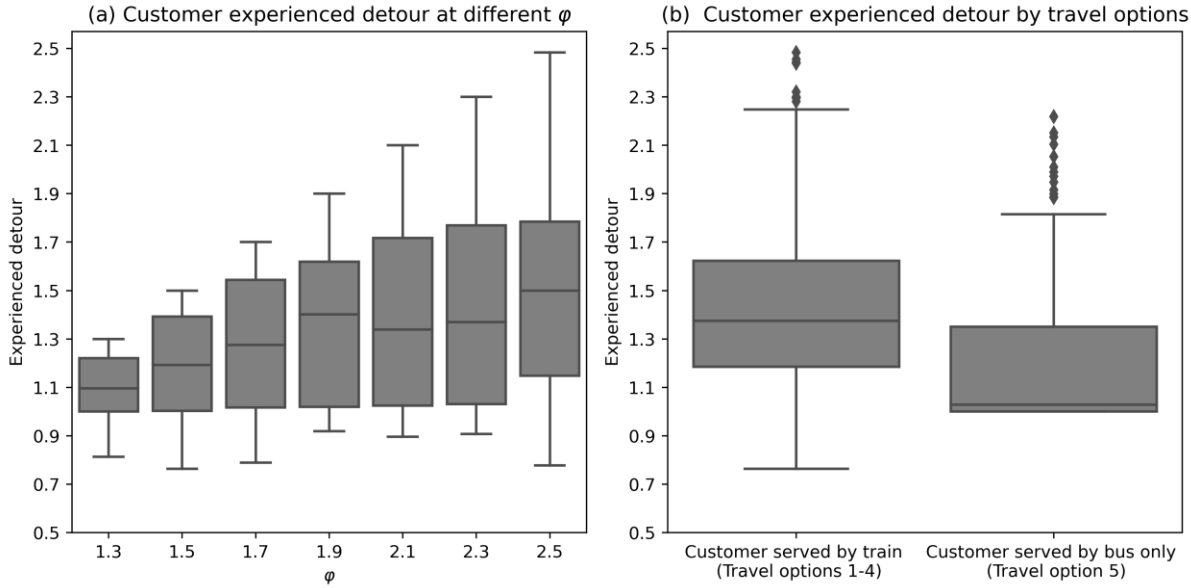


Figure 5.1: Impacts of φ on customers' experienced detour

5.1.2 Weight on customer travel time in the objective function λ_2

Recall that $\lambda_1, \lambda_2,$ and λ_3 are the weights for bus travel time, customer travel time and penalty cost of rejecting customers used in the objective function, respectively. The experiments vary λ_2 from 0.0 to 2.5 while keeping $\lambda_1 = 1.0$ and $\lambda_3 = 1.0$. The results are presented in Table 5.2. As λ_2 increases, CTT decreases while BTT increases. The increase in BTT is due to more customers utilizing bus service but less transit service, as reflected in the changes in CTT-transit and CTT-bus. Moreover, when more customers are served by door-to-door on-demand buses, a decrease in CTT-walk is observed. Unlike other KPIs, WT does not exhibit a consistent trend.

Table 5.2: Impacts of λ_2 on the performance of EIDARP

λ_2	BTT	# used buses	# cus/bus	# cus-transit	CTT	CTT-transit	CTT-bus	CTT-walk	WT
0.0	1884.8	31	4.2	34	23.1	2.8	20.0	0.25	1.5
0.5	1893.9	30	4.5	38	21.5	2.9	18.6	0.00	3.0
1.0[1]	1902.9	29	4.6	39	21.4	2.8	18.4	0.15	2.7
1.5	1934.0	29	4.4	32	20.5	2.3	18.1	0.08	2.8
2.0	1986.1	31	4.4	37	19.9	2.3	17.6	0.00	2.9
2.5	2055.8	33	3.9	35	19.7	2.2	17.5	0.00	1.8

Note: 1. The baseline scenario. No customers rejected and no recharging operations observed for all the cases.

As the customer travel time in EIDARP's objective function excludes waiting time, while the maximum travel time constraint accounts for it, both experienced detour and experienced detour without waiting time are defined.

- Experienced detour for $r \in R$: Customer r's travel time plus waiting time divided by their respective direct travel time by bus.
- Experienced detour-without WT for $r \in R$: Customer r's travel time divided by their respective direct travel time by bus.

Figure 5.2 investigates the impacts of λ_2 on customers' experienced detour with (Figure 5.2 (b)) and without waiting times (Figure 5.2 (a)). As the detour factor φ is set to 1.5, both KPIs are less or equal than this value. In Figure 5.2 (a), an increase in λ_2 corresponds to a reduction in experienced detour without WT, with the distribution becoming more concentrated around 1.0. This trend reflects a higher proportion of customers served by direct bus services without shared rides, which is also observed in the decrease of #cus/bus in Table 5.2. Similarly, customers' experienced detour decreases with the increase in λ_2, though not as significantly as observed in Figure 5.2 (a). This is because the objective function minimizes customer travel times (without WT), leading to a faster reduction in customers' experienced detour without WT. Besides, customers' experienced detour still reaches the upper bound of 1.5. This is attributed to the waiting time not decreasing proportionally, as indicated in Table 5.2.

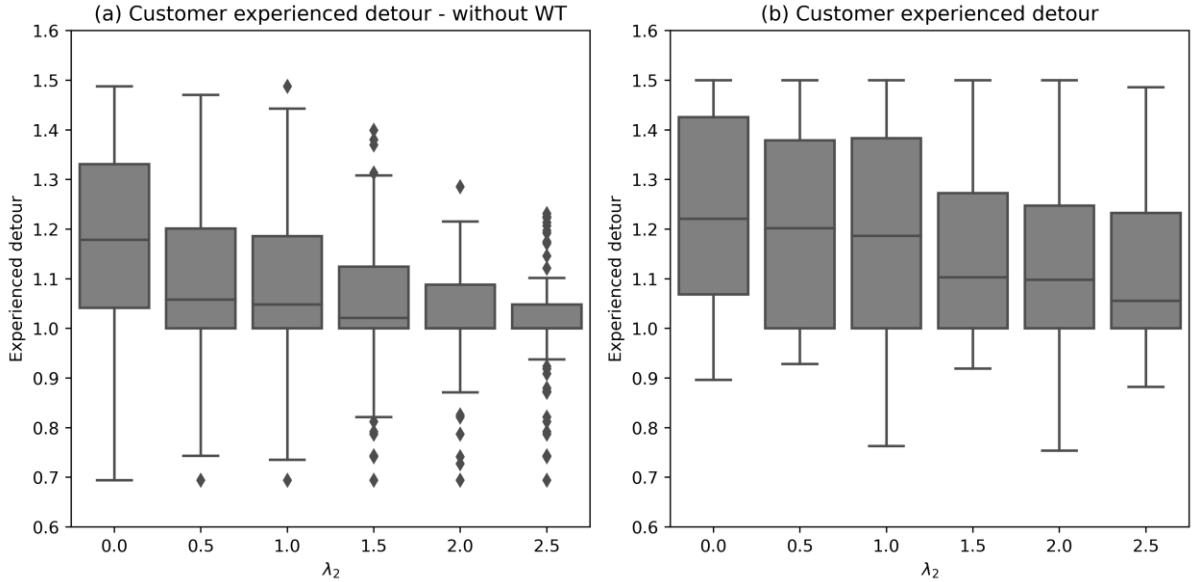


Figure 5.2: Impacts of λ_2 on customers' experienced detour with and without waiting time

5.1.3 Maximum waiting time γ

This section investigates the maximum waiting time constraint at transit stations when customers transfer between bus and transit services. γ is varied from 5 to 15 minutes to evaluate customers' experienced waiting times. Customers' experienced waiting time includes waiting time at transit station for both first- and last- mile connections and the waiting time during the bus service. In Figure 5.3 (a), customer waiting time at each transit station is constrained when $\gamma = 5$ or $\gamma = 6$, but when $\gamma > 7$, customer experienced waiting time ranges between 0 and 14.5 minutes, disregarding the value of γ. This is because customers' experienced waiting time is partially bounded by the maximum travel time constraint. The variability in Figure 5.3 (a) is primarily attributed to customers served by transit, as depicted in Figure 5.3 (b). The group of customers served by transit experience more diverse and longer waiting times, whereas the bus-only group experience shorter and more consistent waiting times, typically less than 2 minutes.

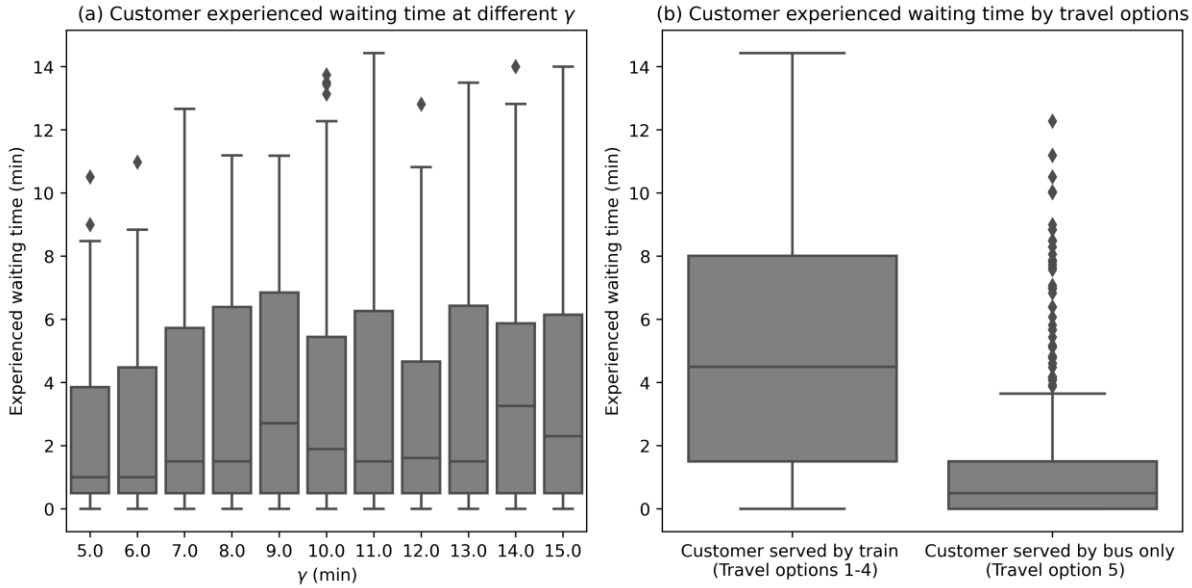


Figure 5.3: Impacts of γ on customer experienced waiting time

5.1.4 Bus fleet size

Table 5.3 shows the impacts of fleet sizes. When the fleet size is 10 (the minimum value), the number of rejected customers reaches 52 and no customers use the transit service. As the fleet size increases, customer rejections decreases, along with more BTT required to serve additional customers. In the unlimited fleet size case, all customers can be served with 29 buses, but CTT increases. Both CTT-transit and CTT-bus show little change with increasing fleet size. However, WT increases from 0.4 minutes (fleet size is 10 buses) to 2.7 minutes (fleet size is 29 buses). This increase is likely due to longer waiting time at transit stations and the added waiting time for customers sharing rides. The average number of served customers per bus does not exhibit a consistent trend.

Table 5.3: Impacts of bus fleet size on the performance of EIDARP

Fleet size	BTT	# used buses	# cus/bus	# reject	# cus-transit	CTT	CTT-transit	CTT-bus	CTT-walk	WT
10	780.8	10	4.8	52	0	15.7	0.0	15.7	0.00	0.4
15	1223.2	15	5.6	31	17	18.1	1.8	16.2	0.10	1.8
20	1504.7	20	5.5	16	30	20.4	2.5	17.8	0.08	2.3
25	1868.2	25	4.8	3	26	21.6	1.9	19.7	0.00	2.1
Unlimited[1]	1902.9	29	4.6	0	39	21.4	2.0	18.4	0.15	2.7

Note: 1. The baseline scenario. No recharging operations observed for all cases.

5.1.5 Bus operational speed

As different service areas (rural/urban) limit bus traveling speed, this section examines the impact of varying the average bus speed from 25 to 50 km/h. The results, presented in Table 5.4, show that increasing bus speed reduces both BTT and CTT. The number of served customers per bus increases, resulting in the reduction in the total number of buses required. When bus speed doubles from 25 km/h to 50km/h, the BTT is reduced by more than half due to more shared rides. When the bus speed reaches 50 km/h (same as transit operational speed), only six customers utilize the transit service. This observation highlights that increasing the bus speed undermines the purpose of transit integration.

Table 5.4: The impacts of bus speed on the performance of the EIDARP

Speed (km/h)	BTT	# used buses	# cus/bus	# cus-transit	CTT	CTT-transit	CTT-bus	CTT-walk	WT
25[1]	2047.7	32	4.3	42	21.3	3.1	18.1	0.11	3.0
30	1535.4	26	4.6	23	17.6	1.7	15.9	0.00	2.2
35	1256.3	22	5.4	18	15.8	1.1	14.7	0.00	1.3
40	1118.1	20	5.9	18	13.7	1.0	12.7	0.00	1.1
45	986.5	19	5.7	10	12.2	0.6	11.6	0.01	1.0
50	868.2	18	5.9	6	11.0	0.4	10.7	0.00	0.7

Note: 1. The baseline scenario. No customers rejected and no recharging operations observed for all the cases.

5.1.6 Initial SoCs of buses

The previous results do not examine bus recharging operations when buses are fully recharged at the beginning of the service. However, in reality, there may be some uncertainties, meaning that 100% initial SoC cannot be ensured for all buses. This section analyzes the impacts of recharging on this service by setting an identical initial SoC level for all buses, ranging between 20% and 100% for all buses. The results are presented in Table 5.5. As this experiment sets unlimited bus fleet sizes, all customers are served. However, this requires 63 buses to serve all customers when the initial SoC level is at 20%. Besides, the case of 20% initial SoC shows the lowest CTT-bus but the highest values in CTT-transit and number of customers using transit, suggesting that buses serve more first-mile and last-mile services instead of providing door-to-door bus connections between customers' origins and destinations. When initial buses' SoCs increase, BTT decreases, resulting from both higher number of served customers per bus and less distance to CSs for recharge. We observe that CTT-bus also increases slightly while CTT-transit reduces slightly, suggesting that buses become more efficient in serving

customers as a door-to-door bus service when recharging is not necessary. Increased CTT-transit is accompanied by a rise in CTT-walk, while no significant changes occur in WT across different initial SoC settings.

Table 5.5: The impacts of initial SoC level on the performance of the EIDARP

Initial SoC	BTT	# used buses	# cus/bus	RT	# cus-transit	CTT	CTT-transit	CTT-bus	CTT-walk	WT
20%	2522.7	63	2.2	130.3	46	19.9	3.9	16.0	0.00	2.7
25%	2211.3	44	3.1	61.4	40	20.3	3.3	17.0	0.08	2.6
30%	1989.6	33	4.0	35.9	36	20.6	2.5	18.1	0.08	2.9
35%	1964.3	31	4.2	11.8	34	20.8	2.7	18.2	0.01	2.5
40%	1971.4	30	4.5	1.3	38	21.2	2.8	18.3	0.01	2.8
100%[1]	1902.9	29	4.6	0.0	39	21.4	2.8	18.4	0.15	2.7

Note: 1. The baseline scenario. No customers rejected for all cases.

5.1.7 Transit frequency

This section analyzes the impact of transit frequency, varying it from every 10 minutes to every 60 minutes for both transit lines. The results are presented in Table 5.6. When transit service is less frequent, both BTT and CTT increase. This is primarily due to more customers receiving direct door-to-door bus service, as reflected in fewer customers using transit, lower CTT for transit users, and a decrease in the number of customers served per bus. With longer transit intervals, fewer customers opt for transit, leading to reduced overall waiting times and shorter walking distances to transit stations. However, the number of buses used does not follow a clear pattern.

Table 5.6: The impacts of transit frequency

Frequency (min)	BTT	# used buses	# cus/bus	# cus-transit	CTT	CTT-transit	CTT-bus	CTT-walk	WT
10	1840.3	30	5.0	53	20.5	4.2	16.2	0.08	3.8
20	1764.6	30	4.7	44	20.4	3.6	16.8	0.07	3.2
30[1]	1902.9	29	4.6	39	21.4	2.8	18.4	0.15	2.7
40	1902.9	26	4.5	18	21.7	1.2	20.5	0.00	1.6
50	1904.6	26	4.4	15	22.3	1.2	21.2	0.00	1.1
60	1971.1	28	4.0	11	21.9	0.8	21.0	0.00	1.2

Note: 1. The baseline scenario. No customers rejected for all cases.

5.1.8 Transit network layout

A denser transit network provides higher accessibility for customers, but also makes the EIDARP problem more complex. We evaluate the service performance with different numbers of transit lines (representing network density) within a 16 by 16 km operational area. Figure 5.4 illustrates the layouts of four different transit network with the number of lines ranging from 1 to 4. For transit networks with more than two lines, the number of transit stations is three for each transit line with one transfer stations located in the middle. Each transit line operates in both directions with a frequency of 30 minutes during a two-hour operational period.

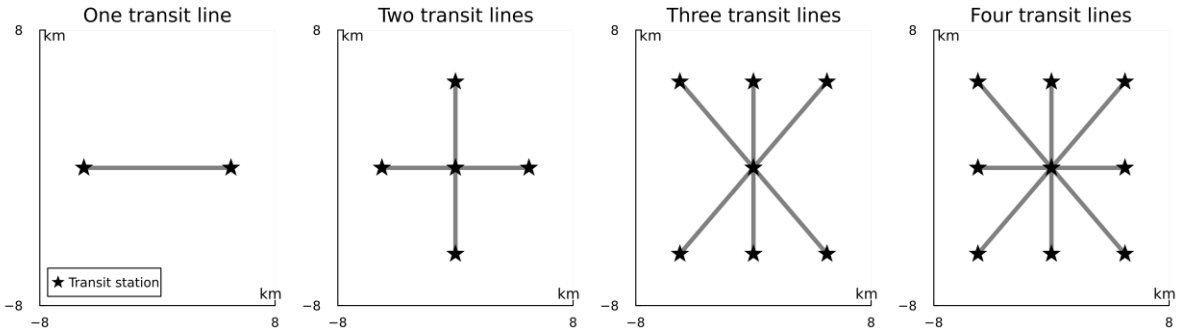


Figure 5.4: The layouts of four transit networks with an increasing numbers of transit lines

Table 5.7 presents the results with the four transit networks and the no-transit scenario. The latter represents a DARP service where all customers are served exclusively by buses, resulting in the highest BTT. As the number of transit lines increases, both BTT and CTT decreases. This is because more customers use transit services, with the number increasing from 0 to 58. This also leads to the increase in CTT-transit and CTT-walk. While more customers utilize the transit service in the cases with more transit lines, the number of buses used does not necessarily decrease, as customers are served by different buses at first-mile and/or last-mile services. Moreover, the average number of served customers per bus increases from 3.6 to 5.2, suggesting more customers are able to share bus rides at their first and last miles. The complexity of the EIDARP problem also depends on the transit network density, which is reflected in the higher CPU time as the number of transit lines becomes higher. For the no-transit scenario, the CPU time to solve the EIDARP is significantly smaller compared with other cases.

Table 5.7: The impacts of the density of transit networks on the performance of the EIDARP

# transit line	BTT	# used buses	# cus/bus	# cus-transit	CTT	CTT-transit	CTT-bus	CTT-walk	WT	CPU (s)
None	1971.1	28	3.6	0	21.8	0.0	21.8	0.00	0.9	148
One	1930.9	27	3.9	5	22.0	0.5	21.4	0.07	1.0	625
Two[1]	1902.9	29	4.6	39	21.4	2.8	18.4	0.15	2.7	2781
Three	1819.8	31	4.7	56	19.2	4.0	14.9	0.33	3.2	10882
Four	1650.4	28	5.2	58	18.9	5.1	13.1	0.72	1.8	40663

Note: 1. The baseline scenario. No customers rejected and no recharging operations observed for all the cases.

5.2 Case study

In this section, the EIDARP model is applied for a case study using empirical ride data from Kussbus, an DRT service offering a mobility service for the cross-border workers in Luxembourg in 2018 (Ma et al., 2021). Kussbus operated a meeting-point based DRT service, i.e. customers are picked up or dropped off at nearby meeting points instead of their origins and destinations. The objective is to assess the performance of EIDRT service by comparing it with Kussbus's DRT service (a meeting-point based DRT), EDRT (door-to-door DRT) and other transport options: car and PT.

We consider the case study of commuting trips during morning peak hours from Arlon (Belgium) to Luxembourg from 5:00 to 9:00 am. The Kussbus demand data covers 70 working days in 2018 with 1143 rides (requests) in total. As depicted in Figure 5.5, customers' origins are primarily concentrated in Arlon with only few pickup locations in-between Luxembourg and Arlon, while all destinations are

located around Luxembourg City. Table 5.8 provides a summary statistics of Kussbus data. With 1,143 trips recorded for 99 customers, each customer utilizes the service more than ten separate days on average.

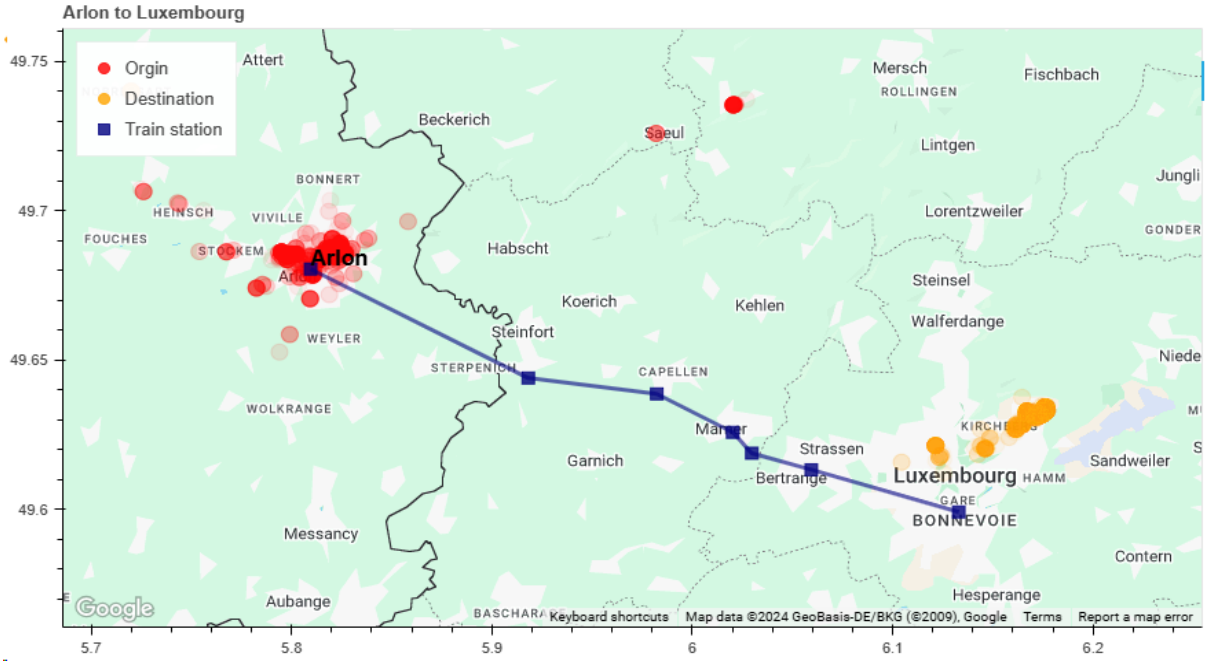


Figure 5.5: Arlon – Luxembourg case study

Table 5.8 Statistic summary of Kussbus data

# days	# total trips	# customers	Average distance of trips	Analyzed period
70 working days	1143	99	26.5 km	5 – 9 A.M.

As illustrated in Figure 5.5, study area covers one train line operating with a frequency of 3 to 4 departures per hour during the analyzed period, including one express train with limited stops. The proposed EIDRT integrates this train service with its actual departure schedule to meet Kussbus's customers' demands. The performance for EIDRT and EDRT services will be obtained by using the proposed hybrid LNS algorithm, while the performances of car and PT are obtained across five workings within a week in January 2025 by using Google Map API. PT represents the travel connections from customers' origins to destinations, as determined by Google Map, utilizing the existing public transport network, which includes trains, buses and trams. The following summarizes the data collection for each travel option:

- Car: Inquire customer travel times from Google Map API (average of five days)
- PT: Inquire customer travel times from Google Map API (average of five days)
- Kussbus: Empirical Kussbus's data
- EDRT: Solutions of the EDARP using the hybrid LNS algorithm
- EIDRT: Solutions of the EIDARP using the hybrid LNS algorithm

As determining an appropriate bus speed is essential for obtaining accurate results for E(I)DARP (as shown in Section 5.1.5), Section 5.2.1 explains the methods used to set realistic bus speed and other problem settings. The outcomes of this case study are presented in Sections 5.2.2 and 5.2.3, providing

insights from both the customer and bus operator perspectives. Section 5.2.4 summarizes the performance comparisons among door-to-door DRT, meeting-point-based DRT (Kussbus), and IDRT service.

5.2.1 Bus speed estimation and the parameter setting for E(I)DARP

The study area covers over 40×20 km, and bus travel speeds vary significantly depending on road types (local roads or highways). To address this issue, we estimated the average bus travel speed using Kussbus data (data fields include bus arrival times and their geographical coordinates of each intermediate stops). We divided the service area into three regions, as illustrated in Figure 5.6. Region 1 encompasses the area around Arlon station, Region 3 covers the vicinity of Luxembourg station, and Region 2 represents the intermediate locations between these two stations.

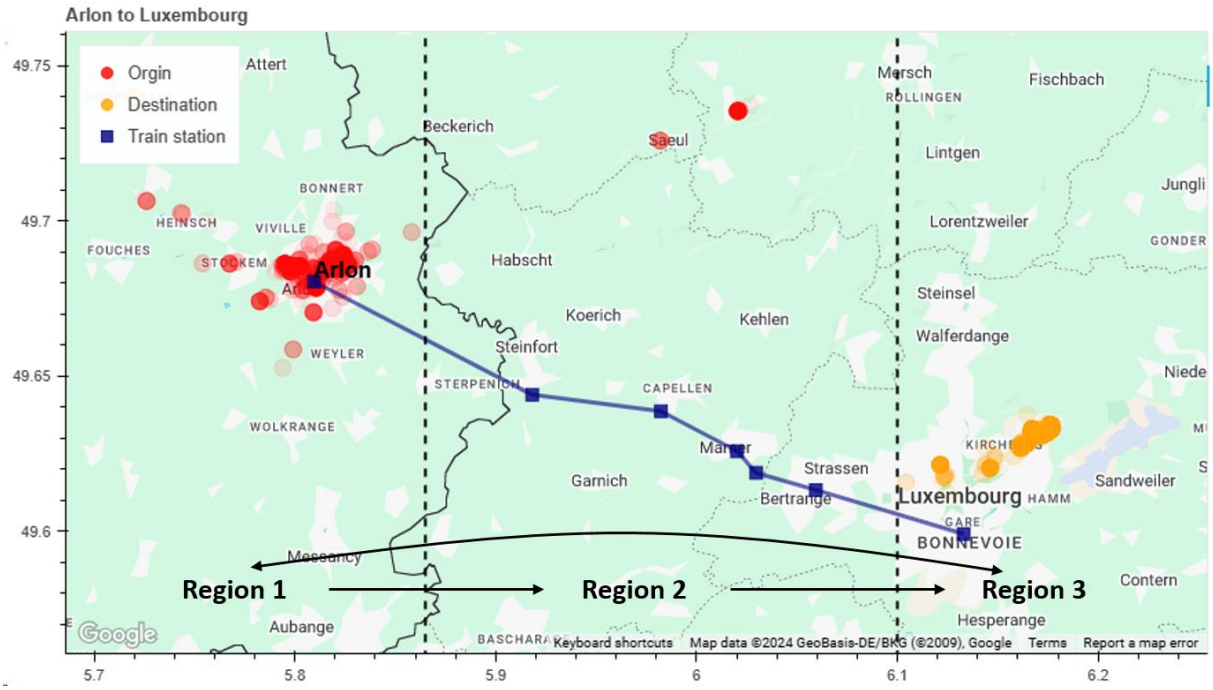


Figure 5.6: The region splits for bus speed estimation

The average speed derived from Kussbus data are presented in Table 5.9 as a speed matrix. The service operates during morning peak hours where all customers commute from Arlon to Luxembourg. As a results, trips from Region 2 to Region 1 and from Region 3 to Region 2 are not observed in the Kussbus's ride data. To estimate the average bus speed in these directions, the overall average speed is assumed for these two directions. Besides, preliminary tests using the Google Maps API indicate that travel from Region 3 to Region 1 is the fastest, as this route predominantly uses highways and experiences no traffic congestion in this direction. However, according to Kussbus data, buses traveling from region 3 to region 1 typically do not have customers on board, and such trips involve waiting time, resulting in a significantly lower average speed than what would be observed in reality. To better reflect real-world conditions, the speed for this section is retrieved from the maximum speed in the matrix in Table 5.9.

Table 5.9: Bus travel speed matrix between regions

Origin \ Destination	Region 1 (Arlon)	Region 2	Region 3 (Luxembourg)
Region 1 (Arlon)	17.8 km/h	49.6 km/h	38.1 km/h
Region 2	48.3 km/h[1]	43.3 km/h	42.1 km/h
Region 3 (Luxembourg)	49.6 km/h[2]	48.3 km/h[1]	21.6 km/h

1. Overall average speed. 2. The maximum speed in the speed matrix.

Since the Kussbus data does not include depot locations, we assume one depot is located at Luxembourg and another depot is at Arlon train stations. Each depot is equipped with one DC fast charger, offering a charging speed of 50 kWh. For the electric bus fleet, we assume five homogeneous electric buses in the bus fleet, same as observed maximum fleet size used in Kussbus's data. Each electric bus has a passenger capacity of 15 and a battery capacity of 69 kWh, with an energy consumption rate of 0.552 kWh/km (bus type 1 in Table 4.2). All the other system parameters are consistent with those in Table 4.2.

5.2.2 Results from customer's perspectives

Figure 5.7 (a) presents a density plot of 1143 requests' travel times across five different transport service options, which is calculated as the difference between their arrival time at destinations and the departure time at origins. Although the results show that cars have the shortest travel time, with a prominent peak at less than 40 minutes, EIDRT and EDRT closely follow. The distribution of Kussbus, EDRT and EIDRT exhibit similar patterns but with wider dispersion compared with the other two. The peak of Kussbus distribution is slightly higher than that of EIDRT and EDRT, indicating longer customer travel times. PT has the longest customer travel times, with a peak around 80 minutes.

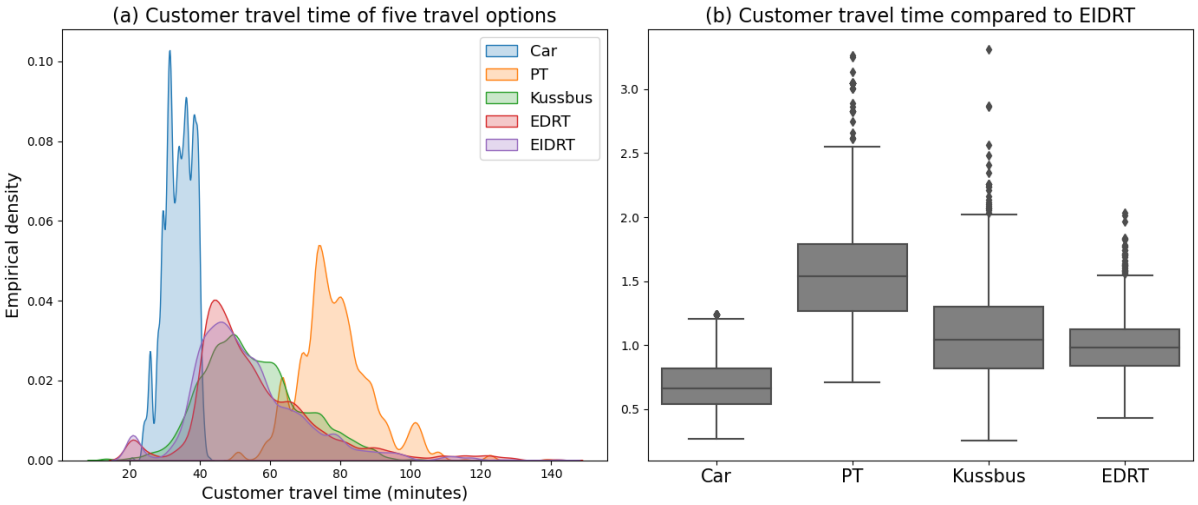


Figure 5.7: Overview of customer travel time

Figure 5.7 (b) shows box plots of customer travel times for car, PT, Kussbus and EDRT normalized by the individual travel times of EIDRT (i.e., a value less than 1.0 indicates a shorter travel time than EIDRT, while a value greater than 1.0 indicates a longer travel time). Most customers traveling by car experience shorter travel times than EIDRT; however, some trips are slower due to traffic congestion during morning peak hours, which reduces driving speeds below train speeds. PT has significantly higher customer travel times than EIDRT, with a median travel time approximately 1.5 times greater. The majority of customer travel times for Kussbus and EDRT are comparable to EIDRT, although some Kussbus users experience travel times exceeding twice those of EIDRT.

As both PT and EIDRT involve multiple transport modes, Figure 5.8 presents the average customer travel time for each day and its composition on used transport modes. Note that PT includes a tram service, while EIDRT only considers train, bus and walking. Similar to the results shown in Figure 5.7, the travel time of EIDRT is significantly less than that of the PT option. One main reason for such a difference is the effect of walking. The EIDRT limits the maximum walking distance of customers, and minimizes customer travel time. Figure 5.8 shows that bus service dominates travel time in PT, whereas EIDRT reduces customer in-bus travel time significantly and improves customer in-train travel time. The on-demand bus service in EIDRT is more efficient than existing fixed-route bus service within PT.

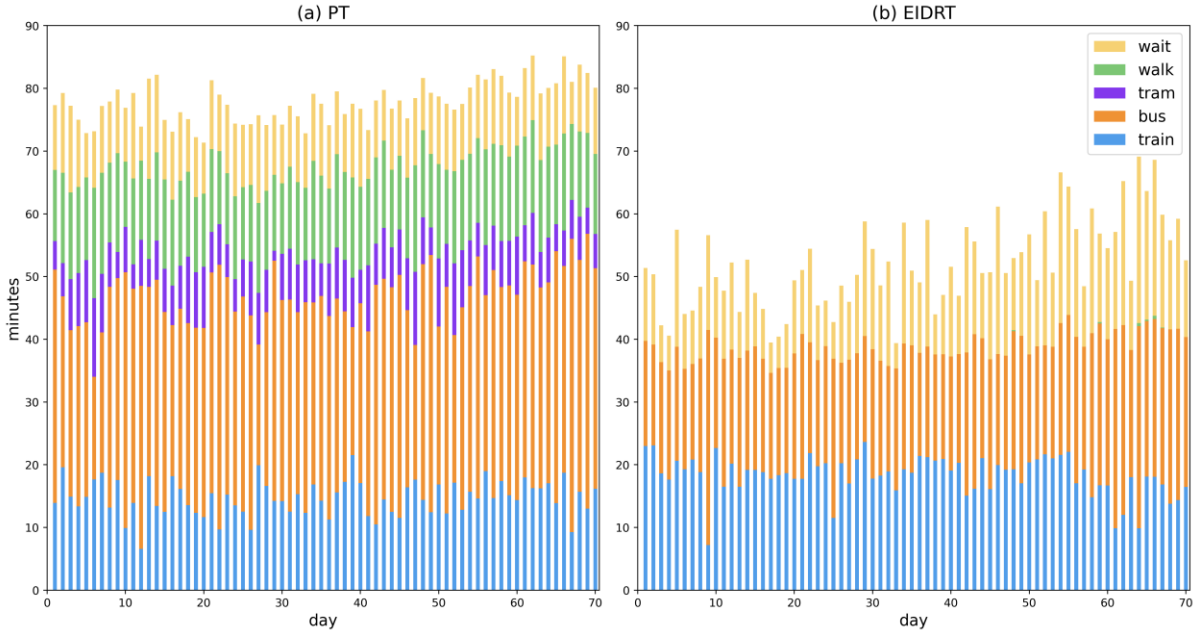


Figure 5.8: Average customer travel time per day and its decomposition for PT and EIDRT

Figure 5.9 provides a detailed analysis on 1143 customers' trips, including in-train travel time (Figure 5.9 (a)), in-bus travel time (Figure 5.9 (b)), waiting time (Figure 5.9 (c)) and walking time (Figure 5.9 (d)). For in-train travel time, the number of trips with zero in-train travel time (indicating no use of train services) under PT is three times higher than that under EIDRT. EIDRT demonstrates higher train utilization with 989 out of 1143 trips involving train services, compared to 641 trips for PT. Most customers of PT have in-train travel time around 20 and 30 minutes, reflecting mainly the express and slow departures from Arlon station to Luxembourg City. In comparison, EIDRT provides greater flexibility by sending most customers to the express departures. Figure 5.9 (b) highlights in-bus travel time. For PT, a significant proportion of customers experience bus travel times exceeding 40 minutes, while EIDRT provides more consistent in-bus travel times with a peak around 20 minutes. This reflects

that EIDRT has more balanced integration of train and bus services. In terms of waiting time, although most customers under EIDRT experience waiting times of less than 20 minutes, some have no waiting time. The distribution of EIDRT's waiting times is more dispersed compared to PT, as illustrated in Figure 5.9 (c). As Kussbus service also includes walking legs, Figure 5.9 (d) presents box plots comparing customer walking times for PT, Kussbus, and EIDRT. Note that the EIDRT rarely involves walking where customer's walking time is constrained within 10 minutes in the problem setting. Consequently, using EIDRT results in the shortest walking times among the three options. Kussbus's walking time is mainly shorter than 10 minutes, while PT results in significantly longer walking time for the majority of customers.

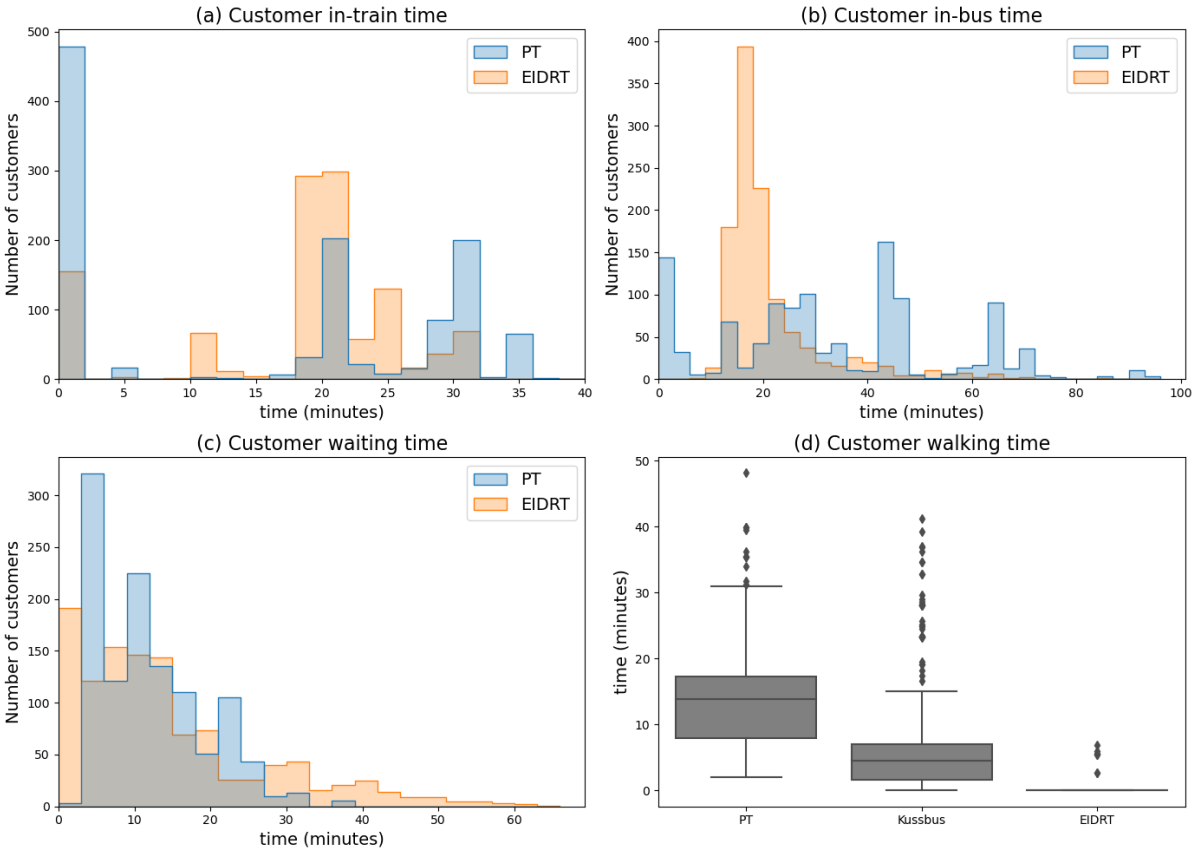


Figure 5.9: Distributions of customer's travel experiences for PT and EIDRT

5.2.3 Results from bus operator's perspectives

As Kussbus, EDRT and EIDRT involve on-demand bus services, this section focuses on comparing these three transport modes from the operator's perspective (see Figure 5.10). The bus VKT refers to the total vehicle kilometers traveled by buses during the analyzed period, while bus empty VKT represents the total vehicle kilometers travelled by buses with no customer on board. Note that bus empty VKT excludes the distance from and to depots, as this information is not included in Kussbus's data. Figure 5.10 (a) shows that EDRT has the longest bus VKT, followed by Kussbus and EIDRT. The EIDRT demonstrates significantly lower VKT because some customers' journeys are replaced by train services. The VKT distribution of Kussbus exhibits two distinct peaks, attributed to Kussbus's deadhead trips from Luxembourg to Arlon (see the empty VKT of Figure 5.10 (c)). EDRT has most buses with no empty VKT, but it has the empty VKT around 70 km because some buses make multiple trips from

Luxembourg to Arlon without customers onboard. In contrast, EIDRT buses rarely undertake long deadhead trips with most in between 0 and 20 km.

Due to the long deadhead trips, Kussbus and EDRT operates fewer buses than EIDRT as depicted in Figure 5.10 (b). Another reason that EIDRT utilizes more buses is because customers traveling by train are served by two different buses: one for the first-mile service in Arlon and another for the last-mile service in Luxembourg City. Moreover, Figure 5.10 (d) shows that the occupancy of EIDRT buses is significantly higher than Kussbus. EIDRT has a maximum of 21 customers per bus, while only 14 for Kussbus and EDRT. The higher number of served customers per bus further explains EIDRT's reduced bus empty VKT.

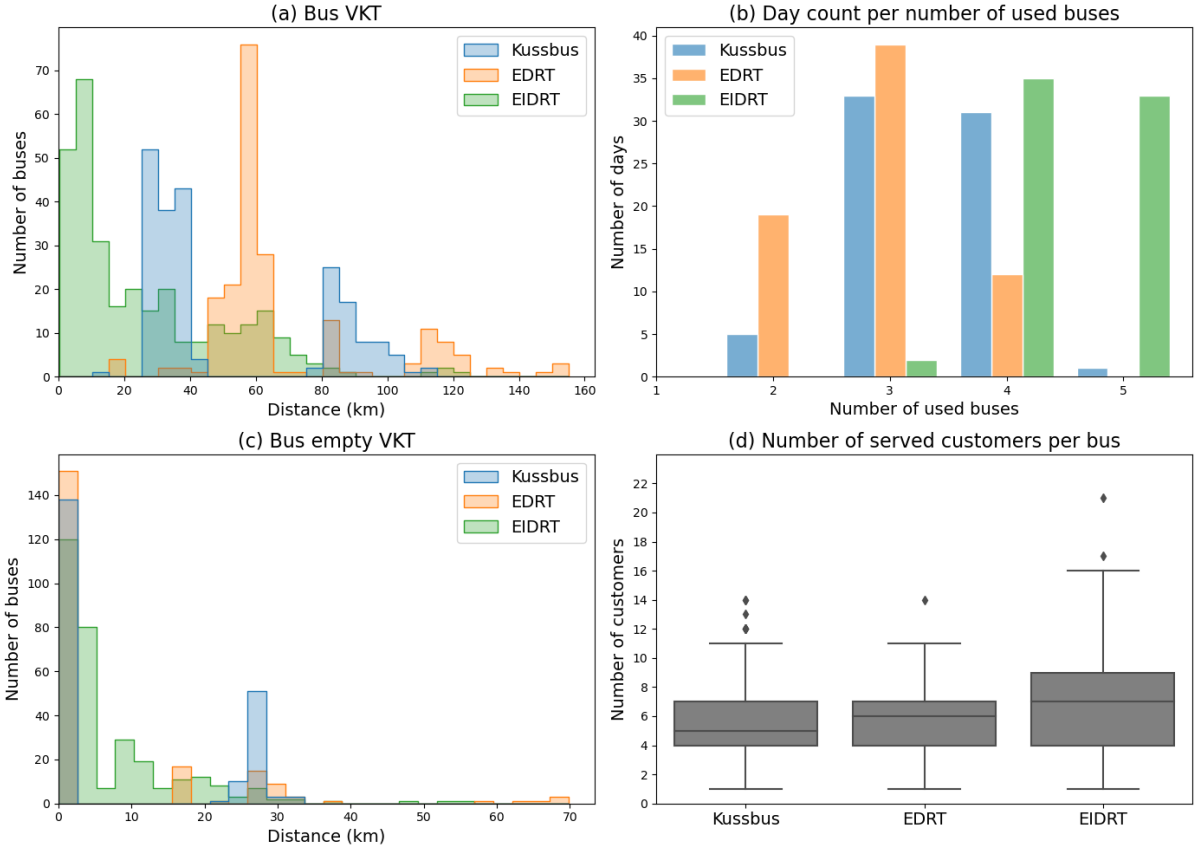


Figure 5.10: KPIs from operator's perspective for Kussbus, EDRT and EIDRT

For charging operations, Figure 5.11 illustrates the total daily charging time for electric buses for the EDRT and EIDRT over 70 days. It shows that EDRT generally requires more frequent and longer charging times compared to EIDRT. This difference is attributed to EIDRT's integration with train services, which significantly reduces bus VKT and, consequently, energy consumption. Notably, at day 43, EIDRT requires charging while EDRT does not. This is because there is one bus in EIDRT travels a significantly greater distance than the others, leading to the need of additional charging.

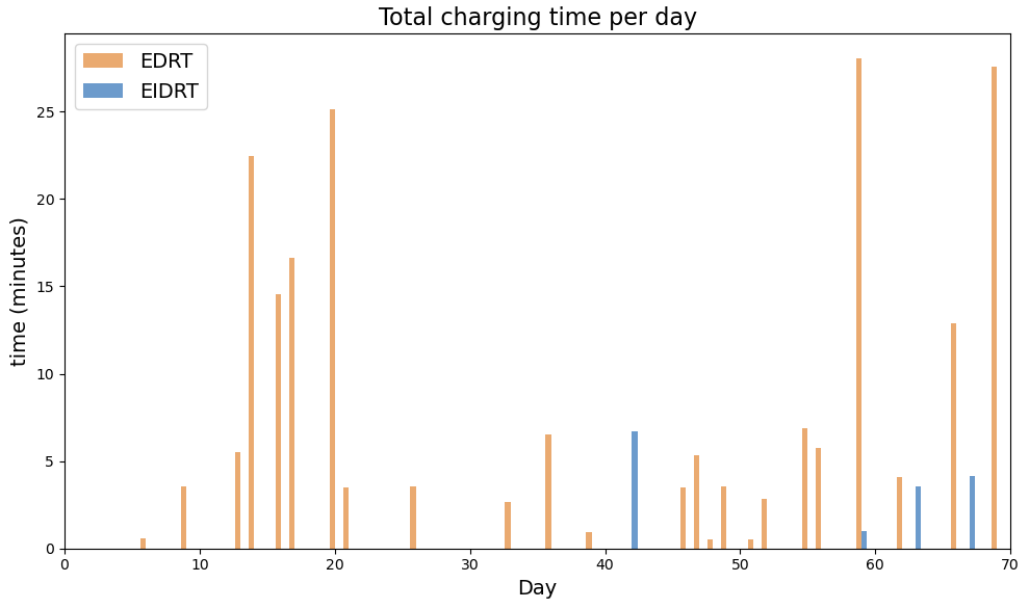


Figure 5.11: The daily charging time of electric buses over 70 days for EDRT and EIDRT

5.2.4 Comparisons of door-to-door, meeting-point based and integrated DRT services

This section investigates three types of DRT services: door-to-door DRT, meeting-point based DRT and IDRT, where the meeting-point based DRT is introduced in Chapter 2, while the IDRT is discussed in Chapter 3 and Chapter 4. This section compares these three services by assessing the results from EDRT, Kussbus and EIDRT, respectively. The summarized results are presented in Table 5.10.

Table 5.10: Results of DRT, IDRT and meeting-point based DRT in the case study

	Customer perspective			**Bus operator's perspective**		
	Avg. Customer travel time[1]	Avg. walking time	Avg. waiting time	Avg. Bus VKT	Avg. daily # used bus	Avg. # cus/bus[3]
Door-to-door DRT	55.0	0.0	7.1	66.6	2.9	5.6
Meeting-point based DRT	55.0	5.2	N.A.[2]	51.4	3.4	5.5
IDRT	54.5	0.1	15.0	26.5	4.4	6.9

Notes: 1. Difference between arrival time at destination and the departure time at origin. 2. Data not available from Kussbus data. 3. Number of served customer per bus.

Consistent with the findings in Figure 5.7, the customer travel times are similar for the three services, but these three DRT services exhibit higher variance in travel times compared to car and PT. Due to the requirement for customers to walk to a meeting point in the meeting-point-based DRT, the average walking time for this service is higher than the other two. Regarding waiting time, the IDRT has an average waiting time that is twice as that of the door-to-door DRT, while waiting time data for the meeting-point-based DRT is unavailable in the Kussbus dataset.

From the bus operator's perspective, the IDRT and meeting-point based DRT achieve significant reductions in bus VKT, saving approximately 60.2% and 22.8% compared to the door-to-door DRT.

These reductions are primarily due to customer walking to meeting points in the meeting-point based DRT and the use of train services in the IDRT. However, the additional VKT of the door-to-door DRT partially attributed to its recharging activities, while Kussbus operates with internal combustion engine vehicles that do not require charging. The decrease in bus VKT for IDRT is also attributed to the higher bus occupancy as reflected in the average number of served customers per bus. Despite the substantial reduction in bus VKT, the number of used buses in the IDRT is increased compared to the other two services.

5.3 Discussion

This chapter first conducts a set of experiments on scenarios resembling real-world problem size to analyze the performance of the EIDRT service based on the EIDARP parameter analysis. The results provides insights for on-demand bus operators to consider the trade-off between operational costs and customer inconvenience. The findings indicate that the relative weights in the objective function and customer maximum ride time constraint (detour factor) significantly impact on the performance of the EIDRT system. The results show that a higher weight on customer travel time in the objective function yields similar outcomes to a lower detour factor. Although a lower bus VKT can be achieved by increasing the detour factors (or reducing the weigh on customer travel time), it negatively impacts customer's experiences. While the integrated service reduces bus VKT compared to the non-integrated service (e.g., DARP), the number of used buses is not necessarily due to customers' first and last miles are served by different bus. It was founded that relatively small fleet size leads to greater customer rejection, indicating that high-quality EIDRT services cost similarly to those of non-integrated service in terms of fleet purchasing. Maximum waiting time does not appear to be critical to overall performance when it is set to more than 10 minutes in the scenario, as the maximum waiting time is restricted at transit station and the maximum travel time constraint already partially control it in the EIDARP.

For bus operators, selecting an appropriate service area is crucial for the efficient operation of an EIDRT service. One key consideration is that bus-travelling speed should be lower than transit speed. If the bus speed is competitive as transit speed, fewer customers are likely to use transit services, which undermines the effectiveness of transit integration in the system. When transit service is more frequent, it benefits both operators and customers, though the improvements are not substantial. This aligns with the findings of Molenbruch et al. (2021) as DRT buses can flexibly optimize their routes and adapt to varying transit timetables. Besides, the scenario with four transit lines achieves the minimum objective function value across all experiments. This configuration results in both the shortest bus travel time and customer travel time, demonstrating that the EIDRT service operates more efficiently in areas with highly connected transit network. Regarding the operations of electric buses, the results suggest that a fully recharged bus fleet at the beginning of the service ensure better service quality.

The conducted case study compared the proposed EIDRT service with the other existing transport options, highlighting the advantages of EIDRT. The results demonstrate that the EIDRT effectively reduces overall operational costs while maintaining desired level of service for customers. Specifically, EIDRT reduces customer travel time by nearly half compared to PT and improves the utilization rate of train services. Although car travel remains the fastest option overall, some EIDRT trips achieve travel times faster than car trips.

Among the three types of DRT services (i.e., door-to-door, meeting-point-based, and integrated), the IDRT service achieves a reduction of more than 50% in bus VKT compared with the other two services.

Although IDRT has higher initial fleet purchasing costs than non-integrated services, its reduced bus travel costs will lead to long-term operational savings. Despite this significant reduction in VKT, the average customer travel time for IDRT remains similar. However, customers might experience higher waiting times in the IDRT.

Chapter 6 Conclusions and future work

This Chapter discusses the contributions and limitations of the dissertation. Then it answers the research questions raised in Chapter 1. Future research directions for EIDRT are provided at the end.

6.1 Contributions and limitations

The IDRT service has received increasing attention in recent years. This innovative transportation service helps the operators of demand-responsive vehicles reduce operational costs while enhancing ridership for existing mass transit services. Several studies have demonstrated the benefits of the integrating DRT with fixed-route transit, yet challenges remain in ensuring seamless and convenient transfers for customers. Furthermore, including EVs into such services introduces additional challenges due to their limited driving range and charging operation scheduling. The dissertation examined an EIDRT service that incorporates transit synchronization and utilization of EVs.

To address the complexity of the EIDRT, Chapter 2 first investigated a meeting-point based first-mile DRT service using electric buses. In this service, DRT buses serve as connectors to the transit stations while accounting for the constraints for charging operations and customer waiting time at transit stations. Customers can be dropped off at multiple transit stations. We proposed a MILP formulation and a hybrid DA-based metaheuristic algorithm. The hybrid DA-based algorithm efficiently schedules charging operations for electric buses. A layered departure-expanded graph was proposed to efficiently model customer waiting time constraints at transit stations and reduce the search space of optimal solution. The solution algorithm was tested on a series of test instances, showing its efficiency to obtain good solutions with up to hundreds of requests. However, the solution quality of the proposed hybrid metaheuristic can still be improved. In the current approach, customer-meeting point assignments are solved at the first stage and then remain unchanged for bus route optimization, resulting in suboptimal solutions. One potential improvement is jointly optimizing the customer-to-meeting-point assignment and bus routing. This would need to design efficient data structure and new search operators. As the objective of the service is to minimize both bus travel costs and customer inconvenience (e.g., excessive waiting time and walking distance), tailored search operators could be developed to better incorporate these factors.

The first-mile meeting-point based EDRT was then expanded to the EIDRT, which is the first study involving electric vehicles in an integrated DRT service. The departure-expanded graph concept and approaches for handling capacitated charging stations were extended and improved to formulate the EIDARP for this service. The objective of this problem minimizes both bus operational costs and customer travel times. An MILP formulation was developed using the departure-expanded graph, and we further improved the efficiency of capacitated charging station constraint modeling to reduce unnecessary node duplications. The departure-expanded graph approach significantly improves computational efficiency compared to existing state-of-the-art methods. To solve the problem to larger problem sizes, a hybrid LNS metaheuristic was developed and tested with instances with up to 100 customers and two transit lines. The solution was benchmarked against an MILP solver (Gurobi), demonstrating better solution qualities with substantial reductions in computational time. The proposed EIDRT service was further analyzed under different configurations within the EIDARP framework and compared with alternative transportation options through a numerical case study in the Arlon-

Luxembourg corridor. Results indicate that the EIDRT system effectively reduces bus travel costs while significantly lowering customer travel times compared to conventional public transportation.

Despite these contributions, several limitations remain. Although the hybrid LNS significantly improves computational efficiency compared to the MILP solver, further improvement is necessary for large-scale applications. Improving computational efficiency would allow for more extensive analyses of the system configurations and real-time applications of EIDRT. For example, the Arlon-Luxembourg case study is based on an unbalanced demand scenario with only one transit line. With a more efficient solution algorithm, more comprehensive case studies with varying scenarios and transit networks could be conducted to better evaluate EIDRT performance and derive more generalizable insights. Additionally, testing the impacts of different customer demand distributions would provide a more comprehensive assessment of the EIDRT system's effectiveness under different conditions.

6.2 Answers to the research questions

This section provides the responses to the research questions from Chapter 1.

Q1: How can the recharging of electric vehicles in EDRT services be efficiently scheduled while ensuring a high level of customer service?

We modeled two types of EDRT services, a meeting-point-based EDRT and an integrated EDRT, named MP-EFCS and EIDARP, respectively. Both problems involve charging scheduling while accounting for capacitated charging station constraints. To optimize charging schedules, the MILP formulations in EIDARP improve the charging station capacity constraints by reducing the number of replicated dummy nodes. In the metaheuristic for MP-EFCS, the time complexity for buses' charging operations conflict checks is $O(1)$, allowing fast checking the feasibility of charging operations of solutions. For charging scheduling in the hybrid LNS algorithm developed for the EIDARP, a greedy insertion strategy is used to schedule charging operations at the earliest available time. This approach does not increase CPU time significantly when buses require recharging.

To maintain service quality, MP-EFCS includes charging time minimization in its objective function, enhancing vehicle availability and reducing charging impacts on customer experience. In contrast, EIDARP does not explicitly minimize charging time but buses only charge as needed to serve customers. Customer travel experience remains unaffected, as customer travel times are minimized in EIDARP's objective function.

Q2: How to manage customers' inter-modal transfers in EIDRT services to minimize their inconvenience?

Since transfers may influence customer willingness to use the EIDRT service, EIDARP incorporates maximum waiting time constraints for transfers between buses and transit services. When this limit is set low, waiting times remain within the specified range. However, increasing the maximum waiting time allowance does not proportionally extend experienced waiting times of users. This is because the maximum travel time constraint of users partially regulates waiting times at transit stations. Moreover, reducing the detour factor further decreases overall waiting time. Therefore, intermodal transfers are managed through both maximum waiting time constraints at transit stations and maximum customer travel time detour limits.

Q3: What are the key factors in determining a desirable service operation for real-world applications of EIDRT services?

A well-designed EIDRT service should meet customer travel needs while maintaining relatively low operational costs. In this service, customers can specify their preferred pickup or drop-off times. Customer journey time should remain reasonable. Like other dial-a-ride services, these requirements are incorporated as constraints in the EIDARP.

Operators could also determine the appropriate weight balance in EIDARP's objective function between customer travel inconvenience and bus travel costs. While prioritizing bus travel costs reduces operational expenses, it can increase customer inconvenience. Operators need to assess this trade-off based on their service goals. Although fleet size decision is not the focus of this dissertation, our findings indicate that maintaining a high service level in EIDRT might require more buses than a traditional door-to-door EDRT service.

However, as EIDRT involves transit services, the performance of the EIDRT service is also influenced by the characteristics of the existing transit network within a service area. Before implementing EIDRT, operators should evaluate transit service conditions within a service area. Transit speed should generally exceed the average bus speed, as lower transit speeds reduce the number of customers opting for transit. When demand is uniformly distributed, a more accessible transit network reduces travel times for both the operator and customers. While different demand distributions were not tested, results from the Arlon-Luxembourg case study indicate that EIDRT can significantly reduce operational costs, even with a single transit line. In this scenario, customer demand is unbalanced during morning peak hours, but the transit line aligns well with customers' travel direction. These findings suggest that improving customer access to transit services could enhance the benefits of integrating transit into the (E)DRT system.

Q3: Are EIDRT services more beneficial for customers and service operators compared to existing transport services?

The case study in Section 5.2 compares EIDRT with other transport services along the corridor from Arlon to Luxembourg during morning peak hours. From customers' perspectives, EIDRT offers shorter travel times than PT, though customers' waiting times in EIDRT service are not necessarily reduced. Customer travel experiences are similar across EIDRT, door-to-door EDRT and meeting-point-based EDRT, but EIDRT significantly reduces bus operational costs compared to the other two. It is important to note that this result is based on an unbalanced demand scenario with a single transit line. The performance of the EIDRT service may vary in different service areas, transit network configurations, and demand conditions.

6.3 Future work for the Electric Integrated Demand-Responsive Transport

This dissertation has explored the development and optimization of the EIDRT system, proposing effective methodologies to enhance its efficiency and practicality. While significant progress has been made, several challenges and potential improvements remain. Addressing these challenges will further advance the applicability of the EIDRT system in real-world transit networks. Future research can focus on improving solution approaches and system efficiency, integrating tactic and strategic decisions, and incorporating dynamic and stochastic elements. These efforts will not only enhance the feasibility of

EIDRT services but also contribute to broader advancements in sustainable and flexible public transportation. We have identified four directions for future research:

1. Enhancing the solution approaches for EIDARP

While we developed a metaheuristic approach to solve the EIDARP, tackling large-scale instances in complex transit networks remains challenging. Further improvements in solution algorithm design are necessary for real-world applications with intricate network layouts and timetables. In addition to enhancing the efficiency of metaheuristic methods, exact solution methods, such as the branch-and-price-and-cut algorithm, could be explored to improve solution quality.

2. Improving the EIDRT system

As this EIDRT service currently operates as a door-to-door service, its implementation in high-demand or clustered-demand areas is not efficient. Incorporating the concept of meeting points could enhance the operational efficiency and reduce costs for bus operators. From the customer perspective, minimizing inconvenience remains a key aspect to improve the service attractiveness. For instance, the number of transfers can be limited. When the transit network becomes complex, a customer journey with more number of transfers will make the service less attractive. Moreover, the system performance should also be tested at different scenarios with different transit network and demand distributions.

3. Incorporating strategic and tactic decisions

While the dissertation focuses on the operational level of the EIDRT system, future research could involve tactical or strategic decisions to optimize the system costs. For instance, determining the optimal bus fleet size is a critical factor influencing system performance. Another important aspect is the integration of charging infrastructure planning with bus routing. The location of charging stations could be jointly optimized with vehicle routing decisions or the optimized configurations of charger types (e.g., superfast, fast, slow) could also be considered to enhance operational efficiency.

4. Developing a dynamic and stochastic EIDARP

The EIDARP studied in this dissertation is formulated as a static and deterministic problem, assuming a reliable transit operation. However, in real-world applications, customers often prefer an immediate, dynamic (online) DRT service that can adapt in real time. Moreover, transit delays, disruptions, and fluctuations in demand can have a profound impact on customer experience and operation costs. To address these challenges, future research could extend the current EIDARP framework to incorporate dynamic and stochastic elements, allowing for real-time decision-making and improved adaptability. Stochastic aspects such as transit delays, stochastic customer demand, and traffic congestions should be integrated to better reflect the scenarios in practice. Moreover, a dynamic and stochastic EIDARP model could also be leveraged by ride-sourcing companies to optimize their operations, reduce costs, and enhance service reliability under uncertainties.

BIBLIOGRAPHY

Aïvodji, U.M., Gambs, S., Huguet, M.J., Killijian, M.O., 2016. Meeting points in ridesharing: A privacy-preserving approach. Transp. Res. Part C Emerg. Technol. 72, 239–253. https://doi.org/10.1016/J.TRC.2016.09.017

Aldaihani, M., Dessouky, M.M., 2003. Hybrid scheduling methods for paratransit operations. Comput. Ind. Eng. 45, 75–96. https://doi.org/10.1016/S0360-8352(03)00032-9

Alinaghian, M., Shokouhi, N., 2018. Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. Omega 76, 85–99. https://doi.org/10.1016/J.OMEGA.2017.05.002

Alonso-González, M.J., Liu, T., Cats, O., Van Oort, N., Hoogendoorn, S., 2018. The Potential of Demand-Responsive Transport as a Complement to Public Transport: An Assessment Framework and an Empirical Evaluation. Transp. Res. Rec. 2672, 879–889. https://doi.org/10.1177/0361198118790842/ASSET/IMAGES/LARGE/10.1177_0361198118790842-FIG6.JPEG

Ammous, M., Belakaria, S., Sorour, S., Abdel-Rahim, A., 2019. Optimal Cloud-Based Routing with In-Route Charging of Mobility-on-Demand Electric Vehicles. IEEE Trans. Intell. Transp. Syst. 20, 2510–2522. https://doi.org/10.1109/TITS.2018.2867519

Arnold, F., Sörensen, K., 2019. Knowledge-guided local search for the vehicle routing problem. Comput. Oper. Res. 105, 32–46. https://doi.org/10.1016/J.COR.2019.01.002

ARUP, 2021. Integrated demand responsive transport in cities.

Behroozi, M., 2023. Understanding the Impact of Ridesharing Services on Traffic Congestion. Reengineering Shar. Econ. 119–145. https://doi.org/10.1017/9781108865630.011

Bian, Z., Liu, X., 2019. Mechanism design for first-mile ridesharing based on personalized requirements part I: Theoretical analysis in generalized scenarios. Transp. Res. Part B Methodol. 120, 147–171. https://doi.org/10.1016/J.TRB.2018.12.009

Bongiovanni, C., Kaspi, M., Geroliminis, N., 2019. The electric autonomous dial-a-ride problem. Transp. Res. Part B Methodol. 122, 436–456. https://doi.org/10.1016/J.TRB.2019.03.004

Braekers, K., Caris, A., Janssens, G.K., 2014. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. Transp. Res. Part B Methodol. 67, 166–186. https://doi.org/10.1016/j.trb.2014.05.007

Bruglieri, M., Mancini, S., Pisacane, O., 2019. The green vehicle routing problem with capacitated alternative fuel stations. Comput. Oper. Res. 112, 104759. https://doi.org/10.1016/J.COR.2019.07.017

Calabrò, G., Le Pira, M., Giuffrida, N., Inturri, G., Ignaccolo, M., Correia, G.H. d. A., 2022. Fixed-Route vs. Demand-Responsive Transport Feeder Services: An Exploratory Study Using an Agent-Based Model. J. Adv. Transp. 2022, 8382754. https://doi.org/10.1155/2022/8382754

Chen, P.W., Nie, Y.M., 2017. Analysis of an idealized system of demand adaptive paired-line hybrid transit. Transp. Res. Part B Methodol. 102, 38–54. https://doi.org/10.1016/J.TRB.2017.05.004

Chowdhury, S., Ceder, A., 2016. Users' willingness to ride an integrated public-transport service: A literature review. Transp. Policy 48, 183–195. https://doi.org/10.1016/J.TRANPOL.2016.03.007

Cordeau, J.F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54, 573–586. https://doi.org/10.1287/OPRE.1060.0283

Cordeau, J.F., Laporte, G., 2007. The dial-a-ride problem: Models and algorithms. Ann. Oper. Res. 153, 29–46. https://doi.org/10.1007/s10479-007-0170-8

Cordeau, J.F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp. Res. Part B Methodol. 37, 579–594. https://doi.org/10.1016/S0191-2615(02)00045-0

Crainic, T.G., Kim, K.H., 2007. Chapter 8 Intermodal Transportation. Handbooks Oper. Res. Manag. Sci. 14, 467–537. https://doi.org/10.1016/S0927-0507(06)14008-6

Currie, G., Fournier, N., 2020. Why most DRT/Micro-Transits fail – What the survivors tell us about progress. Res. Transp. Econ. 83, 100895. https://doi.org/10.1016/J.RETREC.2020.100895

Czioska, P., Kutadinata, R., Trifunović, A., Winter, S., Sester, M., Friedrich, B., 2019. Real-world meeting points for shared demand-responsive transportation systems. Public Transp. 11, 341–377.

https://doi.org/10.1007/S12469-019-00207-Y/TABLES/2
Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271. https://doi.org/10.1007/BF01386390/METRICS
Dragomir, A.G., Doerner, K.F., 2020. Solution techniques for the inter-modal pickup and delivery problem in two regions. Comput. Oper. Res. 113, 104808. https://doi.org/10.1016/J.COR.2019.104808
Ellegood, W.A., Solomon, S., North, J., Campbell, J.F., 2020. School bus routing problem: Contemporary trends and research directions. Omega 95, 102056. https://doi.org/10.1016/J.OMEGA.2019.03.014
Enoch, M., Potter, S., Parkhurst, G., Smith, M., 2006. Why do demand responsive transport systems fail?, in: Enoch, M., Potter, S., Parkhurst, G., Smith, M. (Eds.), Transportation Research Board 85th Annual Meeting. Washington DC.
Errico, F., Crainic, T.G., Malucelli, F., Nonato, M., 2013. A survey on planning semi-flexible transit systems: Methodological issues and a unifying framework. Transp. Res. Part C Emerg. Technol. 36, 324–338. https://doi.org/10.1016/J.TRC.2013.08.010
Fang, Y., Ma, T.-Y., Viti, F., 2024. Electric integrated dial-a-ride services with capacitated charging stations, multiple depots, and customer rejections. https://hal.science/hal-04765441
Fang, Y., Ma, T.Y., 2023. Demand Responsive Feeder Bus Service Using Electric Vehicles with Timetabled Transit Coordination, in: Lecture Notes in Intelligent Transportation and Infrastructure. Springer Nature, pp. 91–103. https://doi.org/10.1007/978-3-031-23721-8_7/COVER
Fehn, F., Noack, F., Busch, F., 2019. Modeling of Mobility On-Demand Fleet Operations Based on Dynamic Electricity Pricing. MT-ITS 2019 - 6th Int. Conf. Model. Technol. Intell. Transp. Syst. https://doi.org/10.1109/MTITS.2019.8883370
Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G., 2014. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transp. Res. Part E Logist. Transp. Rev. 71, 111–128. https://doi.org/10.1016/j.tre.2014.09.003
Fielbaum, A., Tirachini, A., Alonso-Mora, J., 2024. Improving public transportation via line-based integration of on-demand ridepooling. Transp. Res. Part A Policy Pract. 190, 104289. https://doi.org/10.1016/J.TRA.2024.104289
Fiori, C., Ahn, K., Rakha, H.A., 2016. Power-based electric vehicle energy consumption model: Model development and validation. Appl. Energy 168, 257–268. https://doi.org/10.1016/J.APENERGY.2016.01.097
Froger, A., Jabali, O., Mendoza, J.E., Laporte, G., 2021. The Electric Vehicle Routing Problem with Capacitated Charging Stations. Transp. Sci. 56, 460–482. https://doi.org/10.1287/TRSC.2021.1111
Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Comput. Oper. Res. 104, 256–294. https://doi.org/10.1016/J.COR.2018.12.013
Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2017. A Matheuristic for the Electric Vehicle Routing Problem with Capacitated Charging Stations. Cent. Interuniv. Rech. sur les reseaux d'entreprise, la logistique le Transp.
Galarza Montenegro, B.D., Sörensen, K., Vansteenwegen, P., 2021. A large neighborhood search algorithm to optimize a demand-responsive feeder service. Transp. Res. Part C Emerg. Technol. 127, 103102. https://doi.org/10.1016/J.TRC.2021.103102
George, S.R., Marzia, Z., 2018. Electrifying the Ride-Sourcing Sector in California. San Francisco.
Ghilas, V., Cordeau, J.F., Demir, E., Van Woensel, T., 2018. Branch-and-Price for the Pickup and Delivery Problem with Time Windows and Scheduled Lines. https://doi.org/10.1287/trsc.2017.0798 52, 1191–1210. https://doi.org/10.1287/TRSC.2017.0798
Ghilas, V., Demir, E., Van Woensel, T., 2016. An adaptive large neighborhood search heuristic for the Pickup and Delivery Problem with Time Windows and Scheduled Lines. Comput. Oper. Res. 72, 12–30. https://doi.org/10.1016/J.COR.2016.01.018
Gkiotsalitis, K., 2022. Coordinating feeder and collector public transit lines for efficient MaaS services. EURO J. Transp. Logist. 11, 100057. https://doi.org/10.1016/j.ejtl.2021.100057

Gschwind, T., Drexl, M., 2019. Adaptive Large Neighborhood Search with a Constant-Time Feasibility Test for the Dial-a-Ride Problem. https://doi.org/10.1287/trsc.2018.0837 53, 480–491. https://doi.org/10.1287/TRSC.2018.0837

Haglund, N., Mladenović, M.N., Kujala, R., Weckström, C., Saramäki, J., 2019. Where did Kutsuplus drive us? Ex post evaluation of on-demand micro-transit pilot in the Helsinki capital region. Res. Transp. Bus. Manag. 32, 100390. https://doi.org/10.1016/J.RTBM.2019.100390

Häll, C.H., Andersson, H., Lundgren, J.T., Värbrand, P., 2009. The Integrated Dial-a-Ride Problem. Public Transp. 1, 39–54. https://doi.org/10.1007/S12469-008-0006-1/METRICS

Henao, A., Marshall, W.E., 2019. The impact of ride-hailing on vehicle miles traveled. Transportation (Amst). 46, 2173–2194. https://doi.org/10.1007/S11116-018-9923-2/TABLES/6

Hickman, M., Blume, K., 2001. Modeling Cost and Passenger Level of Service for Integrated Transit Service 233–251. https://doi.org/10.1007/978-3-642-56423-9_14

Jenn, A., 2019. Electrifying Ride-Sharing: Transitioning to a Cleaner Future, UC Davis Policy Briefs . Davis.

Jung, J., Chow, J.Y.J., Jayakrishnan, R., Park, J.Y., 2014. Stochastic dynamic itinerary interception refueling location problem with queue delay for electric taxi charging stations. Transp. Res. Part C Emerg. Technol. 40, 123–142. https://doi.org/10.1016/J.TRC.2014.01.008

Kancharla, S.R., Ramadurai, G., 2020. Electric vehicle routing problem with non-linear charging and load-dependent discharging. Expert Syst. Appl. 160, 113714. https://doi.org/10.1016/J.ESWA.2020.113714

Keskin, M., Çatay, B., 2016. Partial recharge strategies for the electric vehicle routing problem with time windows. Transp. Res. Part C Emerg. Technol. 65, 111–127. https://doi.org/10.1016/j.trc.2016.01.013

Keskin, M., Laporte, G., Çatay, B., 2019. Electric Vehicle Routing Problem with Time-Dependent Waiting Times at Recharging Stations. Comput. Oper. Res. 107, 77–94. https://doi.org/10.1016/J.COR.2019.02.014

Kim, M. (Edward), Schonfeld, P., 2014. Integration of conventional and flexible bus services with timed transfers. Transp. Res. Part B Methodol. 68, 76–97. https://doi.org/10.1016/j.trb.2014.05.017

Kucukoglu, I., Dewil, R., Cattrysse, D., 2021. The electric vehicle routing problem and its variations: A literature review. Comput. Ind. Eng. 161, 107650. https://doi.org/10.1016/J.CIE.2021.107650

Lam, E., Desaulniers, G., Stuckey, P.J., 2022. Branch-and-cut-and-price for the Electric Vehicle Routing Problem with Time Windows, Piecewise-Linear Recharging and Capacitated Recharging Stations. Comput. Oper. Res. 145, 105870. https://doi.org/10.1016/J.COR.2022.105870

Lee, A., Savelsbergh, M., 2017. An extended demand responsive connector. EURO J. Transp. Logist. 6, 25–50. https://doi.org/10.1007/s13676-014-0060-6

Liaw, C.F., White, C.C., Bander, J., 1996. A decision support system for the bimodal dial-a-ride problem. IEEE Trans. Syst. Man, Cybern. Part ASystems Humans. 26, 552–565. https://doi.org/10.1109/3468.531903

Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhkopf, T., Sass, R., Hutter, F., 2022. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. J. Mach. Learn. Res. 23, 1–9.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. Oper. Res. Perspect. 3, 43–58.

Ma, T.-Y., Fang, Y., Connors, R.D., Viti, F., Nakao, H., 2024. A hybrid metaheuristic to optimize electric first-mile feeder services with charging synchronization constraints and customer rejections. Transp. Res. Part E Logist. Transp. Rev. 185, 103505. https://doi.org/10.1016/J.TRE.2024.103505

Ma, T.Y., 2021. Two-stage battery recharge scheduling and vehicle-charger assignment policy for dynamic electric dial-a-ride services. PLoS One 16, e0251582. https://doi.org/10.1371/JOURNAL.PONE.0251582

Ma, T.Y., Chow, J.Y.J., Klein, S., Ma, Z., 2021. A user-operator assignment game with heterogeneous user groups for empirical evaluation of a microtransit service in Luxembourg. Transp. A Transp. Sci. 17, 946–973. https://doi.org/10.1080/23249935.2020.1820625

Ma, T.Y., Fang, Y., 2021. Survey of charging management and infrastructure planning for electrified

demand-responsive transport systems: methodologies and recent developments. Eur. Transp. Res. Rev.

Ma, T.Y., Rasulkhani, S., Chow, J.Y.J., Klein, S., 2019. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. Transp. Res. Part E Logist. Transp. Rev. 128, 417–442. https://doi.org/10.1016/J.TRE.2019.07.002

Masson, R., Lehuédé, F., Péton, O., 2014. The Dial-A-Ride Problem with Transfers. Comput. Oper. Res. 41, 12–23. https://doi.org/10.1016/J.COR.2013.07.020

Melis, L., Queiroz, M., Sörensen, K., 2024. The integrated on-demand bus routing problem: Combining on-demand buses with a high-frequency fixed line public transport network. Comput. Oper. Res. 164, 106554. https://doi.org/10.1016/J.COR.2024.106554

Melis, L., Sörensen, K., 2022. The static on-demand bus routing problem: large neighborhood search for a dial-a-ride problem with bus station assignment. Int. Trans. Oper. Res. 29, 1417–1453. https://doi.org/10.1111/ITOR.13058

Molenbruch, Y., Braekers, K., Hirsch, P., Oberscheider, M., 2021. Analyzing the benefits of an integrated mobility system using a matheuristic routing algorithm. Eur. J. Oper. Res. 290, 81–98. https://doi.org/10.1016/J.EJOR.2020.07.060

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017a. The electric vehicle routing problem with nonlinear charging function. Transp. Res. Part B Methodol. 103, 87–110. https://doi.org/10.1016/j.trb.2017.02.004

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017b. The electric vehicle routing problem with nonlinear charging function. Transp. Res. Part B Methodol. 103, 87–110. https://doi.org/10.1016/J.TRB.2017.02.004

Mortazavi, A., Ghasri, M., Ray, T., 2024. Integrated Demand Responsive transport in Low-Demand Areas: A case study of Canberra, Australia. Transp. Res. Part D Transp. Environ. 127, 104036. https://doi.org/10.1016/J.TRD.2023.104036

Pantelidis, T.P., Li, L., Ma, T.-Y.Y., Chow, J.Y.J.J., Jabari, S.E.G., 2022. A Node-Charge Graph-Based Online Carshare Rebalancing Policy with Capacitated Electric Charging. Transp. Sci. 56, 654–676. https://doi.org/10.1287/trsc.2021.1058

Park, J., Kim, B.I., 2010. The school bus routing problem: A review. Eur. J. Oper. Res. 202, 311–319. https://doi.org/10.1016/J.EJOR.2009.05.017

Parragh, S.N., Doerner, K.F., Hartl, R.F., 2011. Demand Responsive Transportation. Wiley Encycl. Oper. Res. Manag. Sci. https://doi.org/10.1002/9780470400531.EORMS0243

Parragh, S.N., Doerner, K.F., Hartl, R.F., 2010. Variable neighborhood search for the dial-a-ride problem. Comput. Oper. Res. 37, 1129–1138. https://doi.org/10.1016/J.COR.2009.10.003

Pavlenko, N., Slowik, P., Lutsey, N., 2019. When does electrifying shared mobility make economic sense? - International Council on Clean Transportation [WWW Document]. URL https://theicct.org/publication/when-does-electrifying-shared-mobility-make-economic-sense/ (accessed 1.20.25).

Pimenta, V., Quilliot, A., Toussaint, H., Vigo, D., 2017. Models and algorithms for reliability-oriented Dial-a-Ride with autonomous electric vehicles. Eur. J. Oper. Res. 257, 601–613. https://doi.org/10.1016/j.ejor.2016.07.037

Pisinger, D., Ropke, S., 2019. Large neighborhood search. Int. Ser. Oper. Res. Manag. Sci. 272, 99–127. https://doi.org/10.1007/978-3-319-91086-4_4/COVER

Posada, M., Andersson, H., Häll, C.H., 2017. The integrated dial-a-ride problem with timetabled fixed route service. Public Transp. 9, 217–241. https://doi.org/10.1007/S12469-016-0128-9/TABLES/2

Posada, M., Häll, C.H., 2020. A metaheuristic for evaluation of an integrated special transport service. Int. J. Urban Sci. 24, 316–338. https://doi.org/10.1080/12265934.2019.1709533

Ronald, N., Thompson, R., Winter, S., 2015. Simulating Demand-responsive Transportation: A Review of Agent-based Approaches. Transp. Rev. 35, 404–421. https://doi.org/10.1080/01441647.2015.1017749

Ropke, S., Pisinger, D., 2006. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. https://doi.org/10.1287/trsc.1050.0135 40, 455–472. https://doi.org/10.1287/TRSC.1050.0135

Santini, A., Ropke, S., Hvattum, L.M., 2018. A comparison of acceptance criteria for the adaptive large

neighbourhood search metaheuristic. J. Heuristics 24, 783–815. https://doi.org/10.1007/S10732-018-9377-X/TABLES/5

Şatir Akpunar, Ö., Akpinar, Ş., 2021. A hybrid adaptive large neighbourhood search algorithm for the capacitated location routing problem. Expert Syst. Appl. 168, 114304. https://doi.org/10.1016/J.ESWA.2020.114304

Schaller, B., 2021. Can sharing a ride make for less traffic? Evidence from Uber and Lyft and implications for cities. Transp. Policy 102, 1–10. https://doi.org/10.1016/J.TRANPOL.2020.12.015

Schittekat, P., Kinable, J., Sörensen, K., Sevaux, M., Spieksma, F., Springael, J., 2013. A metaheuristic for the school bus routing problem with bus stop selection. Eur. J. Oper. Res. 229, 518–528. https://doi.org/10.1016/J.EJOR.2013.02.025

Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. 48, 500–520. https://doi.org/10.1287/TRSC.2013.0490

Schoenberg, S., Dressler, F., 2021. Reducing Waiting Times at Charging Stations with Adaptive Electric Vehicle Route Planning.

Shaw, P., 1998. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 1520, 417–431. https://doi.org/10.1007/3-540-49481-2_30

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2015. The benefits of meeting points in ride-sharing systems. Transp. Res. Part B Methodol. 82, 36–53. https://doi.org/10.1016/J.TRB.2015.07.025

Tirachini, A., Gomez-Lobo, A., 2020. Does ride-hailing increase or decrease vehicle kilometers traveled (VKT)? A simulation approach for Santiago de Chile. Int. J. Sustain. Transp. 14, 187–204. https://doi.org/10.1080/15568318.2018.1539146

Turkeš, R., Sörensen, K., Hvattum, L.M., 2021. Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search. Eur. J. Oper. Res. 292, 423–442. https://doi.org/10.1016/J.EJOR.2020.10.045

Vallera, A.M., Nunes, P.M., Brito, M.C., 2021. Why we need battery swapping technology. Energy Policy 157, 112481. https://doi.org/10.1016/J.ENPOL.2021.112481

Vansteenwegen, P., Melis, L., Aktaş, D., Montenegro, B.D.G., Sartori Vieira, F., Sörensen, K., 2022. A survey on demand-responsive public bus systems. Transp. Res. Part C Emerg. Technol. 137, 103573. https://doi.org/10.1016/J.TRC.2022.103573

Wang, Y., Bi, J., Guan, W., Zhao, X., 2018. Optimising route choices for the travelling and charging of battery electric vehicles by considering multiple objectives. Transp. Res. Part D Transp. Environ. 64, 246–261. https://doi.org/10.1016/j.trd.2017.08.022

Wilson, N.H.M., Weissberg, R.W., Hauser, J.R., 1976. Advanced Dial-a-ride Algorithms Research Project: Final Report, Research report. Massachusetts Institute of Technology, Department of Materials Science and Engineering.

Windras Mara, S.T., Norcahyo, R., Jodiawan, P., Lusiantoro, L., Rifai, A.P., 2022. A survey of adaptive large neighborhood search algorithms and applications. Comput. Oper. Res. 146, 105903. https://doi.org/10.1016/J.COR.2022.105903

Zha, L., Yin, Y., Du, Y., 2018. Surge pricing and labor supply in the ride-sourcing market. Transp. Res. Part B Methodol. 117, 708–722. https://doi.org/10.1016/J.TRB.2017.09.010