Katholische Universität
Eichstätt-Ingolstadt

Fair and efficient urban freight management: Mathematical models, algorithms and machine learning approaches

Johannes Gückel

Kumulative Dissertation
zur Erlangung des akademischen Grades
Doctor rerum politicarum (Dr. rer. pol.)

Januar 2026

Tag der mündlichen Prüfung: 28. April 2026
Referent: Prof. Dr. Pirmin Fontaine
Korreferent: Prof. Dr. Heinrich Kuhn

Danksagung

Die vorliegende Dissertation ist während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Logistik und Operations Analytics an der Katholischen Universität Eichstätt-Ingolstadt entstanden. Diese Jahre waren für mich eine unvergessliche und sehr bereichernde Zeit, für die ich vielen Menschen zu Dank verpflichtet bin.

Mein besonderer Dank gilt in erster Linie meinem Doktorvater Prof. Dr. Pirmin Fontaine. Seine exzellente Betreuung, sein Vertrauen, die gewährte wissenschaftliche Freiheit und seine kontinuierliche Unterstützung haben diese Arbeit maßgeblich ermöglicht. Sein kritisches, zugleich stets konstruktives und äußerst schnelles Feedback hat in wesentlichem Maße zum Gelingen meiner Promotion beigetragen.

Ebenso danke ich meinen Kolleginnen und Kollegen, die durch gemeinsame Kaffeepausen, Konferenzbesuche, Forschungsworkshops, private Treffen und viele wertvolle Anregungen zum Erfolg dieser Arbeit beigetragen haben: Ayşe Nur Bakla, Büşra Nur Dağ, Dr. Paul Geuchen, Moritz Hundhammer, Merve Ibis, Dennis John, Simon Mader, Dr. Daniel Müllerklein, Philipp Parche, Tobias Potoczki, Dr. Alexander Rave, Dr. Stefan Voigt und Dr. Sandra Zajac.

Außerdem bin ich Birgit Jürgens sehr dankbar für ihre Unterstützung bei zahlreichen organisatorischen Fragen sowie für viele nette und aufmunternde Gespräche. Mein Dank gilt zudem meinem langjährigen studentischen Mitarbeiter Vincent Bläske für seine hervorragende Unterstützung in einer Vielzahl von Projekten.

Prof. Dr. Heinrich Kuhn danke ich für die Übernahme des Zweitgutachtens, sein hilfreiches Feedback in Forschungsworkshops sowie die sehr angenehme und kollegiale Arbeitsatmosphäre am Lehrstuhl. Zudem danke ich ihm für sein beeindruckendes Engagement und seine hervorragende Lehre, die meine Zeit als Masterstudent maßgeblich geprägt haben und ein wesentlicher Grund dafür waren, hier an der Universität mit einer Promotion weiterzumachen.

Mein Dank gilt außerdem meiner Freundin Susanna, die mich insbesondere in der Endphase meiner Promotion begleitet und diese Zeit in besonderer Weise bereichert hat.

Zu guter Letzt gilt mein Dank meinen Eltern Friedemann und Veronika. Ohne die unbeschwerte Kindheit, die ich erfahren durfte, wäre eine Promotion nicht möglich gewesen.

Johannes Gückel

Abstract

This cumulative dissertation consists of four core contributions, each of which has been published or accepted for publication. In these contributions, various mathematical models, algorithms, and machine learning approaches are employed to address problems related to the fair and efficient design of urban logistics systems. The following four listed papers represent the core contributions of this dissertation.

1. Gückel, J., Crainic, T.G., Fontaine, P., 2025. A two-step large neighborhood search for a collaborative two-tier city logistics system. Published in European Journal of Operational Research (VHB4: A).
2. Gückel, J., Crainic, T.G., Fontaine, P., 2025. Tactical planning in cooperative two-tier city logistics systems with fairness constraints. Published in Sustainability Analytics and Modeling 5 (VHB4: B).
3. Gückel, J., Fontaine, P., 2025. Fast Shapley value approximation through machine learning with application in routing problems. Published in Networks 86 (4) (VHB4: A).
4. Gückel, J., 2025. Approximating Shapley values with subcoalition Shapley values in routing problems. Accepted for publication in Transportation Research Procedia (VHB4: C).

   The published or submitted versions of the papers may differ slightly from the versions included in this dissertation (e.g., formatting of tables). Contents remain unaffected, with the exception of further revisions during the review process.

The first contribution addresses collaboration among logistics service providers in two-tier city logistics systems. It examines how to optimally employ available resources and how to allocate total system costs to individual logistics service providers. Methodologically, the problem is formulated as a mixed-integer linear program that combines a service network design problem for the first tier with a vehicle routing problem for the second tier. Since the model can solve only small instances optimally, a tailored two-step large neighborhood search with adaptive components is developed for larger instances. Numerical experiments show that the metaheuristic outperforms the solver Gurobi for larger instances and that collaboration yields significant cost reductions that primarily stems from first-tier collaboration. Various cost allocation approaches, ranging from game-theoretic methods to proportional rules, are also analyzed. The results show that simple allocation rules can violate basic fairness criteria and that logistics service providers of different sizes have diverging interests in the selection of a cost allocation method.

The second contribution builds on a similar setting but places greater emphasis on cooperative rules by introducing a service network design formulation with fairness constraints addressing workload distribution, cost sharing, and service regularity. A multi-day scenario is analyzed to assess whether fairness constraints should be enforced on average over the planning horizon or on a daily basis. Numerical results indicate that enforcing strict day-by-day fairness requirements leads to a substantial increase in total system costs, both monetary and environmental. In contrast, imposing

fairness constraints over longer planning horizons proves to be more cost-effective and environmentally sustainable through increased vehicle utilization and improved consolidation, thereby offering practical guidance for the design of cooperative city logistics systems.

The third contribution shifts the focus to allocating routing costs to individual customers. While the Shapley value provides a fair allocation method, it is computationally demanding for larger instances. To address this, a machine learning-based approximation method is developed that exploits structural properties of routing problems through problem-specific features. Extensive experiments show that the approach outperforms benchmarks and is transferable to other allocation contexts, such as bin packing. Scalability is demonstrated by showing that the method remains effective when trained on biased labels, which can be generated much faster for large instances with only a minor loss in approximation quality.

The fourth contribution also addresses the approximation of Shapley values but proposes a problem-independent method. Instead of exploiting structural properties, exact Shapley values are computed only for smaller subcoalitions of fixed size, and an approximation for the full coalition is obtained through weighted averaging. This leads to substantial computational savings for NP-hard problems such as routing. A numerical study shows that the method achieves a favorable balance between runtime and approximation quality and provides a broadly applicable alternative to existing approaches.

Introduction

The fast and reliable transportation of goods is an essential component of modern societies and a key driver of economic trade and growth. At the same time, the negative impacts of freight transport in urban environments pose a significant challenge for both logistics service providers and municipalities. These impacts include, in particular, emissions, pollution, noise, and land and resource consumption, and are expected to increase further due to ongoing urbanization and population growth. In response to these challenges, the European Commission emphasizes the need for a more sustainable and efficient organization of urban freight transport (European Commission 2021). The goal of city logistics, therefore, is to efficiently manage the transportation of goods while limiting the negative consequences of urban freight transport (Savelsbergh and Van Woensel 2016).

To achieve this goal, concepts are required that, on the one hand, reduce the negative impacts of urban freight transport and, on the other hand, accommodate growing demand. As city logistics systems typically involve multiple actors within an urban area, a central challenge lies in the design of such multi-actor systems and in the allocation of the resulting total system costs among them (Guajardo and Rönnqvist 2016). For the long-term success of collaborative urban logistics initiatives, it is therefore essential not only to investigate how these systems can be designed efficiently, but also which cost allocation mechanisms or decision rules can be applied to ensure fairness among the participating actors and to support stable collaboration.

The fair and efficient design of urban logistics systems is therefore more important than ever. This dissertation addresses this challenge from multiple perspectives. Contributions 1 and 2 examine fair and efficient collaboration among logistics service providers in two-tier city logistics systems (Crainic et al. 2004), while Contributions 3 and 4 focus on the fair allocation of transportation costs to customers, with particular emphasis on the Shapley value (Shapley 1953). Together, these contributions provide a comprehensive framework for the fair and efficient design of urban logistics systems.

Methodologically, the dissertation draws on various approaches. In Contributions 1 and 2, mathematical optimization models are formulated; additionally, in Contribution 1, an innovative two-step large neighborhood search metaheuristic with adaptive components is developed that can efficiently solve larger instances. Contribution 3 employs a machine learning-based algorithm to approximate Shapley values, while Contribution 4 proposes a generally applicable approximation method based on the weighted Shapley values for smaller subcoalitions.

Thus, this dissertation makes a significant contribution to current research both managerial and methodologically: managerial, because the fair and efficient design of modern city logistics concepts is of utmost relevance given current developments; methodologically, by developing and applying a diverse set of problem-tailored methodologies, ranging from mixed-integer linear programming and metaheuristics to machine learning–based regression approaches. The scientific relevance is also demonstrated by the fact that all core contributions of this dissertation have been accepted for publication in highly recognized academic journals.

The following section first provides an overview of the individual contributions before they are summarized in more detail.

Overview of Dissertation Contributions

This dissertation comprises four interrelated contributions that address fairness and efficiency in urban logistics at different decision levels.

Contributions 1 and 2 investigate cooperation mechanisms in two-tier city logistics systems, focusing on how cooperation, fairness constraints and cost allocation affect operational performance. These contributions primarily address system-level design questions and rely on optimization-based modeling and solution approaches.

Contributions 3 and 4 shift the perspective to the customer level and study the fair allocation of routing costs within vehicle routing problems. Both contributions build on cooperative game theory and focus on scalable approximations of the Shapley value to enable practical implementation in real-world settings.

Contribution 1 - A two-step large neighborhood search for a collaborative two-tier city logistics system

This contribution examines a two-tier city logistics system in which several logistics service providers collaborate. Freight is first delivered to large distribution centers at the city outskirts. From there, satellite depots are supplied using larger vehicles such as trucks or trams. Starting from the satellites, shipments are finally delivered to end customers using smaller and more environmentally friendly vehicles, such as cargo bikes. It is assumed that different service providers operate on both tiers and coordinate their activities. The collaboration occurs in two dimensions: first, shared infrastructures are used, such as satellite capacities or vehicle fleets. Second, logistics service providers can also serve each other's customers. Thus, the collaboration involves both shared resource utilization and shared customer service.

The resulting planning problem is formulated as a mixed-integer linear program. The first tier is modeled through a service network design formulation for the transport between city distribution centers and satellites, while the second tier addresses the routing problem for the final customer delivery between satellites and the customer's location. These two tiers are connected with each other through synchronization constraints that ensure synchronization by time and location. As this combined problem is highly complex and exact methods can only solve small instances optimally, a specialized two-step large neighborhood search is developed. This metaheuristic iteratively employs a large neighborhood search for the first tier and an adaptive large neighborhood search for the second tier. The metaheuristic explicitly leverages the two-tier structure through problem-specific operators, adaptive thresholds, and a solution memory, which together significantly improve scalability and allow the efficient solution of larger instances with higher demand and richer network configurations.

Extensive numerical experiments demonstrate that the proposed metaheuristic outperforms a state-of-the-art mixed-integer linear programming solver, and that its problem-specific components are crucial to achieving this performance. Moreover, a significant portion of the potential cost savings through collaboration is already realized through (partial) collaboration on the first tier.

Various mechanisms for distributing total costs among cooperating service providers are presented, ranging from proportional allocation methods to more advanced game-theoretical approaches. The results indicate that logistics service providers of different sizes exhibit distinct preferences with respect to cost allocation methods. While larger logistics service providers tend to favor allocation methods that ensure equitable relative cost savings, smaller logistics service providers prefer methods that emphasize marginal cost contributions when joining a coalition. Moreover, the findings suggest that simple proportional allocation rules may undermine the stability of collaborative arrangements, as they can lead to incentives for individual participants to deviate from the coalition.

Contribution 2 - Tactical planning in cooperative two-tier city logistics systems with fairness constraints

The second contribution addresses a similar problem setting as Contribution 1, but with a stronger focus on collaboration rules and their effects on coalition outcomes. Again, a two-tier city logistics system with multiple cooperating logistics service providers is considered. Unlike in Contribution 1, this study examines a multi-day planning horizon, allowing cooperative effects over time to be captured. Additionally, multiple fairness rules are introduced. Workload fairness ensures that each provider serves a certain proportion of its own customers and that total workload is balanced among providers. Cost fairness rules ensure that each provider experiences similar relative cost savings.

The problem is modeled as a mixed-integer linear program. The first tier is represented through a service network design formulation. The second-tier costs are included through an approximated cost for assigning customers to satellites rather than explicit routing decisions.

A comprehensive numerical study provides several managerially relevant findings. First, the study demonstrates a substantial potential for collaboration to improve performance across multiple monetary and environmental metrics. Then, the influence of fairness constraints and their intensities on coalition cost is analyzed. The results further show that enforcing consistent daily operation of the same transport services leads to significant cost increases, particularly under fluctuating customer demand across days. The study highlights that both cost-saving fairness and workload fairness are necessary for stable and equitable collaboration.

Contribution 3 - Fast Shapley value approximation through machine learning with application in routing problems

The third contribution changes the perspective and addresses how routing costs can be fairly allocated to customers. Among many allocation methods, the Shapley value originating from cooperative game theory literature is one of the most prominent. However, computing the Shapley value for routing problems is computationally very expensive, making it impractical for larger instances.

This contribution proposes a new approximation method based on supervised machine learning techniques. Machine learning models are trained on a large dataset of already computed Shapley values and learn problem-specific features that influence these values in order to predict Shapley values for unseen instances. As the total routing cost of an instance is known, the predictions are subsequently scaled so that their sum matches the total instance cost. For larger instances, where exact Shapley values cannot be computed in reasonable time, it is further proposed to use approximated Shapley values as labels for training the machine learning models.

Numerical experiments show that the approach outperforms existing approximation methods for the Traveling Salesman Problem (TSP) and performs well for the Capacitated Vehicle Routing Problem (CVRP). Using biased labels, which are approximate Shapley value estimates obtained through computationally efficient procedures such as Monte Carlo sampling, reduces dataset construction time dramatically while only marginally reducing approximation performance. The method also generalizes beyond routing, as demonstrated by applying the same methodology to a bin packing problem with size and weight constraints, with the only modification being the use of problem-specific features for bin-packing. Notably, this topic is of increasing relevance, as allocation mechanisms can be applied not only to monetary costs but also to CO_2 emissions, which are expected to play an increasingly important role in future decision-making.

Contribution 4 - Approximating Shapley values with subcoalition Shapley values in routing problems

The fourth contribution also addresses Shapley value approximation but proposes a fundamentally different idea: approximating Shapley values for large instances using exact Shapley values from smaller subcoalitions weighted by the routing cost of the respective subcoalition.

The proposed method, called β-Subcoalition Shapley Approximator (β-SSA), computes exact Shapley values for all subcoalitions of size β. For each customer, the weighted contribution across subcoalitions is aggregated and scaled to approximate the overall Shapley value. The approximation quality is influenced by β: $\beta = 1$ corresponds to stand-alone costs, while $\beta = |N|$ yields exact Shapley values.

Numerical experiments for the TSP and CVRP show that β-SSA achieves an excellent trade-off between computation time and approximation performance compared to sampling-based methods. The approach is also generalizable to other NP-hard problems where smaller instances can be evaluated more efficiently than larger ones. It may also serve as a procedure for generating biased labels for the machine learning approach introduced in the third contribution.

Contents

List of Figures

List of Tables

Contribution 1 - A two-step large neighborhood search for a collaborative two-tier city logistics system

1 A two-step large neighborhood search for a collaborative two-tier city logistics system

Johannes Gückel, Teodor Gabriel Crainic, Pirmin Fontaine

Abstract The rapid transport of freight is an essential feature of modern societies and an enabling factor for economic trade and growth. Nevertheless, the negative impact of freight transportation in urban areas poses challenges for Logistics Service Providers (LSPs) as well as for municipalities. In this context, a centrally coordinated Two-Tier City Logistics System (2T-CLS), in which LSPs voluntarily agree to collaborate with each other, has the potential to reduce both economic and environmental impact costs. In order to plan such a system, it is important not only to make efficient use of the resources provided but also to have a mechanism that allocates the costs incurred to the individual LSPs. We introduce a mixed-integer linear program (MILP) formulation for the tactical planning of a 2T-CLS involving multiple LSPs that share their resources and customer demands. This MILP comprises a service network design formulation on the first tier and a vehicle routing problem formulation on the second tier, which are connected with each other. To address larger instances, we introduce an Integrative Two-Step Large Neighborhood Search with adaptive components that integrates first and second-tier decisions. In order not only to minimize the costs incurred but also to distribute them fairly, we investigate different problem-specific proportional methods, as well as more advanced game theoretical methods. Numerical experiments show that collaboration leads to average cost savings of 26.91%, which primarily stems from first-tier collaboration.

Published: European Journal of Operational Research
URL: `https://doi.org/10.1016/j.ejor.2025.08.016`

1.1 Introduction

The transportation of goods and people is an indispensable part of modern societies and economies. At the same time, transportation is a major disruptive factor, especially for urban life, due to congestion, noise, emissions, space consumption, and other negative external effects. According to a United Nations (2018) forecast, urbanization will increase from 55% in 2018 to 68% in 2050. Simultaneously, the increasing degree of digitalization is leading to a trend towards more online orders, which means that e-commerce volume will rise as well (Lone et al. 2021). These two developments result in an increasing number of goods and people that need to be transported within cities. Making cities sustainable requires urban logistics concepts that not only meet the increasing demand for goods and services efficiently but also prioritize environmental responsibility as a key aspect of sustainability. While the need for efficient and sustainable transport concepts for inner-city transportation is more important than ever, Logistics Service Providers (LSPs) struggle with low load factors, empty trips, long dwell times at loading and unloading points, and a large number of deliveries to individual customers (Cepolina and Farina 2015).

The goal of city logistics is to efficiently manage the transportation of goods while minimizing the negative consequences of transportation (Savelsbergh and Van Woensel 2016). Introduced by Crainic et al. (2004), Two-Tier City Logistics Systems (2T-CLSs) feature the preliminary delivery of freight to CDCs located on the outskirts of urban areas. From these CDCs, large urban vehicles deliver freight to satellites — small facilities located in the inner city. Proceeding from the satellites, the last-mile deliveries to customer locations are completed using vehicles, hereinafter referred to as city freighters, which are smaller, environmentally friendly, and cost-efficient. This two-tier structure leads to challenging optimization problems because of the appearance of NP-hard problems on both tiers that require coordination and synchronization.

Efficiently managing 2T-CLSs can be achieved either by a single LSP operating throughout a city, as is the case in Monaco, or through collaboration — i.e., resource and/or demand sharing among multiple LSPs (Fontaine et al. 2023). While the first option may be difficult to implement in many cities due to political barriers, collaboration often faces resistance from LSPs because they wish to retain direct contact with customers. This raises the question of whether collaboration on both tiers is necessary and what benefits it might offer. Although collaboration in vehicle routing in single-tier environments has been well-studied in the literature, collaboration within 2T-CLSs remains relatively unexplored.

Previous studies on the collaboration of LSPs have focused on single-tier environments, particularly in vehicle routing or similar settings, and consistently reported significant cost savings. However, these studies do not consider the interdependencies between tiers, where collaboration on one tier can influence operations and costs on the other.

This paper specifically examines collaboration among LSPs in the tactical planning of 2T-CLSs, allowing us to assess how collaboration on each tier contributes to overall cost savings. By doing so, we can analyze the interaction effects between tiers and evaluate the mutual impact of collaboration on both tiers.

In a survey, Cruijssen et al. (2007b) found that LSPs have a strong belief in the efficiency gains that collaboration can offer. Further, Cruijssen et al. (2007c) surveyed a considerable number of LSPs in Belgium, finding that although collaboration presents clear advantages, developing a justifiable cost-sharing plan poses a significant obstacle to LSPs' joint ventures.

To maintain the stability of collaboration among partners, the question of how to allocate the resulting total system cost to the participating LSPs in a way that is perceived as fair is essential. Such a mechanism provides each partner with a strong incentive to participate. In 2T-CLSs, cost structures are shaped by interdependent decisions on both tiers, such as service selection, routing efficiency, demand pooling, and temporal alignment of operations. Models that focus on one tier of the system while approximating the other can capture only part of the potential cost savings and do not allow for a comprehensive evaluation of their allocation. Furthermore, current studies do not provide insights into the impact of different cost allocation methods in city logistics systems, even though these systems are typically regarded as inherently collaborative. Our study is the first to analyze these aspects in a fully modeled two-tier setting, providing new insights into both the impact of collaboration on monetary and environmental outcomes and the design of fair allocation mechanisms in collaborative city logistics.

While there is increasing interest in collaborative two-tier city logistics, there is still insufficient insight into how resources and customer demands should be shared among LSPs to achieve efficient and equitable outcomes in tactical planning. Existing cooperative studies focus on vehicle routing problems rather than service network design models, and do not evaluate the effects of different cost allocation methods, even though service network design is commonly used for tactical planning in city logistics. Furthermore, models and solution approaches for collaborative two-tier city logistics remain limited, especially when it comes to providing insights into how resources should be deployed and how costs arising from the tactical plan should be allocated to individual LSPs. This paper fills this research gaps by making three key contributions.

1. We introduce an innovative Mixed-Integer Linear Program (MILP) formulation for tactical planning of a 2T-CLS with collaborating LSPs that combines and connects both first-tier multi-modal service network design with second-tier arc-based vehicle routing. This formulation includes the collaboration of LSPs on both tiers individually and regulates the collaboration intensity through constraints that limit the demand sharing among these LSPs.

2. We develop a problem-specific two-step large neighborhood search with adaptive components that leverages the two-tier problem structure through problem-specific operators as well as through thresholds and a solution memory. This method is designed to efficiently solve larger instances, characterized by increased demands, services, and network complexity with additional City Distribution Centers (CDCs) and satellites.

3. Through an extensive numerical study, we show the performance of our metaheuristic and, as the first to do so in this context, provide managerial insights into the environmental and monetary impacts of collaboration in 2T-CLSs. We also evaluate the effects of different proportional and game-theoretical cost allocation methods, particularly with respect to LSPs of different sizes, which is a question that naturally arises in collaborative planning. These

insights are highly relevant for both LSPs and municipalities pursuing collaborative city logistics systems.

The paper is organized as follows: After the literature review in Section 1.2, we detail the problem setting in Section 1.3. We present the MILP for resource planning in Section 1.4. Section 1.5 details our metaheuristic. Section 1.6 introduces problem-specific proportional methods and shows game theoretical cost allocation methods. In Section 1.7, we conduct an extensive numerical study regarding the performance of the metaheuristic as well as cost benefits and cost allocations. The paper concludes with a summary and future research directions in Section 1.8.

1.2 Literature review

The literature review is divided into four sections. Section 1.2.1 presents literature related to 2T-CLSs. Section 1.2.2 focuses on collaboration in transportation, while Section 1.2.3 reviews the literature on cost allocation methods. Finally, Section 1.2.4 outlines the research gaps.

1.2.1 Two-tier city logistics

In their review on two-echelon vehicle routing problems, Sluijk et al. (2023) highlight that there are numerous publications with various model formulations and objective functions. They specifically identify city logistics as a key application area, where the two-tiered structure helps achieve demand consolidation and environmental benefits. They showed that there are essentially two ways of formulating such problems. An arc-based vehicle routing formulation (Jepsen et al. 2013) or a route-based formulation (Baldacci et al. 2013) in which there are sets for feasible routes. Our study aligns with the second formulation grounded in the specific literature on tactical planning of 2T-CLSs, a field primarily characterized by the use of service network design models, which are based on the pre-generation of services that represent, for example, a tram service traveling from a CDC to a sequence of satellites at specific times. Consideration of 2T-CLSs in the scientific community was introduced by the work of Crainic et al. (2004), which addressed the problem of the optimal location of satellites. A case study with data from the city of Rome clearly showed that a 2T-CLS can greatly reduce the distance traveled by large trucks within the city and, in return, smaller, more environmentally friendly vehicles can be used for the final delivery to the customers. The general framework for service network design models in the tactical planning problem in a 2T-CLS was introduced by Crainic et al. (2009). They took into account the time dimension of the demand and the associated need to synchronize and schedule the vehicles of the first- and second tier. In Crainic and Sgalambro (2014), a service network design formulation was proposed for the tactical planning problem in a 2T-CLS, including a discussion on algorithmic solution perspectives. Crainic et al. (2016) explicitly considered demand uncertainty in tactical planning. They proposed a two-stage stochastic programming formulation and presented different strategies for adjusting the plan to the observed demand. Crainic et al. (2021) proposed a scheduled service network design formulation as a modeling framework for tactical planning of a 2T-CLS and adapted it to the specific problem characteristics. Scherr et al. (2019) introduced a service network design formulation for the tactical planning of parcel delivery using mixed autonomous fleets, highlighting cost savings and coordination strategies based on infrastructure and demand. Fontaine et al. (2021) expanded the existing literature by introducing a more realistic problem setting, incorporating both inbound and outbound demand, a multimodal network with diverse vehicle types, and satellite capacity constraints tailored to demand volume and vehicle modes. They presented a scheduled service network design formulation and developed an efficient Benders decomposition algorithm to solve it. In their investigation, they demonstrated that a multi-modal fleet on the first tier of a two-tier system can reduce costs and improve utilization. Considering the decisions about the number and location of facilities on both tiers, Winkenbach et al. (2016) presented a MILP for the two-echelon capacitated location routing problem. Schmidt et al. (2022) investigated the integration of public transport service providers in a two-tier system to deliver freight from the outskirts to satellite depots. Further, Fontaine et al. (2023) recently analyzed when single-tier or two-tier systems benefit cities, finding that higher customer density and greater distances between depots and distribution areas favor two-tier strategies. They also showed that consolidation among LSPs reduces emissions. Related to the literature on 2T-CLSs, Perboli et al. (2021) addressed the joint problem of operating satellites and providing services to customers.

Overall, there are no models that combine service network design on the first tier with arc-based vehicle routing on the second tier and allow LSPs to collaborate with a certain intensity on both tiers. The only investigation explicitly considering different LSPs collaborating in a 2T-CLS was conducted by Crainic et al. (2020). They assumed that demand could be satisfied by any participant of the coalition of LSPs. As they pointed out, the collaboration leads to significant efficiency improvements in terms of monetary costs and environmental footprint. However, they did not consider the allocation of costs to the LSPs and only considered collaboration on the first tier of the system.

1.2.2 Collaboration in transportation

While collaboration in the maritime and airline industries has been used and explored for some time, it is still a growing area of research in urban freight management, taking on different forms. Vertical collaboration involves organizations at different levels of the supply chain, such as suppliers, retailers, and LSPs. On the other hand, horizontal collaboration refers to collaboration among organizations operating at the same level of the supply chain. Additionally, diagonal collaboration incorporates both horizontal and vertical collaboration (see Rusich et al. (2017) for a framework of collaborative logistics). In this work, we focus on horizontal collaboration. According to EU Commission (2001), horizontal collaboration refers to collaboration through an agreement or concerted practice between companies operating at the same level in the market that are potentially competing with each other.

The literature on horizontal collaboration largely consists of studies on the classification of its forms and empirical studies on its potential benefits and barriers. Cruijssen et al. (2007c) highlighted that horizontal collaboration can identify and exploit win-win situations between companies at the same supply chain level. This collaboration can take various forms. As Pérez-Bernabeu et al. (2015) noted, collaborating companies—whether competing or unrelated suppliers, manufacturers, retailers, receivers, or LSPs—can share information, facilities, or resources to reduce costs and/or improve service.

In the context of transportation logistics, horizontal collaboration has gained increased attention in recent years as an efficient instrument for the sustainable and efficient design of freight transport (Pan et al. 2019). For example, Agarwal and Ergun (2010) proposed a mechanism for directing liner shipping carriers to prioritize the collective interests of the alliance while optimizing their own profitability. Similarly, Nataraj et al. (2019) investigated the joint opening of a CDC and concluded that significant overall cost savings can be achieved with higher collaboration intensity. Regarding satellite depots, Bruni et al. (2024) explored the advantages of vertical and horizontal collaboration, identifying substantial cost reductions in both cases. For a comprehensive review of problems, approaches, and further research areas related to horizontal collaboration among LSPs, we refer to Pan et al. (2019).

Recent years have also seen numerous publications presenting the potential benefits of collaborative planning in single-tier settings, particularly regarding total profit improvements (e.g., Montoya-Torres et al. 2016). As highlighted by Gansterer and Hartl (2018) in their survey on collaborative vehicle routing, horizontal collaboration can lead to cost savings ranging from 20% to 30%. However, findings from single-tier settings cannot fully reflect the effects of collaboration across upstream or downstream tiers, nor determine which proportion of costs can be extracted from collaboration in each tier.

When constructing stable collaborations, ensuring the fair treatment of coalition members is crucial. Beyond the cost allocation methods considered in this study, some research has introduced constraints to address the workload of LSPs (e.g., Gansterer et al. 2018, Mancini et al. 2021). Additionally, auction models have been employed to manage demand exchange, where a shared pool allows participants to submit their demands, which are then traded among them through auctions (e.g., Elting et al. 2025, Gansterer et al. 2020a, Karels et al. 2020).

Finally, despite the growing body of literature in this area, Parisa et al. (2019) pointed out that the implementation of collaborative approaches in practice remains challenging due to the lack of a viable business model. This highlights the urgent need for a fair cost allocation mechanism to distribute the benefits of collaboration among participating LSPs.

1.2.3 Cost allocation methods

Cost allocation in collaborative games has already been studied in the literature for a variety of use cases. For a detailed review on cost allocation mechanisms in collaborative transportation, we refer to Guajardo and Rönnqvist (2016) in which over 40 different allocation mechanisms were identified in the reviewed literature. The mechanisms most commonly used in practice are based on proportional methods, where each participant is allocated a cost share in relation to a predefined reference value. In the simplest case, each participant bears an equal share of the costs. Other classic criteria are, for example, the share of demand or the stand-alone costs (Guajardo and Rönnqvist 2016).

Widely used methods based on cooperative game theory are the Shapley Value (Shapley 1953), the core allocation (Gillies 1959), and the nucleolus (Schmeidler 1969). In addition, cost allocation mechanisms were introduced in the context of specific applications, such as the Equal Profit Method (EPM) developed by Frisk et al. (2010) in the context of collaborative forest transportation. Further cost allocation mechanisms based on the distinction between separable and non-separable costs are introduced by Tijs and Driessen (1986).

Building on these approaches, recent studies have examined cost allocation methods in specific collaborative scenarios. For instance, Vanovermeire et al. (2014) compared different cost allocation methods in collaborative transport, especially with partners who have different characteristics. They showed that the choice of the appropriate cost allocation method depends heavily on the characteristics of the coalition. An example of cost allocation in horizontal carrier collaboration can be found in Verdonck et al. (2016), where they investigated the cooperative carrier facility location problem, comparing three different cost allocation mechanisms. Kimms and Kozeletskyi (2016a) determined the Shapley Value for the cooperative traveling salesman problem. Further, Kimms and Kozeletskyi (2016a) proposed an a priori core cost allocation for a horizontal cooperating traveling salesman, which provided expected costs for the coalition participants.

Nevertheless, neither proportional nor more advanced game-theoretical cost allocation methods have been investigated in the context of 2T-CLSs, highlighting a significant gap in the literature. Despite city logistics systems often being regarded as inherently collaborative, no concrete insights into cost allocation mechanisms have been developed for this domain, even though the consequences of different allocation mechanisms are highly relevant for the success of future collaborative city logistics projects.

1.2.4 Research gap

Based on our literature analysis, three key research gaps emerge. First, collaboration in 2T-CLSs remains largely unexplored, and little is known about the cost implications of (partial) collaboration at one or both tiers or how collaboration at one tier impacts cost and operations on the other tier, as single-tier collaborative models cannot reflect this. Second, the literature lacks efficient heuristic solution methods for multi-modal service network design at the first tier connected with vehicle routing at the second tier. Third, while cost allocation mechanisms have been studied in cooperative game theory, no studies provide insights into the potential collaborative savings in fully modeled 2T-CLSs and their implications for different cost allocation methods, including both proportional and game-theoretical approaches.

1.3 Problem setting

We describe the general functioning of 2T-CLSs in Section 1.3.1. Section 1.3.2 describes the tactical planning in 2T-CLSs. Section 1.3.3 outlines the collaborative aspects of the problem.

1.3.1 2T-CLS

For the illustration of 2T-CLSs, we use the terms introduced by Crainic et al. (2009). A 2T-CLS consists of two levels of physical infrastructure, the CDCs and the satellite platforms.

CDCs represent the first tier of facilities and are typically located on the outskirts of a city. Freight is first delivered from its origin location to the CDCs. For this, we assume costs for the delivery of each customer demand to each CDC. At the CDCs, the loads are sorted and consolidated for further distribution to satellites via urban vehicle services. A service starts at a certain time period at a CDC, visits one or more satellites, and then returns to the same CDC. Multimodality is represented by different transport modes for the first tier, including different types of urban vehicles. As Fontaine et al. (2021), we differentiate between line-based modes (e.g., tram) that can only drive along predefined lines and free-roaming modes (e.g., truck). Satellites represent the second-tier facilities and are usually located near the city center. Here, the load is again consolidated before the final last-mile delivery to the demand location takes place, using city freighters. These city freighters start at a satellite, visit several demand locations and return back to the satellite. A major challenge in these systems is the synchronization between the first-tier operations and the second-tier operations.

1.3.2 Tactical planning in a 2T-CLS

According to Crainic (2000) and Crainic and Kim (2007), tactical planning aims to develop a transportation plan to facilitate efficient operations and resource utilization while satisfying the demand for transportation within the quality criteria publicized or agreed upon with the respective customers. The tactical plan is designed for a short time horizon, called schedule length, which corresponds to the length of regularity for parameter setting. It must be set up sometime before based on forecasted parameters. Further, we assume a known deterministic demand, each demand associated with a specific volume. Daily demand fluctuations and uncertainties are carried out on the operational level and are therefore out of the scope of our paper.

The tactical plan of a 2T-CLS determines which services to operate, how to allocate demands to those services and satellites, and addresses second-tier routing. The goal is to satisfy the regular demand most efficiently in terms of a cost-efficient and environmentally friendly use of resources, and at the same time to meet demand conditions such as release and due dates. Strategic decisions, such as the location of satellites, are assumed to be given and not addressed in tactical planning. A key challenge within these two-tier systems is to ensure that the tactical plan enables the synchronization of operations between the first and second tiers at the operational level. This synchronization involves precise coordination in terms of location and timing. Concretely, the satellite to which the urban vehicle delivers a demand determines the starting position for that demand on the second tier. Further, in terms of timing, the delivery to a satellite must precede the departure from that satellite of each demand.

1.3.3 Collaboration in a 2T-CLS

We assume that multiple LSPs operate within the city using the same CDCs and satellites. These LSPs form a coalition and collaborate on a voluntary basis. Each LSP has its own set of demands, each with a certain demand volume, which is brought into the coalition when they join. On the other side, each LSP has its own resources for transportation. These resources are first-tier services,

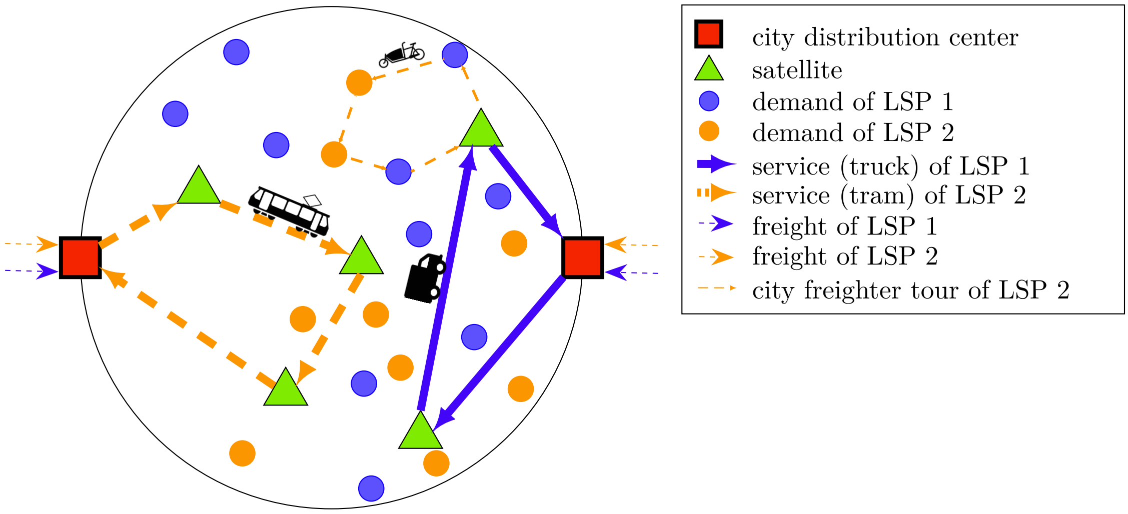

Figure 1.1: 2T-CLS with collaborating LSPs

capacities at satellites that they are allowed to use, as well as vehicle fleets. We assume that these LSPs agree to collaborate with each other through a central coordinator responsible for deciding which resources to use in order to fulfill all demands. The central coordinator can either be a third-party provider or a coordinating entity that is jointly operated by the LSPs involved. In either case, each LSP must be willing to share its information about demands and transportation resources.

We assume that the participating LSPs can offer transportation capacities on both the first tier and second tier. Thus, each LSP could satisfy its own demand. Therefore, collaboration can take place in two possible ways. First, by sharing their customer demands, so that each LSP can also have their demands delivered by services of other LSPs and vice versa. Second, by utilizing the same resources such as vehicle fleets, satellite depots, and CDCs. This enables freight consolidation of the participating LSPs by using common transport resources for first- and second-tier delivery. Figure 1.1 represents such a 2T-CLSs in which two LSPs collaborate with each other.

Within collaborations, it is important not only that the total costs resulting from the tactical planning of the provided resources are low but also that each individual LSP benefits from the collaboration and perceives the distribution of costs as fair. Therefore, a cost allocation method is required that takes the results of tactical planning as input and allocates the total system costs to the LSPs involved in order to enable fair and stable collaboration.

1.4 General approach and model

This section introduces a formulation that integrates first-tier multimodal service network design with second-tier vehicle routing, linking the two through connection constraints that align decisions by time periods and handover locations (satellites). We are the first to combine service network design and vehicle routing in this way, considering (partial) collaboration across both tiers with specific collaboration intensity constraints that limit the volume of shared demands on both tiers individually. First-tier vehicles handle larger volumes where collaboration mainly improves utilization, while on the second tier, cost savings largely result from more efficient route planning, which requires explicit vehicle routing modeling. Section 1.4.1 introduces the general modeling outline and notation. Building upon this, Section 1.4.2 presents our mathematical formulation for resource planning in a collaborative 2T-CLS.

1.4.1 General outline and notation

At the core of the problem is a set of LSPs $\mathcal{N}$. Each LSP $n \in \mathcal{N}$ has a subset of customer demands $\mathcal{D}(n)$ out of the set of all demands $\mathcal{D}$. Each demand $d \in \mathcal{D}$ is characterized by a volume v_d, the origin, the destination, the release date that specifies when the demand is available at the CDCs and a due date b_d that specifies when the demand must be delivered to its final location.

Consistent with Crainic et al. (2004), we model the first tier as a service network design. The time dimension is represented by a schedule length that is divided in the periods $1, ..., |\mathcal{P}|$. All considered time-related parameters are assumed to be integer multiples of the period length. Let $\mathcal{E}$ be the set of CDCs and $\mathcal{S}$ the set of satellites. For the first-tier deliveries, a set of services $\mathcal{R}$ is available, characterized by transportation mode m_r, vehicle type t_r and cost c_r, representing not only monetary but also environmental impact cost. Each service $r \in \mathcal{R}$ starts at a CDC $e_r \in \mathcal{E}$ and visits an ordered sequence of satellites. Transportation modes are represented by the set $\mathcal{M}$. Each mode of transportation is associated with different vehicle types (e.g., the mode "truck" can consist of the types "small truck" and "big truck"), where $\mathcal{T}$ represents the set of vehicle types. Each type t has a specific capacity u_t and a specific fleet size h_{etn} at CDC e provided by LSP n.

We consider three different capacity constraints on satellites. Operations at satellite $s \in \mathcal{S}$ are limited by a maximum number of urban vehicles a_{spn} LSP n can transfer in period p. The same type of constraint is additionally considered by $\overline{a}_{spmn}$ for each mode m to account for different capacity constraints per mode. In addition, we consider an upper limit for the total volume of freight g_{spn} LSP n can accommodate at satellite s in period p. The costs incurred to deliver demand d from its origin to a CDC e are also externally given and denoted as f_{de}. Note that these costs could further include reallocation costs of demands to CDCs of other LSPs if CDCs are not operated by all LSPs together.

To account for the different LSPs, each LSP n is able to provide a subset of services $\mathcal{R}(n) \subseteq \mathcal{R}$. Thereby $\mathcal{R}(n) \cap \mathcal{R}(\hat{n}) = \emptyset \quad \forall n, \hat{n} \in \mathcal{N} : \hat{n} \neq n$ holds. Further, each service has a certain service time w_r for unloading operations at each satellite, as well as an arrival time τ_{rs} at satellite s, respectively. $\mathcal{R}(p, s)$ and $\mathcal{R}(p, s, m)$ define the subsets of $\mathcal{R}$ that include all services (of mode m) that operate at satellite s during period p, while $\mathcal{R}(p, e, t)$ represents the subset that includes services of vehicle type t, starting at CDC e and operating in period p. We consider the subset $\mathcal{R}(d, s)$ that represents the services that fulfill the release date for demand d at the CDC the service starts from, and further include satellite s in their route.

We include the second-tier routing in our model rather than relying on approximations. This enables us to model collaboration effects on both tiers individually and to evaluate their respective cost impacts in a differentiated manner. To achieve this, we integrate a Vehicle Routing Problem with Release and Due Dates (VRPRDD) (Shelbourne et al. 2017) on each satellite, where the release dates of each demand are determined by the arrival period of the service that the demand is assigned to at the corresponding satellite resulting from the first-tier, plus a service time w_r. For the second-tier routing, we construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V} = \mathcal{D} \cup \mathcal{S} \cup \hat{\mathcal{S}}$ are the nodes consisting of the set of demand locations $\mathcal{D}$, satellite nodes $\mathcal{S}$ that serve as the starting points of the city freighters and a duplication of the satellite nodes $\hat{\mathcal{S}}$ that serve as the ending points of the city freighters. $\mathcal{K}$ represents the set of city freighters. Each city freighter k has a satellite node $s_k^+ \in \mathcal{S}$ at which it starts and a satellite node $s_k^- \in \hat{\mathcal{S}}$ at which it ends. These two satellite nodes are identical, i.e., they refer to the same physical location, as each city freighter starts and ends at the same satellite. The subset $\mathcal{K}(s)$ defines the city freighters starting from satellite s. The set $\mathcal{A}$ represents the arcs connecting the nodes. The subsets $\mathcal{A}(k)$ represent the set of arcs that city freighter k can travel, as each city freighter is only allowed to start and end at the satellite it is assigned to. We specify the allowable arcs for city freighter k to end in node i as $\delta_k^-(i)$, and the arcs starting from node i as $\delta_k^+(i)$. The costs that arise if arc $(i,j) \in \mathcal{A}$ is used by a city freighter are represented by $\hat{c}_{ij}$ and the time distance of each arc $(i,j) \in \mathcal{A}$ is represented by $\bar{t}_{ij}$. We assume service times l of city freighters at each node. Each city freighter has a capacity of q. Again, we account for different LSPs through subsets of city freighter $\mathcal{K}(n)$ provided by LSP n.

Additionally, we define two parameters, α_1 and α_2. These parameters specify for each LSP the lower bound of demand volume that must be assigned to own services on the first tier (α_1) and to own city freighters on the second tier (α_2), respectively. For example, $\alpha_1 = 0.2$ and $\alpha_2 = 0.3$ imply that at least 20% of an LSP's demand volume must be assigned to its own services and at least 30% of an LSP's demand volume must be fulfilled by its own city freighters. Therefore, these two parameters limit the demand sharing between the LSPs. A full table with the notation is provided in Table A.1 in Appendix A.1.

1.4.2 Mathematical problem formulation

In this section, we introduce the MILP for our problem setting in which a central coordinator minimizes the total system costs. After presenting the decision variables and the objective function, we start with the first-tier service network design constraints. Then, we describe the second-tier vehicle routing constraints. On both tiers, we introduce constraints that limit the sharing of demands according to the parameters α_1 and α_2. Further, we introduce linking constraints, which connect the first-tier with the second-tier and enable synchronization on the operational level.

We consider the following decision variables: the binary variable y_r takes the value one if service $r \in \mathcal{R}$ is selected, the binary variable x_{dsr} takes the value one if demand $d \in \mathcal{D}$ is assigned to satellite $s \in \mathcal{S}$ and service $r \in \mathcal{R}$ and the binary variable z_{ijk} takes the value one if second-tier city freighter k uses arc $(i,j) \in \mathcal{A}(k)$. Finally, p_{ik} is specifying the arrival time of city freighter k at vertex $i \in \mathcal{D} \cup \{s_k^+, s_k^-\}$.
Grounded on this, we formulate the following MILP:

Objective function

$$min \qquad \sum_{r\in\mathcal{R}} c_r \cdot y_r + \sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}} f_{de_r} \cdot x_{dsr} + \sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}(k)} \hat{c}_{ij} \cdot z_{ijk} \tag{1.1}$$

The objective function (1.1) minimizes the total cost incurred. These costs are made up of three components: 1. the costs related to the operation of services, 2. the costs for assigning demands to services and thus simultaneously to the CDCs from which the services start, and 3. the second-tier costs associated with routing the demands from the satellites to their final location. This objective function thus differs from other service network design models in the field of 2T-CLSs, in which the costs of the second tier are either linearly approximated or neglected at all (e.g., Fontaine et al. 2021, Scherr et al. 2019). Such linear approximations typically rely on the Euclidean distance between customer locations and satellites, offering only a rough estimate of actual routing costs. Moreover, they fail to capture the operational efficiencies enabled by cooperation, such as higher vehicle utilization, improved demand consolidation, and better temporal coordination. By explicitly modeling second-tier routing, our approach accounts for these effects and enables a more accurate assessment of collaboration benefits on each tier.

First-tier service network design constraints

$$\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}} x_{dsr} = 1 \qquad \forall d \in \mathcal{D} \tag{1.2}$$

$$\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}\setminus\mathcal{R}(d,s)} x_{dsr} = 0 \qquad \forall d \in \mathcal{D} \tag{1.3}$$

$$\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}} v_d \cdot x_{dsr} \leq u_{t_r} \cdot y_r \qquad \forall r \in \mathcal{R} \tag{1.4}$$

$$\sum_{r\in\mathcal{R}(p,e,t)} y_r \leq \sum_{n\in\mathcal{N}} h_{etn} \qquad \forall e \in \mathcal{E}, t \in \mathcal{T}, p \in \mathcal{P} \tag{1.5}$$

$$\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}(p,s)} v_d \cdot x_{dsr} \leq \sum_{n\in\mathcal{N}} g_{spn} \qquad \forall s \in \mathcal{S}, p \in \mathcal{P} \tag{1.6}$$

$$\sum_{r\in\mathcal{R}(p,s)} y_r \leq \sum_{n\in\mathcal{N}} a_{spn} \qquad \forall s \in \mathcal{S}, p \in \mathcal{P} \tag{1.7}$$

$$\sum_{r\in\mathcal{R}(p,s,m)} y_r \leq \sum_{n\in\mathcal{N}} \overline{a}_{spmn} \qquad \forall s \in \mathcal{S}, p \in \mathcal{P}, m \in \mathcal{M} \tag{1.8}$$

$$\alpha_1 \cdot \sum_{d\in\mathcal{D}(n)} v_d \leq \sum_{d\in\mathcal{D}(n)}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}(n)} x_{dsr} \cdot v_d \qquad \forall n \in \mathcal{N} \tag{1.9}$$

$$x_{dsr} \in \{0,1\} \qquad \forall d \in \mathcal{D}, s \in \mathcal{S}, r \in \mathcal{R} \tag{1.10}$$

$$y_r \in \{0,1\} \qquad \forall r \in \mathcal{R} \tag{1.11}$$

Constraints (1.2) and (1.3) guarantee that each demand is assigned to a single service for which the release date at the CDC is not violated. The capacity limit for each urban vehicle type of each service is ensured by Constraints (1.4). Constraints (1.5) limit the maximum number of used urban vehicle types t at CDC e for each period p. Constraints (1.6) limit the amount of freight operated in period p at satellite s. The number of urban vehicles operating in period p at satellite s is limited by Constraints (1.7) while Constraints (1.8) explicitly limit the number of urban vehicles of mode m. To limit the demand sharing, Constraints (1.9) enforce that each LSP satisfies at least a fraction α_1 of its own demand volume by its own services.

Second-tier vehicle routing constraints

For the resulting VRPRDD on the second tier, we use the previously defined graph $\mathcal{G}$.

$$\sum_{k\in\mathcal{K}}\sum_{j\in\delta_k^+(d)} z_{djk} = 1 \qquad \forall d\in\mathcal{D} \tag{1.12}$$

$$\sum_{j\in\delta_k^+(s_k^+)} z_{s_k^+jk} = 1 \qquad \forall k\in\mathcal{K} \tag{1.13}$$

$$\sum_{j\in\delta_k^-(d)} z_{jdk} - \sum_{j\in\delta_k^+(d)} z_{djk} = 0 \qquad \forall d\in\mathcal{D}, k\in\mathcal{K} \tag{1.14}$$

$$\sum_{i\in\delta_k^-(s_k^-)} z_{is_k^-k} = 1 \qquad \forall k\in\mathcal{K} \tag{1.15}$$

$$\sum_{d\in\mathcal{D}}\sum_{j\in\delta_k^+(d)} v_d\cdot z_{djk} \leq q \qquad \forall k\in\mathcal{K} \tag{1.16}$$

$$p_{ik} + \bar{t}_{ij} + l - p_{jk} \leq M_1\cdot(1 - z_{ijk}) \qquad \forall (i,j)\in\mathcal{A}(k), k\in\mathcal{K} \tag{1.17}$$

$$p_{dk} \leq b_d \qquad \forall k\in\mathcal{K}, d\in\mathcal{D} \tag{1.18}$$

$$\alpha_2\cdot\sum_{d\in\mathcal{D}(n)} v_d \leq \sum_{d\in\mathcal{D}(n)}\sum_{k\in\mathcal{K}(n)}\sum_{j\in\delta_k^+(d)} z_{djk}\cdot v_d \qquad \forall n\in\mathcal{N} \tag{1.19}$$

$$z_{ijk}\in\{0,1\} \qquad (i,j)\in\mathcal{A}(k), \forall k\in\mathcal{K} \tag{1.20}$$

Constraints (1.12) ensure that each demand location is visited by exactly one city freighter. Constraints (1.13) to (1.15) guarantee that each city freighter starts and ends at the respective satellite the city freighter belongs to while ensuring the correct flow. Constraints (1.16) limit the load of each city freighter to the maximum capacity. Constraints (1.17) ensure the correct time-flow of each city freighter route, while Constraints (1.18) guarantee that each demand location is delivered before its due date. Similar to Constraints (1.9), Constraints (1.19) limit the demand sharing on the second tier.

Connection constraints

We connect the two tiers with the following constraints:

$$\sum_{r\in\mathcal{R}} x_{dsr} = \sum_{k\in\mathcal{K}(s)}\sum_{j\in\delta_k^+(d)} z_{djk} \qquad \forall d\in\mathcal{D}, s\in\mathcal{S} \tag{1.21}$$

$$\sum_{r\in\mathcal{R}}\sum_{s\in\mathcal{S}}\tau_{rs}\cdot x_{dsr}+w_r\leq p_{s_k^+k}+M_2\cdot(1-\sum_{j\in\delta_k^+(d)}z_{djk})\ \ \forall d\in\mathcal{D},k\in\mathcal{K} \tag{1.22}$$

Constraints (1.21) state that each demand must be assigned to a city freighter that departs from the same satellite the demand is released to by the chosen first-tier service. Constraints (1.22) ensure that for each demand, the second-tier departure period at the satellite must be greater or equal to the first-tier arrival period plus service time. The incorporation of these constraints forms the basis for the synchronization of the two tiers at the operational level.

1.5 Solution approach

As our problem formulation combines two NP-hard problems, we propose a metaheuristic to address large instances. Therefore, we develop an Integrative Two-Step Large Neighborhood Search (I2S-LNS) with adaptive components that integrates decisions on both tiers. This design makes it possible to utilize the existing two-tier problem structure efficiently. We first describe the solution representation and search space in Section 1.5.1. Afterward, we illustrate the general framework of the metaheuristic in Section 1.5.2. In Section 1.5.3, we show how we construct an initial feasible solution. Sections 1.5.4 and 1.5.5 describe the two steps of the procedure.

1.5.1 Solution representation and search space

A solution $\mathcal{S}$ is represented by a set of selected services $\mathcal{R}_s$ (referred to as the service design), the assignment of demands to services and satellites $\mathcal{X}$, and the second-tier routing $\mathcal{Y}$ capturing the sequence of nodes including information on starting and departure times at each visited node. During the entire procedure, the search space is limited to the space of feasible solutions.

1.5.2 General outline of the I2S-LNS

The general procedure of the I2S-LNS is depicted in Algorithm 1. We start by constructing an initial feasible solution (Section 1.5.3). Then, our heuristic consists of iteratively applying a two-step procedure, where the First-Step Large Neighborhood Search (1S-LNS) applies operators on the services, followed by the Second-Step Adaptive Large Neighborhood Search (2S-ALNS), which applies operators on the demands. Further, we introduce a solution memory to intensify the search in promising search regions. This two-step design makes it possible to efficiently utilize problem specifics of both service network design on the first tier and vehicle routing on the second tier and to generate globally good solutions.

Precisely in the 1S-LNS, we destroy and repair our solution by applying operators on the service design $\mathcal{R}_s$ and adapting $\mathcal{X}$ and $\mathcal{Y}$ to ensure feasibility. Since it is unlikely to find new best solutions with a service design that has high service operating costs, denoted as $c^r(\mathcal{R}_s)$, we employ a threshold $(1+\theta_1)\cdot c^r(\mathcal{R}^*)$ where $c^r(\mathcal{R}^*)$ represents the cheapest service operating cost that are found so far during the search process. If the service operating cost of the current solution exceeds this threshold, we repeat Step 1. If this happens ϕ times in a row, we randomly select one of the stored solutions in our solution memory Γ and go back to this solution. This solution memory is constantly updated during the search process and includes the $|\Gamma|$ best solutions with distinct service design $\mathcal{R}_s$. It plays a crucial role in relocating the search process back to the regions that have proven to lead to high-quality solutions and thus helps to intensify the search in those regions. In the 2S-ALNS, we destroy and repair our solution by applying operators on the demands. Consequently, the demand assignment $\mathcal{X}$ and routing $\mathcal{Y}$ changes while keeping $\mathcal{R}_s$ fixed. In both steps, the best solution $\mathcal{S}^*$ is always updated when a new best solution is found. We stop after a minimum number of iterations without improving the objective value of the best found solution $f(\mathcal{S}^*)$ or after a time limit is reached.

Algorithm 1: I2S-LNS

$\mathcal{S}_c \leftarrow$ Generate a feasible initial solution; `// Section 1.5.3`
while *Stopping criterion not met* **do**
    $it \leftarrow 0$;
    repeat
        if $it > \phi$ **then**
            $\mathcal{S}_c \leftarrow$ drawRandomElement(Γ); `// Use solution memory`
            break;
        $\mathcal{S}_c \leftarrow$ 1S-LNS($\mathcal{S}_c$); `// Step 1: Section 1.5.4`
        $it \leftarrow it + 1$;
    until $c^r(\mathcal{R}_s) < (1 + \theta_1) \cdot c^r(\mathcal{R}^*)$;
    $\mathcal{S}_c \leftarrow$ 2S-ALNS($\mathcal{S}_c$); `// Step 2: Section 1.5.5`

Compared to standard large neighborhood searches from the literature, our approach integrates two large neighborhood searches within a single procedure. One large neighborhood search with operators related to the services and another adaptive large neighborhood search with operators related to the demands. These two are connected and interact with each other through an intelligent embedding of thresholds on both steps and a solution memory. Within these procedures, we have operators that exploit both service network design and vehicle routing problem specificities.

In contrast, most existing LNS methods, such as those in Breunig et al. (2016), Grangier et al. (2016), and Voigt et al. (2022), use a single-step procedure that applies all operators globally, even when targeting different aspects like satellite selection, routing, or location decisions. Hemmelmayr et al. (2012) distinguish between small- and large-impact operators in a two-echelon setting, but their approach still uses a unified LNS structure and applies large-impact changes only periodically after a certain number of iterations without improvement. So far, no existing method combines two coordinated LNS components that separately address service network design and vehicle routing, making our approach particularly well suited to the structure of two-tier city logistics systems.

1.5.3 Construction heuristic

We sequentially generate a feasible initial solution for the first tier and then for the second tier and ensure connection.

First tier: On the first tier, the solution consists of $\mathcal{R}_s$ and $\mathcal{X}$. In a first step, a service that complies with all the capacity restrictions (see Constraints (1.5) to (1.8)) is randomly selected. Afterward, demands that are not assigned to a service yet are randomly drawn one after the other and assigned to the selected service and assigned to the closest possible satellite the service visits, if constraints allow. We repeat this procedure until all demands are assigned. Assignments to satellites and services are also only permitted if it is ensured that there is still enough time for at least a commuting tour of a city freighter from the satellite to the final demand location to ensure feasibility on the second tier. This procedure is repeated η_1 times to generate η_1 different starting solutions. The solution with the lowest costs is taken as the initial solution for the first tier.

Second tier: Resulting from the first-tier assignments to satellites, we have a VRPRDD on each satellite. For each satellite, we greedily insert each demand in its cheapest possible position until all demands are served. This generates the initial solution for the routing $\mathcal{Y}$.

1.5.4 1S-LNS

We start by destroying the current solution S_c by applying removal operators on the service design $\mathcal{R}_s$. As the solution gets infeasible when removing services because previously assigned demands are getting unassigned, we try to reassign the unassigned demands to other services in the existing service design in their cheapest possible position (in terms of demand assignment to services and satellites as well as routing). If not all demands can be reassigned into the existing service design, we insert new services using the insertion operators. We repeat this until all demands are assigned to a service, resulting in a fully repaired solution. Algorithm 2 depicts the complete logic of the 1S-LNS.

Algorithm 2: 1S-LNS

$\mathcal{S}_c \leftarrow removalOperator(\mathcal{S}_c)$;
$\mathcal{D}_u \leftarrow$ Demands that have been assigned to removed services;
for $d \in \mathcal{D}_u$ **do**
  if $bestInsertion(d, \mathcal{S}_c)$ **then**
    $\mathcal{D}_u \leftarrow \mathcal{D}_u \setminus \{d\}$;
while $\mathcal{D}_u \neq \emptyset$ **do**
  $\mathcal{S}_c \leftarrow insertionOperator(\mathcal{S}_c)$;
  for $d \in \mathcal{D}_u$ **do**
    if $bestInsertion(d, \mathcal{S}_c)$ **then**
      $\mathcal{D}_u \leftarrow \mathcal{D}_u \setminus \{d\}$;
$\mathcal{S}^* \leftarrow update(\mathcal{S}^*, \mathcal{S}_c)$; $\Gamma \leftarrow update(\Gamma, \mathcal{S}_c)$; $\mathcal{R}^* \leftarrow update(\mathcal{R}^*, \mathcal{S}_c)$;

For removing services, we use the following operators:

- **Random removal**: We randomly select between 1 and $\lambda_{\mathcal{R}}$ services and remove them.
- **Worst removal**: For each currently active service, we determine the ratio between the demand volume that is currently assigned to the service and the service operating cost. Between 1 and $\lambda_{\mathcal{R}}$ services with the worst ratio are selected and closed.

To insert new services, we first determine all services that can still be inserted without violating any capacity constraints. Further, we determine if the service must be provided by a specific LSP to not violate the demand sharing Constraints (1.9). Then, we apply the following operators to this set of services.

- **Random insertion**: We randomly select a service and add it to the set of selected services $\mathcal{R}_s$.
- **Best-fit insertion**: We select the service with the best ratio of demand volume of the unassigned demands for which the service fulfills the release and the due date, and service operating cost c_r. In a preprocessing step before the heuristic starts, we determine for each demand and satellite combination the subset of services, that fulfill the release and due dates. Through this preprocessing step, we can easily evaluate this operator.

For continuous diversification purposes, we multiply the sorting criteria for the worst removal and the best-fit insertion operators by a random number in the interval [0.8, 1.2]. The selection of operators is based on the probabilities μ_r and μ_i, which determine the probability of utilizing the random operator for removal and insertion, respectively. The calibration of these probabilities is detailed in Section 1.7.2.1.

1.5.5 2S-ALNS

In this step, we iteratively destroy and repair the solution of Step 1 by applying operators on the demands. Throughout this step, the service design $\mathcal{R}_s$ is fixed. Only the demand assignment $\mathcal{X}$ and routing $\mathcal{Y}$ are modified. In each iteration, we first choose a demand removal operator to destroy our current solution. After that, we repair our solution by applying an insertion operator. We only update the current solution if the objective value is not increased by more than a factor of $(1+\theta_w)$. We again take advantage of a threshold to prevent wasting computation time in unpromising search regions. Specifically, we break the ALNS if the objective value of the current solution $f(\mathcal{S}_c)$ exceeds a dynamic threshold that starts in the first iteration with $(1+\theta_2)\cdot f(\mathcal{S}^*)$ and decreases linearly with the number of iterations until it reaches $f(\mathcal{S}^*)$. We employ this logic since strong improvements in the current solution can be expected, especially during early iterations. We set a maximum number of iterations η_2 as stopping criteria. Algorithm 3 depicts the complete logic of the second step ALNS.

Algorithm 3: 2S-ALNS

```
it ← 0;
while it < η2 do
    S'_c ← Repair(Destroy(S_c)) through applying operators on D;
    it ← it + 1;
    if f(S'_c) < (1 + θ_w) · f(S_c) then
        S_c ← S'_c;                                  // Update current solution
    S* ← update(S*, S_c);  Γ ← update(Γ, S_c);
    if f(S_c) > (1 + θ2 · (1 − it/η2)) · f(S*) then
        break;                                       // Break if threshold is exceeded
```

We consider the following problem-specific operators to remove demands from the solution:

- **Random removal**: We randomly draw between 1 and $\lambda_{\mathcal{D}}$ demands.

- **CDC regret removal**: For each demand d, we determine whether the assignment to another CDC e would lead to lower costs f_{de}. We then sort the demands according to the potential cost savings and take between 1 and $\lambda_{\mathcal{D}}$ demands with the highest potential cost savings when assigned to another CDC.
- **Satellite regret removal**: For each demand, we determine the deviation between the distance to the satellite the demand is assigned to and the distance to the closest other satellite. We sort the demands increasing according to this deviation and select between 1 and $\lambda_{\mathcal{D}}$ demands with the highest deviation. The idea behind this operator is that demands that are not assigned to their closest satellite are more likely to lead to cost savings.
- **Random route removal**: A city freighter route is randomly selected and all the demands that are on this city freighter are removed.
- **Worst route removal**: For each city freighter route, we determine the ratio between the routing costs and the demand volume that is fulfilled by this city freighter. All demands of the city freighter route with the worst ratio are removed.

For each of the operators that are not random, we multiply the sorting criteria by a random number in the interval [0.8, 1.2] for better continuous diversification.

We consider the following insertion operator

- **Best insert**: We insert the demand where the costs, in terms of service assignment cost as well as second-tier routing insertion cost, are lowest.
- **2-regret insert**: For each demand, we determine the cheapest insertion cost for assigning the demand to each service and satellite combination. The regret value is then defined by the difference between the cheapest and the second cheapest service / satellite combination. Then we sort the demands in decreasing order by their regret value and insert them one after the other in their cheapest possible position in terms of service assignment cost as well as second-tier routing cost.

If, for insertion purposes, a new city freighter must start, we assign the city freighter to the LSP of the respective demand.

Operator selection Operators are chosen based on a roulette wheel mechanism, as described by Pisinger and Ropke (2007), utilizing the reaction factor ρ. Each pairing of removal and insertion operators is assigned a reward σ, allocated as follows: σ_1 for worsening the current objective value, σ_2 for maintaining the current objective value, σ_3 for improving the current objective value. Unlike many studies, we do not assign a reward for finding a new best global solution because such outcomes are significantly influenced by the current service design $\mathcal{R}_s$, which is not impacted by these operators.

1.6 Cost allocation in a collaborative 2T-CLS

After minimizing the total system costs, the question now arises of how to allocate these costs to the participating LSPs. In this section, we present the different cost allocation methods that we analyze in our numerical study. As there are currently no insights into the consequences of different cost allocation methods in city logistics, we include some easy-to-communicate proportional methods (Section 1.6.1), as well as two very distinct game-theoretical cost allocation methods that represent different schools of thought: the Shapley value (Section 1.6.2) and the EPM (Section 1.6.3). While the Shapley value is based on marginality, the EPM is based on equality.

Given that each method captures different aspects of cost allocation, it is unclear how they will behave or whether they will lead to vastly different results. This is especially relevant in the context of 2T-CLSs, where a fully collaborative model captures cost savings across both tiers through pooling of demand, coordinated service selection, and temporal alignment. In such systems, especially smaller LSPs may benefit even more in relative terms, which underlines the need to investigate suitable cost allocation methods.

1.6.1 Proportional methods

The methods most commonly used in practice because of their simplicity are proportional methods. Proportional methods distribute costs in proportion to a predefined reference value (Vanovermeire et al. 2014). Although these methods are easy to calculate and interpret, they only take one aspect into account and leave out other key elements that influence the cost of each LSP.

In our setting, the number of demands assigned to each LSP (demand numbers) and the associated total volume of these demands are intuitive, problem-specific metrics for cost allocation. However, these proportional methods do not account for the cost structure of an LSP, including neither the stand-alone cost nor the marginal costs, which reflect the additional costs incurred when collaborating with others. To address this limitation, we propose a further proportional method based on the stand-alone cost. Specifically, we consider the following cost allocation approaches:

- **Demand-based Allocation (DA)**: The cost allocation to LSP n is proportional to the number of demands.

$$DA_n = \frac{|\mathcal{D}(n)|}{|\mathcal{D}|} \tag{1.23}$$

- **Volume-based Allocation (VA):** The cost allocation is proportional to the total volume of demand. In contrast to considering only the number of demands, this approach accounts for the cumulative demand volume assigned to each LSP, reflecting differences in demand magnitude.

$$VA_n = \frac{\sum\limits_{d \in \mathcal{D}(n)} v_d}{\sum\limits_{d \in \mathcal{D}} v_d} \tag{1.24}$$

- **Demand- and Volume-based Allocation (DVA):** The DVA is calculated as a weighted sum (weight ω) of the DA and VA where $0 \leq \omega \leq 1$.

$$DVA_n = \omega \cdot DA_n + (1 - \omega) \cdot VA_n \tag{1.25}$$

- **Stand-Alone-based Allocation (SAA)**: The cost allocation is determined by the ratio of the stand-alone costs of each LSP $C(\{n\})$ when no collaboration occurs and the cumulated stand-alone cost of all LSPs. In contrast to the other proportional methods, however, it is necessary to calculate the costs that incur for each LSP when acting independently.

$$SAA_n = \frac{C(\{n\})}{\sum\limits_{\hat{n} \in \mathcal{N}} C(\{\hat{n}\})} \tag{1.26}$$

Each of the four cost allocations is interpreted as a share of the total costs that LSP n must bear.

1.6.2 Shapley Value

Shapley (1953) introduced the Shapley Value as a cost allocation method in cooperative game theory. The Shapley Value is calculated as the average marginal contribution that the participant makes to each possible subcoalition when added to the coalition according to the order. It provides a unique cost allocation solution that satisfies a lot of fairness axioms. For details about the formulation of these axioms, we refer to Shapley (1953). Analytically, the Shapley Value for a participant n is calculated as follows:

$$SV_n = \sum_{\mathcal{O} \subset \mathcal{N}: n \in \mathcal{O}} \frac{(|\mathcal{O}|-1)!(|\mathcal{N}|-|\mathcal{O}|)!}{|\mathcal{N}|!} \cdot [C(\mathcal{O}) - C(\mathcal{O} - \{n\})] \tag{1.27}$$

Thereby, the summation is over all subcoalitions $\mathcal{O}$ within the set of all participants $\mathcal{N}$ that contain participant n whereby $[C(\mathcal{O}) - C(\mathcal{O} - \{n\})]$ represents the participants marginal contribution to the subcoalition $\mathcal{O}$. The Shapley Value is therefore a marginal cost-based method that does not emphasize an equal distribution of costs but derives its fairness aspect from marginal costs. In the context of city logistics, the marginal cost perspective is particularly relevant because it highlights how smaller LSPs with high stand-alone costs can benefit disproportionately within a coalition. By sharing resources, these smaller LSPs achieve significant gains in utilization and efficiency, whereas larger LSPs, already operating with higher utilization and greater efficiency, see comparatively smaller marginal improvements. The Shapley Value thus captures not only individual cost burdens but also the unique collaborative benefits that may not align with proportional allocations, making it particularly well-suited for evaluating cost-sharing among LSPs with diverse operational scales.

1.6.3 Equal Profit Method

The EPM was developed by Frisk et al. (2010) and is based on the idea that the maximum difference of the pairwise relative savings should be minimized. Thus, the relative savings of the participants should be as similar as possible. With decision variable c_n representing the costs allocated to participant n and $C(\{n\})$ representing the stand-alone costs of participant n the EPM allocates the costs through the following linear model:

$$\min \quad f \tag{1.28}$$

$$\text{s.t.} \quad f \geq \frac{c_n}{C(\{n\})} - \frac{c_{\hat{n}}}{C(\{\hat{n}\})} \qquad \forall n, \hat{n} \in \mathcal{N} : n \neq \hat{n} \tag{1.29}$$

$$\sum_{n \in \mathcal{O}} c_n \leq C(\mathcal{O}) \qquad \forall \mathcal{O} \subset \mathcal{N} \tag{1.30}$$

$$\sum_{n \in \mathcal{N}} c_n = C(\mathcal{N}) \tag{1.31}$$

The objective function (1.28) minimizes the maximum pairwise difference in relative savings that is measured by Constraints (1.29). Constraints (1.30) and (1.31) ensure the rationality and the efficiency condition. The constraints ensure a core allocation if a cost allocation that lies in the core exists. In the case that the core is empty, we use the epsilon-core as proposed by Frisk et al. (2010) to keep the coalition stable.

1.7 Numerical study

Section 1.7.1 describes the numerical setup and the generation of instances. Then, we calibrate the parameters and evaluate the performance of the I2S-LNS in Section 1.7.2. Section 1.7.3 presents managerial insights regarding the impact of collaboration as well as about the cost allocation methods.

1.7.1 Numerical setup and instance generation

We implemented the MILP in Python 3.12 using Gurobi 11 as a solver. The I2S-LNS is implemented in C++. All experiments are carried out on an AMD Ryzen 9 5950X 16-Core Processor, 3.40 GHz with 128 GB RAM. We generate new specific instances for this problem. Similar to related papers in the field of 2T-CLSs, we generate the instances based on a real city (Crainic et al. 2004, Fontaine et al. 2021). In our case, we take a large German city as a basis. We locate CDCs in easily accessible places outside the city. For satellites, we mainly use tram stops or larger squares in the city center. We go one step further than other papers and sample demand destinations for all instances in relation to publicly available data on population density in different city districts. We use two different networks, hereinafter called N1 and N2. N1 has two CDCs and four satellites, N2 has three CDCs and six satellites. We consider a planning horizon of 36 periods, ten minutes each. We generate a set of services with different numbers of satellites starting and ending at a CDC. The first start period is chosen randomly between periods five and ten. Each drawn service is then replicated two times during the planning horizon, with eight periods in between. We consider four different urban vehicle types: small tram, large tram, small truck, and large truck, with capacities of 500 for small vehicles and 750 for large vehicles. The cost of the services c_r depends on the vehicle type and the travel distance. We set the fixed cost for small vehicles to 15 and for large vehicles to 20. The variable costs are the travel distance multiplied by 1.2 for small trams, 1.5 for small trucks, 1.7 for large trams, and 2.0 for large trucks. The lower cost of trams is justified by their lower environmental impact cost. Further, the travel speed for trams is set to 25 km/h and for trucks to 20 km/h because of traffic. The capacity of city freighters on the second tier is 250, and the travel speed is 20 km/h. A service time of one period is assumed. In each instance, each

LSP individually draws random services out of the generated services, with half of these services being operated by large urban vehicles. Demand volumes are uniformly distributed between 50 and 100. For each demand, the release date RD is randomly selected from the range 1 to 18. The due date DD is calculated using the formula $DD = RD + 12 + \text{randint}(0, 6)$ to ensure time feasibility. For each demand / CDC combination, we assume random assignment cost (f_{de}) ranging from one to five. For capacity constraints, we assume that each LSP is allowed to operate one vehicle at a time (a_{spn}) at each satellite, implying $a_{spnm} \leq 1$ for all modes, and use 300 units of the satellite capacity (g_{spn}). With this setting, we generate instances of different sizes with up to three LSPs, 100 demands, and 60 services for our numerical experiments in the following sections.

1.7.2 I2S-LNS performance

We start with calibrating the parameter setting of the I2S-LNS (Section 1.7.2.1). No benchmark instances in the literature exist integrating service network design on the first tier and vehicle routing on the second tier. Further, comparing to pure service network design problems or pure two-echelon vehicle routing problems does not consider essential parts of our problem setting. Therefore, we evaluate the performance of the I2S-LNS through three steps: First, by comparing it against Gurobi on small instances (Section 1.7.2.2), second, by analyzing its performance on larger instances (Section 1.7.2.3) and benchmarking against a single-step version of our heuristic with the same parameters and a solution memory, lastly, by measuring the performance impact of removing newly introduced components of the I2S-LNS (Section 1.7.2.4). Throughout all performance experiments, we assume two LSPs that are fully allowed to share their customers ($\alpha_1 = \alpha_2 = 0$). Demands and services are evenly split across these two LSPs.

1.7.2.1 Calibration

To calibrate the parameter setting, we conducted tests on medium-sized instances with 40 and 60 demands. Initial values for each parameter were determined based on preliminary tests conducted throughout the development phase, ensuring a realistic starting point. Subsequently, we systematically varied each parameter within a specified range while holding all others constant to identify the setting that yielded the best average performance. The finalized parameter settings that are used throughout our experiments are presented in Table 1.1.

Table 1.1: I2S-LNS parameter setting

Parameter	Value	Description
η_1	100	Number of iterations of first-tier construction heuristic
η_2	$1.1 \cdot \lvert\mathcal{D}\rvert$	Number of iterations in Step 2
θ_1	0.25	Parameter for threshold in Step 1
θ_2	0.35	Parameter for threshold in Step 2
θ_W	0.01	Parameter for threshold for worsening the objective value
λ_D	6	Max. number of removed demands
$\lambda_{\mathcal{R}}$	3	Max. number of removed services
μ_r	0.2	Probability for random removal operator
μ_i	0.4	Probability for random insertion operator
ϕ	5	Max. number of threshold violations in Step 1
$\lvert\Gamma\rvert$	5	Length of solution memory
ρ	0.1	ALNS reaction factor
σ_1, σ_2, σ_3	{1,3,5}	ALNS scores for updating weights

1.7.2.2 Benchmark against Gurobi

We compare the performance of the I2S-LNS with the solutions obtained by solving the MILP using Gurobi. A time limit of one hour is imposed on Gurobi, whereas the I2S-LNS is set to terminate after 1.000 iterations (full iterations of Step 1 and Step 2) without improvement or once a time limit of ten minutes is reached. We execute the I2S-LNS five times for each instance. We conduct two experiments for this purpose. In the first experiment, we increase the number of demands ($|\mathcal{D}|$) while keeping the number of services constant ($|\mathcal{R}| = 24$). In the second experiment, the number of services ($|\mathcal{R}|$) is increased while the number of demands ($|\mathcal{D}| = 15$) is kept constant. To exclude the influence of the demands on the complexity, corresponding instances share identical demand sets. For example, the first instance with $|\mathcal{R}| = 12$ has the same demands as the first instance with $|\mathcal{R}| = 36$, and this pattern is maintained across all instances. For the first experiment, we use both networks. For the second experiment, we only use N2 to generate a wider range of services. The aggregated results for these experiments are shown in Table 1.2 and Table 1.3 (detailed results for all benchmarks are presented in Appendix A.2). We report the best and the average objective values as well as the percentage deviation (σ[%]) between the best and average. Further, we report the percentage difference between the objective value obtained by Gurobi and the average objective value of the I2S-LNS (Δ[%]) and the corresponding runtime.

Table 1.2: Aggregated comparison of Gurobi and I2S-LNS with varying $|D|$ and constant $|R| = 24$

		Gurobi			**I2S-LNS**				
Network	$\lvert\mathcal{D}\rvert$	**Costs**	**GAP [%]**	**Time [s]**	**Best costs**	**Avg. costs**	σ **[%]**	**Avg. time [s]**	Δ**[%]**
N1	5	108.85	0.0	2	108.85	108.85	0.0	17	0.0
N1	10	169.63	1.17	1731	169.63	169.63	0.0	37	0.0
N1	15	234.33	24.95	3600	226.21	226.55	0.14	107	-3.01
N1	20	285.50	24.22	3600	275.30	275.81	0.19	180	-3.29
N2	30	412.80	33.91	3600	371.58	373.68	0.55	405	-9.32
N2	40	-	-	3600	486.65	490.13	0.65	544	-

In the first experiment, we observe that Gurobi is only able to prove optimality for very small instances with $|\mathcal{D}| = 5$ and $|\mathcal{D}| = 10$. For larger instances, Gurobi exhibits large gaps of up to 33.91% on average for instances with $|\mathcal{D}| = 30$. In instances with $|\mathcal{D}| = 40$, Gurobi fails to obtain a feasible solution within the time limit. Our metaheuristic significantly outperforms Gurobi by finding exactly the same solutions for very small instances where Gurobi finds the optimal solution and better solutions for larger instances in less computation time. Notably, for larger instances, the performance of our metaheuristic remains very stable, with mean σ significantly below 1%.

Table 1.3: Aggregated comparison of Gurobi and I2S-LNS with varying $|R|$ and constant $|D| = 15$

	Gurobi			**I2S-LNS**				
$\lvert R\rvert$	**Costs**	**GAP [%]**	**Time [s]**	**Best costs**	**Avg. costs**	σ **[%]**	**Avg. time [s]**	Δ **[%]**
12	230.01	15.21	3600	226.62	226.62	0.00	93	-1.49
36	222.71	20.56	3600	215.45	215.46	0.00	104	-2.88
60	205.19	24.58	3600	198.87	198.91	0.02	109	-2.98

In the second experiment, we observe a slight increase in the gaps shown by Gurobi as the number of services rises. Notably, the difference between the I2S-LNS and Gurobi also increases slightly as the number of services increases. In all instances, the I2S-LNS delivers equal or better solutions than Gurobi in a much shorter computation time. Our metaheuristic also shows remarkable stability, with σ equals 0 in most instances, reflecting minimal variation between the best and average results. The computation time is only slightly influenced by the increased number of services. This analysis highlights the number of demands as the key complexity driver in our problem setting.

1.7.2.3 Performance on larger instances

In Table 1.4, we assess the heuristics stability on larger instances, creating five instances each for three configurations: $\mathcal{D} = 50, \mathcal{R} = 36$; $\mathcal{D} = 75, \mathcal{R} = 48$; and $\mathcal{D} = 100, \mathcal{R} = 60$. In addition, we benchmark against a single-step version of our metaheuristic with the same operators and a solution memory embedded. This version of the heuristic is explained in pseudocode in Appendix A.3. We set a time limit of 30 minutes, which is the only stopping criterion for the single-step version whereas the two-step version has the stopping criterion described above. Since defining a comparable dynamic stopping criterion for the single-step version could introduce bias, we fixed its runtime at 1800 seconds, matching the maximum runtime of the I2S-LNS. Thus, the comparison is made with the single-step version always running for 1800 seconds, while the I2S-LNS stops earlier if its dynamic stopping criterion is met.

Table 1.4: Aggregated performance benchmark against a single-step version on larger instances

$\|D\|$	$\|R\|$	I2S-LNS				Single-step version			
		Best costs	**Avg. costs**	σ [%]	**Avg. time** [s]	**Best costs**	**Avg. costs**	σ [%]	Δ [%]
50	36	587.16	590.48	0.57	1268	597.33	604.07	1.20	1.76
75	48	877.74	884.02	0.71	1778	893.89	905.20	1.02	1.75
100	60	1105.47	1113.75	0.75	1800	1126.56	1142.34	0.99	1.96

We observe that even for larger instances, our heuristic delivers very stable results with σ remaining below 1% in almost all of the instances. It also shows that the single-step version leads to significantly worse results with an average of 1.82% higher cost. This clearly shows that our two-step version makes better use of the existing problem structure than standard single-step large neighborhood searches.

1.7.2.4 Impact of special components

In this section, we assess the performance impact of special components of the I2S-LNS, specifically the thresholds and the solution memory. Table 1.5 presents the average percentage increase in the objective value when removing the special components. Note that when leaving out threshold θ_1, we also do not take advantage of the solution memory Γ. We use the previously introduced larger instances with 50 to 100 demands.

Table 1.5: Increase in objective value when taking out special components

$\|\mathcal{D}\|$	**Threshold** θ_1 **and memory** Γ	**Threshold** θ_2
50	2.66%	0.25%
75	2.65%	0.79%
100	3.17%	0.75%

As can be clearly seen, the inclusion of threshold θ_1 and, thus, the inclusion of the solution memory Γ dramatically boosts the performance of our heuristic among all instance sizes. Also, the inclusion of the threshold θ_2 significantly contributes to a better performance of our heuristic. These results show that the thresholds in this two-step procedure are enhancing the performance of the I2S-LNS.

1.7.3 Managerial insights

Firstly, we analyze the impact of varying the collaboration intensity through modifying the $\alpha-$parameters. Subsequently, we compare and analyze the cost allocation methods that were previously introduced. For these investigations, we generate ten instances each with 48 demands distributed across three LSPs, each offering 21 services. We take three cases into account, with each case being a modification of the previous.

- In **Case 1**, demands are evenly split among the LSPs, meaning each LSP has 16 demands.
- In **Case 2**, we adjust Case 1 by changing the allocation of demands to LSPs; one LSP receives eight demands, another 16, and the last one 24 demands.
- In **Case 3**, we further modify Case 2 by also varying the demand volume. The demand volume for the LSP with 16 demands is multiplied by 0.75, and for the LSP with eight demands by 0.5, to assess the effects of altering both the number and the volume of demands.

Thus, we have in Case 1 equally sized LSPs, in Case 2 LSPs that differ by the number of demands, and in Case 3 LSPs that differ by the number of demands as well as by the demand volume. For all experiments LSP 1 refers to the smallest, LSP 2 to the medium and LSP 3 to the largest LSP.

1.7.3.1 Impact of varying the collaboration intensity

In this section, we examine the effects of the collaboration intensity constraints. Starting from a fully collaborative system in which both resources and demands are fully shared ($\alpha_1 = \alpha_2 = 0$), we gradually increase the α parameters to 1 (no demand sharing). First, we increase α_1 and α_2 both at the same time (Figure 1.2). Then, we increase them individually while holding the other α parameter constant at 0 (Figure 1.3).

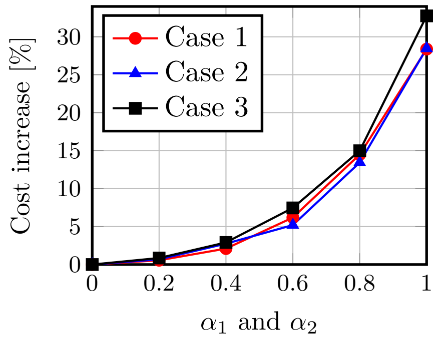

Figure 1.2: Sensitivity analysis for increasing α_1 and α_2

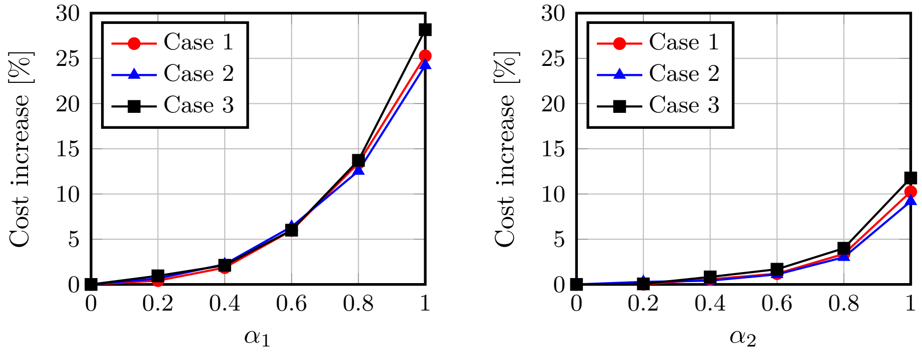

Figure 1.3: Sensitivity analysis for increasing α_1 and α_2 individualy

We observe that increasing the $\alpha-$parameters to 1 leads to significant increases in total system costs across all cases. On average, total cost increases by 29.25% when there is no demand sharing compared to total demand sharing. Additionally, we find that sharing even 20% of demands ($\alpha = 0.8$) can lead to a large portion of the potential cost savings achievable through full demand sharing. This is especially true for collaboration on the second tier, as nearly the whole increase in cost is happening for $\alpha_2 > 0.8$. This is attributed to the fact that LSPs use identical city freighters, eliminating structural cost differences among these LSPs. Further, we identify that sharing demands in the first tier leads to a higher cost impact than sharing demands in the second tier. This is due to the fact that first-tier collaboration impacts the assignment to CDCs, the utilization of the services, as well as the assignment to the satellites and thus the starting point for the second-tier routing. Moreover, the larger capacity of first-tier urban vehicles enhances the demand pooling potential across different LSPs. These effects result in a very interesting insight for LSPs, as a very large part of the potential cost savings is possible through collaboration without many of the customers even realizing this, as the final delivery to their homes is carried out by the LSPs they have engaged. This is demonstrated by the fact that almost the full amount of potential cost savings can be realized through full collaboration on the first tier and only the sharing of a small part of the demands on the second tier.

In addition to the cost increase compared to full collaboration, we further analyze the effects of demand sharing on several KPIs. These include the utilization of first-tier urban vehicles, the utilization of city freighters, the share of provided capacity by selected services operated by large vehicles (large trams and large trucks), and the cost increases at the individual tiers. The first-tier costs are calculated as $\sum_{r\in\mathcal{R}} c_r \cdot y_r + \sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} \sum_{r\in\mathcal{R}} f_{de_r} \cdot x_{dsr}$, while the second-tier costs are represented by $\sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}(k)} \hat{c}_{i,j} \cdot z_{ijk}$. The results are presented in Table 1.6.

We observe the following effects: (1) Collaboration, in general, increases vehicle utilization and leads to a higher share of larger urban vehicles being employed. (2) First-tier costs are almost exclusively influenced by collaboration at the first tier, while second-tier costs are affected by collaboration at both tiers and fall particularly sharply when collaboration takes place at both the first and second tiers. (3) Full collaboration leads to disproportionately high cost savings on the second tier. (4) Collaboration restricted to the first tier improves vehicle utilization at this tier

Table 1.6: KPIs for different collaboration scenarios

KPI	Full coll.	First tier coll. only	Second tier coll. only	No coll.
Util. first tier vehicles [%]	89.07	89.05	76.13	77.90
Util. second tier vehicles [%]	82.42	69.39	79.07	72.77
Share of large vehicles [%]	60.14	61.14	47.09	43.61
Cost increase [%]	-	10.40	25.89	29.87
Cost increase first tier [%]	-	1.44	25.06	24.25
Cost increase second tier [%]	-	24.16	28.20	40.19

Table 1.7: Average cost savings [%] compared to stand-alone cost

Case	LSP	SV	EPM	DA	VA	DVA	SAA
Case 1	1	26.82	26.32	24.59	25.87	25.23	26.32
	2	25.71	26.32	26.96	26.44	26.70	26.32
	3	26.54	26.32	26.98	26.34	26.66	26.32
Case 2	1	41.29	26.36	36.69	38.62	37.65	26.36
	2	25.24	26.36	24.36	24.07	24.21	26.36
	3	21.11	26.36	23.25	25.51	24.38	26.36
Case 3	1	47.31	28.08	33.59	61.77	47.68	28.08
	2	27.84	28.08	21.78	29.77	25.78	28.08
	3	21.41	28.08	29.51	14.82	22.17	28.08

but reduces it slightly on the second tier. This effect is due to pooling demands of different LSPs into a single service. Consequently, diverse demands from these LSPs reach the satellites leading to reduced utilization on the second tier as each LSP is required to operate its own city freighters independently. Nevertheless, even though the utilization is lower, it still leads to cost savings on the second tier through more efficient route planning through shorter traveled distances.

These results clearly highlight the interaction effects between the two tiers, where, for example, collaboration on the first tier also impacts operations on the second tier, demonstrating that demand sharing not only generates monetary savings for the LSPs, but also significantly reduces the environmental impact on each tier.

1.7.3.2 Cost allocation

In this section, we examine the effects of the different cost allocation methods. In particular, we analyze the consequences for LSPs of different sizes when applying the distinct methods, and to investigate whether there are differing interests in the selection of the cost allocation method depending on LSPs size. This is the first study that compares and evaluates different cost allocation methods in the context of city logistics, providing important insights for future collaborative projects. For this analysis, we evaluate the characteristic function of each instance by running the heuristic for each possible sub-coalition of LSPs. We then apply the previously introduced cost allocation methods to the individual instances. We use $\omega = 0.5$ as the weight for the DVA.

Our analysis reveals that by fully sharing resources and demands, the coalition could achieve cost savings of on average 26.91% by comparing the total stand-alone costs of all LSPs to the costs of the entire coalition across all cases. To compare the different cost allocation methods, Table 1.7 shows the average percentage cost savings of each LSP compared to its stand-alone cost using the cost allocation methods. Thereby, LSP 1 stands for the smallest, LSP 2 for the medium, and LSP 3 for the largest LSP.

Based on the results, we identify the following effects for the three cases:

- In Case 1, the average cost savings do not differ significantly neither between the LSPs nor between the individual cost allocation methods. This is not surprising as the LSPs do not differ significantly as they have exactly the same number of demands and services drawn from the same population. Differences at the level of the individual instances balance each other out on average.
- In Case 2, we find that LSP 1 is particularly favored by the use of the demand-based proportional methods and the Shapley Value, as the relative cost savings are significantly higher when using this method than when using the other methods. In contrast, it is better for the two larger LSPs to allocate costs using methods that generate similar relative cost savings. This is due to the fact that larger LSPs usually already have higher utilization of services because of their high number of demands. Demand-based proportional cost allocation methods do not take this effect into account, which leads to lower relative cost savings of larger LSPs compared to smaller LSPs when using these allocation methods.
- In Case 3, we observe similar effects. LSP 1 benefits in particular from the use of the Shapley Value as well as from the use of the VA, as this takes into account both the lower number of demands and the lower demand volume. LSP 3 again benefits, in particular, from the EPM and the SAA, but this time also from the DA, as this method does not take into account the higher demand volume compared to the other LSPs.

Notably, we identify that in every instance, the SAA and EPM deliver the same results. This is due to the fact, that allocating costs according to the SAA method already satisfies the additional constraints that are included in the EPM method.

Further, the experiments show that there are diverging interests of the LSPs with regard to the selection of a cost allocation method based on their size. Simple proportional methods may seem plausible and easy to calculate but can lead to very different relative cost savings among the LSPs. In particular, it would be difficult to agree on a fair reference value for the proportional methods, as this leads to immense differences in cost allocation. With respect to the two game theoretical allocation methods, it is particularly desirable for large LSPs to use the EPM or the SAA, as they lead to the same relative cost savings among the LSPs. For the smallest LSP, the allocation according to the Shapley Value is advantageous, as it takes into account the comparatively low marginal costs once the smallest LSP joins a coalition. These results show the diverging interests of the individual LSPs in relation to the selection of a fair allocation method.

Detailed analysis further reveals the following effects:

- In Case 3, the VA results in cost allocations that seem to be unfair for the two larger LSPs. Specifically, in three out of ten instances, the costs allocated to the two larger LSPs exceed the costs of the subcoalition of these two LSPs. In Case 2, we observe a similar effect in the allocation according to the DA. In several cases, this leads to an allocation where the two largest LSPs have almost no cost savings compared to forming a joint subcoalition. This suggests that simple proportional allocation methods can potentially lead to cost allocations that do not allow for stable coalitions, as some subcoalitions are more profitable than the coalition of all LSPs.

- Although the average cost allocations in Case 1 are quite similar, significant differences in individual instances exist, especially between the game theoretical methods and the proportional methods (except SAA). This discrepancy arises because the proportional methods do not account for variations in costs, which are influenced by the location of demands, the composition of the services, and the release and due dates for these demands.

Another important insight that we identify is that the smallest LSP (LSP 1) only incurs a very small marginal cost when added to the coalition of LSP 2 and LSP 3. Figure 1.4 displays both the stand-alone costs (SAC) of LSP 1 when acting independently and the marginal costs (MC) of adding LSP 1 to the coalition throughout the ten instances for Case 2 and 3. As shown, the marginal costs are by far lower compared to the stand-alone costs. Remarkably, in one instance in Case 3, the marginal costs are even negative. This is attributed to the fact that LSP 1 contributes only a few low-volume demands to the coalition but offers as much capacity at the satellites and provides as many services as the other LSPs. These counteracting cost effects result in very low marginal costs. This clearly indicates that incorporating smaller LSPs, which also have transport capacities, can be highly beneficial for the coalition overall.

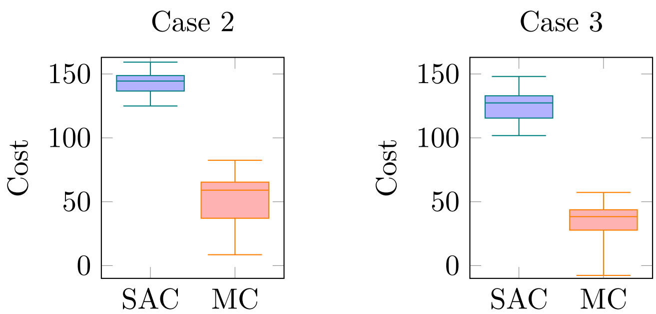

Figure 1.4: Distribution of stand-alone cost and marginal cost for LSP 1

1.8 Conclusion and further research

This study presented a decision model for collaboration in 2T-CLSs, which integrates a service network design on the first tier and a vehicle routing problem on the second tier. Additionally, we proposed an efficient problem-specific metaheuristic that is based on a two-step procedure consisting of a large neighborhood search for the service design and an adaptive large neighborhood search for the demand assignment and routing. Furthermore, various cost allocation methods were applied for this problem setting.

Through an extensive numerical study, we demonstrate the performance of our solution approach. Further, we could show that the collaboration among LSPs can result in significant cost savings. Even when only a small percentage of the demands is shared among the coalition, a large part of the potential cost savings through collaboration can be achieved. Further, collaboration contributes to increased vehicle utilization on both tiers and thereby contributes towards a more sustainable city logistics system. What is particularly important for LSPs is that we were able to show that almost all potential cost savings through full collaboration can be realized through full collaboration on the first tier and the sharing of only a small share of the demands on the second tier. Regarding the cost allocation methods, our experiments showed diverging interests of the LSPs. While for larger LSPs, a cost allocation based on methods that lead to similar relative cost savings is advantageous, for smaller LSPs, the Shapley Value is advantageous because it pays particular attention to the very low marginal costs. Simple proportional methods based on demand characteristics such as the number or the volume of demand also lead to very different cost savings for LSPs depending on their demand characteristic. Within a coalition, it may be difficult to agree on a simple proportional method, as depending on the choice of the reference value, some LSPs will be advantaged and others disadvantaged.

For future research, it could be exciting to investigate other aspects of collaboration, such as risk sharing, which could lead to the development of a more resilient collaborative city logistics system. With regard to the strategic level, the collaborative selection and location of sites for satellites and CDCs could offer scope for further research.

Acknowledgments

We thank the anonymous reviewers for their valuable recommendations, which have significantly improved our paper. The paper was written as part of a project funded by the German Research Foundation. We would like to express our gratitude to the German Research Foundation (project number: ROCOCO FO1468/1-1) for the financial support of our research project. Additionally, our gratitude goes to the Bavarian Research Alliance for their support of the research stay in Montreal, which contributed significantly to the development of this paper. While working on this paper, the second author was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréal, Canada. We gratefully acknowledge the financial support provided by the Canadian Natural Sciences and Engineering Research Council (NSERC) through its Discovery grant program, as well as by the Fonds de recherche du Québec through their Teams and CIRRELT infrastructure grants.

Contribution 2 - Tactical planning in cooperative two-tier city logistics systems with fairness constraints

2 Tactical planning in cooperative two-tier city logistics systems with fairness constraints

Johannes Gückel, Teodor Gabriel Crainic, Pirmin Fontaine

Abstract Two-tier city logistics systems (2T-CLSs) offer the potential for more efficient and environmentally friendly freight management. Since such systems require substantial infrastructure and multiple Logistics Service Providers (LSPs) operate within a city, cooperation among LSPs presents opportunities for cost and emissions reductions.

However, cooperation requires ensuring every LSP has an incentive to participate and feels fairly treated. To address this, we present a service network design formulation for multi-day tactical planning in a 2T-CLS with cooperating LSPs, incorporating fairness constraints on workload, costs, and service regularity.

Through a numerical study, we quantify the impact of fairness constraints, showing that overly strict constraints harm the coalition. Lower cost increases and greater environmental benefits occur when fairness is enforced over multiple days rather than daily. Further, we observe a 47% reduction in CO2 emissions through cooperation, providing valuable policy implications for LSPs and municipal authorities pursuing sustainable city logistics.

Published: Sustainability Analytics and Modeling 5

URL: `https://doi.org/10.1016/j.samod.2025.100046`

2.1 Introduction

Recent years have witnessed both rapid urbanization and increased e-commerce activity (Lone et al. 2021, United Nations 2018). These trends are expected to continue in the future, leading to an increased volume of freight delivery in cities. As a result, numerous LSPs traverse cities daily, delivering freight to customers. Despite its indispensable role in societies and economies, this practice contributes to traffic congestion, noise, pollution, and harmful emissions, negatively impacting the environment and public health (Savelsbergh and Van Woensel 2016). As highlighted by Raihan and Tuspekova (2022), urbanization is a major driver of greenhouse gas emissions. Therefore, the development of city logistics systems that promote efficient and environmentally friendly urban delivery solutions represents a key strategy in advancing global sustainability agendas. In particular, such systems can significantly contribute to achieving the United Nations Sustainable Development Goals, notably *Sustainable Cities and Communities* and *Climate Action* (United Nations 2015).

In recent years, numerous public sector initiatives have been put forth to improve urban freight management (Holguín-Veras et al. 2020). A key challenge is finding solutions that consider the interests of various stakeholders (Galambos et al. 2024). While LSPs and their customers are primarily concerned with minimizing transportation costs and ensuring timely deliveries, urban communities focus on reducing the negative external effects caused by transportation. One way of increasing monetary and ecological efficiency and thus bringing together the interests of different stakeholders is the concept of 2T-CLSs (Crainic et al. 2004). These systems involve large urban vehicle services to deliver freight from CDCs, located on the outskirts of a city, to satellite facilities located within the city. The final delivery to the customer locations is conducted with smaller, environmentally-friendly vehicles, referred to as city freighters, starting from the satellites. Our study focuses on tactical planning, which involves the medium-term design of the service network and schedule, the allocation of demand flows, and the management of required resources, with the goal of ensuring that the system operates efficiently and remains economically viable over time (Crainic and Laporte 1997).

As 2T-CLSs require extensive infrastructure and as several LSPs operate within a city, often without fully utilizing their resources, cooperation among them presents an opportunity to reduce both monetary costs for LSPs and their customers, as well as the negative environmental impact on the city and its residents. In such cooperative systems, each LSP contributes its own customer demands and transportation resources, which are jointly coordinated to minimize total system costs. Building on this perspective, Browne et al. (2012) have underscored the significant potential of cooperative delivery systems to alleviate the negative social and environmental consequences of urban freight transport. Therefore, we explicitly consider different LSPs with their own resources and their own demand that is allowed to be shared between each other by assigning demands of one LSP to other LSPs services. Although city logistics is seen as an integrated logistics system in which several different players operate within a city (Crainic et al. 2023), cooperative concepts and their fair design have so far received little attention in the literature. This fair design encompasses mainly two aspects. Firstly, there is a monetary aspect in which cost savings are fairly distributed among the coalition of LSPs. Secondly, it involves a workload consideration, ensuring that each

LSP primarily handles its own workload to maintain clear responsibility for its operations.

A further crucial aspect of fairness is the time horizon. Enforcing fairness in workload and cost distributions on a daily basis can often be highly inefficient compared to considering fairness in total over a longer period.

While existing models on 2T-CLS can, in principle, be applied to multi-day planning, they do not account for aspects that span across days, for instance, ensuring that cost or workload differences between cooperating partners remain balanced on average over multiple days. Most of these models are built around a single planning horizon composed of discrete time periods. Although this structure allows for multi-day planning, it fails to capture interdependencies between days that are crucial in (cooperative) city logistics. This also overlooks key challenges in city logistics, where certain urban vehicles, such as trams, may need to operate with consistent service patterns each day. The impact of such service regularity on cost dynamics remains largely unexplored in the literature. Moreover, a multi-day perspective is essential to understanding how fairness criteria can be balanced over time rather than rigidly applied to each individual day.

To summarize, there is a clear lack of insight into tactical planning in 2T-CLSs about the monetary and environmental impacts of cooperation, with focus on fairness for LSPs' cooperation, and the regularity of services over a multi-day period. Therefore, we aim to provide insights into how such a collaborative system can be operated, thereby supporting the achievement of the UN Sustainable Development (United Nations 2018), since clear rules of cooperation are essential for sustainably establishing collaboration in city logistics.

For LSPs as well as for public authorities, it is essential not only to have detailed planning models but also general policies that give them guidelines on how such a cooperative 2T-CLS must be designed so that it works in the long term and is successful for every actor involved. Our paper aims to derive general guidelines and policy implications for cooperation in 2T-CLSs in a more realistic multi-day planning environment through appropriate numerical experiments, which are of great interest for LSPs as well as for urban communities and policymakers. While the potential for cost savings through cooperation has already been investigated in various areas of transportation logistics, we focus on the following research questions:

- What is the impact of cooperation among LSPs on cost performance and environmental Key Performance Indicators (KPIs) that are relevant for different stakeholders within a city?
- How do different fairness constraints regarding the workload and the cost balance impact the system?
- What is the impact of the time horizon of fairness constraints? Should they be considered on a daily basis or on average over several days or the full planning horizon?
- What are the effects of service regularity?

We make five key contributions to existing literature by answering these research questions.

1. We introduce a new problem setting for fair tactical planning in a cooperative 2T-CLS explicitly considering multi-day periods.

2. We present a novel service network design formulation for a fair multi-day 2T-CLS with cooperating LSPs.

3. We formulate various fairness constraints regarding the workload and the costs of each LSP, as well as regarding the regularity of selecting services.

4. We conduct an extensive numerical study investigating the effects of cooperation, the impact of fairness constraints, and the consequences of service regularity based on several monetary and environmental criteria.

5. We derive valuable policy implications for the design of a sustainable cooperative city logistics system.

The paper unfolds as follows: Section 2.2 details the problem description, followed by a comprehensive literature review in Section 2.3. Section 2.4 introduces the service network design formulation. In Section 2.5, we analyze the benefits of cooperation, the impact of fairness constraints, and the consequences of service regularity. Based on the results, we then derive general policy implications in Section 2.6. Finally, Section 2.7 summarizes our study and gives an outlook on future research opportunities.

2.2 Problem description

In this section, we start with the general problem environment in 2T-CLSs in Section 2.2.1 before we focus on the cooperative and fairness aspects in Section 2.2.2.

2.2.1 Tactical planning in two-tier city logistics

For the illustration of 2T-CLSs, we use the terms introduced by Crainic et al. (2009). We do the tactical planning for a 2T-CLS with inbound demand over a schedule length of several days. Each demand has an origin location outside the city, a destination within the city, and a specific demand volume. Further, time restrictions for each demand are considered. Specifically, there is a release date for each demand, on which the demand is available at the earliest at the CDCs, and a due date, on which each demand should arrive at its destination at the latest. We consider a schedule length that is further differentiated into multiple days, where different demands arise on each individual day and must be fulfilled on the exact same day they occur. The release and due dates are, therefore, related to time periods within one day.

In 2T-CLSs, these demands are first delivered from their origins to a CDC that is chosen in the tactical plan. These CDCs are usually located on the outskirts of a city, in a place that is easily accessible for large vehicles such as trucks or trams. For this, we assume costs for the initial demand delivery to a CDC. Starting from the CDCs, the first-tier delivery is operated by services using large urban vehicles of different vehicle types, such as trams or trucks of various modifications. These vehicles differ in terms of their capacity and their fixed and variable monetary and environmental cost components. Each service starts at a specific time period at a particular CDC and visits an ordered sequence of satellites. These satellites are typically located within the city, around the city center, to ensure efficient second-tier routing operations. The demands arrive at these satellites and are consolidated and prepared for the final second-tier delivery. Please note that each satellite has limited space, so there are capacity limits on the number of vehicles and demand volume that can be handled simultaneously on a satellite. Thus, satellites represent the meeting point between first and second-tier vehicles. Smaller, environmentally friendly vehicles, like cargo bikes or electric vehicles, operate the delivery to the final demand destinations. These start at a specific time at a satellite and travel to a sequence of demand destinations. However, the focus of our study is on tactical planning. To this end, we follow an established approach in the literature and do not solve the actual routing problem on the second tier, but rely on approximations (Fontaine et al. 2021, Crainic et al. 2020). We aim to provide decision support for selecting services and allocating demands to services, satellites, and CDCs on the first tier on each day of the schedule length.

We assume that the demands for the entire schedule length of several days are known deterministically beforehand, with each day potentially having a different number of demands, different demand locations, and varying demand volumes. Any deviations between the forecasted demand and the demand actually occurring on the day are out of the scope of this study as different strategies have already been investigated (Crainic et al. 2016). In addition, we assume that the same set of potential services is provided by the LSPs on each day and can be selected. So, in contrast to the demands, where on every day different demands occur, the set of potential services stays the same for the entire schedule length, where each service can be selected on each day. The connection between the individual days is expressed through two aspects. Firstly, it may be desirable for services of the LSPs if at least some of their services operate regularly each day to plan in a more stable manner. Such policies are helpful both for services with trams, as they have to coordinate their schedules with passenger trams already running and should therefore not change their times every day, and for all services, as it enables more stable schedules and better planning for drivers and vehicles. The resulting impact on the whole system has not yet been considered in any study on 2T-CLSs. Further, the multi-day perspective also influences the cooperation of the LSPs, specified in the next section. Even if this is not explicitly the subject of our study, the multi-day planning described above could also be used for rolling horizon planning, where the rolling planning horizon consists of several days.

2.2.2 Cooperation in two-tier city logistics

We consider different LSPs that operate on the same infrastructure within this system. However, each LSP has different demands on each day. Further, each LSP provides a set of services where a subset should be selected and which is capable of satisfying its own demands. Therefore, we assume that each LSP contributes both demands and transportation resources to the coalition, with the transportation resources comprising different urban vehicle services and usable capacity at satellites.

Figure 2.1 illustrates such a cooperative 2T-CLS and Figure 2.2 explicitly illustrates the cooperative aspects through mutually sharing demands and resources. This cooperation allows LSPs to exchange demands, enabling one LSP to fulfill the demands of another using its own services. Additionally, each LSP contributes its transportation resources to the coalition, which include vehicle

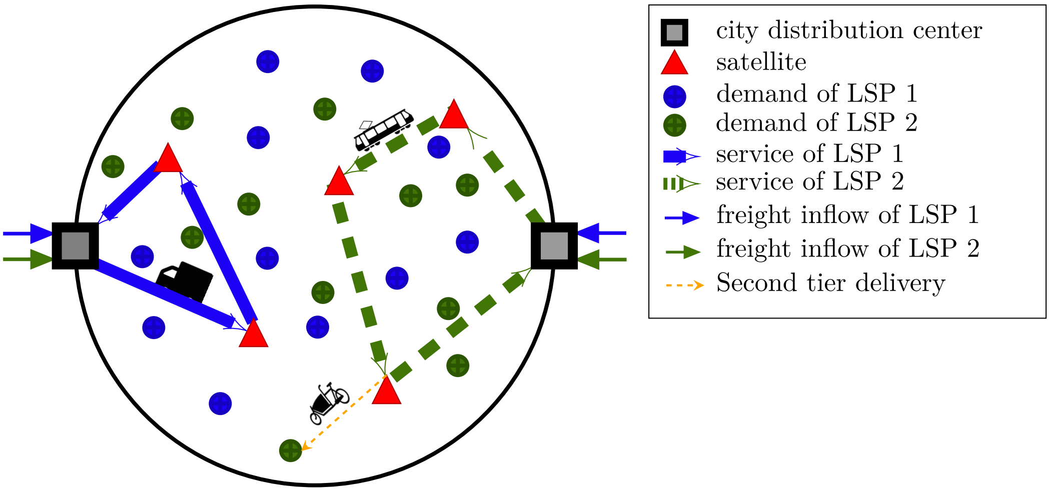

Figure 2.1: 2T-CLS with cooperating LSPs

Demands of different LSPs in urban vehicle service of LSP 1

Demands of different LSPs served via same satellite

Figure 2.2: Cooperative aspects

fleets, service capabilities, and satellite capacity.

We assume that each LSP itself is responsible for delivering its demands to a CDC, as well as subsequently delivering them from any satellite to their final demand locations. Cooperation thus takes place on the first tier through the joint provision of resources, as well as through demand sharing, which means that each LSP can also deliver demands from another LSP from the CDCs to the satellites. This setup ensures that LSPs keep the customer contact in the final delivery and solves one of the main point of criticism in cooperation in delivery systems.

It is assumed that these LSPs agree to cooperate under the guidance of a central coordinator, who is responsible for determining the optimal allocation of resources to achieve a system-wide optimum. This central coordinator could either be a third-party provider or a jointly managed entity operated by the participating LSPs. In either scenario, each LSP must be willing to share information regarding its demands and transportation resources.

Achieving a system optimum where, e.g., LSPs potentially share all demands or resources may result in solutions that individual LSPs perceive as unfair due to workload and cost distribution imbalances. Consequently, the challenge lies not only in minimizing total system costs but also in designing a framework where each LSP feels fairly treated. This includes both financial considerations, ensuring that each LSP gains a tangible benefit from the cooperation, and workload considerations, ensuring that no LSP faces significant disadvantages in areas such as market share (Mancini et al. 2021) or the ability to effectively serve its own demands. In addition, the relative differences between the LSPs in cost savings and the gain or loss of market share should not be too far apart. Each LSP should also serve a certain proportion of its own customers in order to, first, preserve its customer relationships and, second, ensure that it carries a significant share of its own operational workload. In particular, situations should be avoided in which an LSP performs a large part of the executed services but only contributes a small proportion of the demands to the coalition and vice versa. Therefore, it should be ensured that there is a link between the services provided and executed and the demands brought into the coalition, which leads to fair cooperation from both a workload and a monetary perspective.

For this purpose, we consider fairness constraints to prevent an unjustified imbalance in workload and costs. Given that these constraints affect the decision model, we assess various fairness constraints by examining their effects on the total system costs and their implications for fairness. The multi-day perspective described above becomes essential when considering service regulartiy and assessing the extent to which it matters whether these constraints must be met on each day or, on average, over several days.

2.3 Literature review

We structure the literature review as follows: Firstly, we delve into the existing literature concerning planning problems within 2T-CLSs in Section 2.3.1. Subsequently, Section 2.3.2 shows various approaches presented in the literature concerning cooperation in transportation, explicitly focusing on establishing fairness in cooperation. Lastly, Section 2.3.3 outlines the research gaps.

2.3.1 Two-tier city logistics systems

2T-CLSs were first introduced to the scientific literature by the work of Crainic et al. (2004). They considered the strategic positioning of satellite depots, formulated as a location-allocation problem. Since then, many publications have considered strategic, tactical, and operational planning problems. Crainic et al. (2009) presented the first general modelling framework for tactical planning in 2T-CLSs. Therefore, they considered the synchronization between the first and the second tier. Based on this, several studies have considered the exact operation synchronization of the first and second tiers (e.g., Grangier et al. 2016, Groß et al. 2020). Crainic et al. (2016) proposed a two-stage stochastic programming formulation to consider demand uncertainty in tactical planning. Crainic and Sgalambro (2014) presented a service network design formulation for the tactical planning in 2T-CLSs and discussed promising algorithmic solution perspectives. Fontaine et al. (2021) contributed a service network design model for a multi-modal 2T-CLS with inbound and outbound demand and solved it through a Benders decomposition algorithm. Studies on strategic issues focus mainly on the location of physical infrastructure like satellites or CDCs (e.g., Gianessi et al. 2016, Rao et al. 2015). Fontaine et al. (2023) proposed a general methodological framework based on continuous approximation theory for the design of city logistics networks with multiple LSPs. Zhu et al. (2023) concluded in a study on city logistics that co-modality is especially suitable for the middle mile in city logistics. Furthermore, Nataraj et al. (2019) considered the location of urban consolidation centers in a horizontal cooperation scenario. Other innovative approaches in the literature even consider the integration of freight transport into bus networks, which implicitly also assumes a 2T-CLS in which the first tier is carried out by bus services (Machado et al. 2023, Masson et al. 2017). For a broader overview of routing problems in the context of city logistics, we refer to Cattaruzza et al. (2017).

Regarding cooperation, Crainic et al. (2020) is the first publication that explicitly considers the cooperation of LSPs in a 2T-CLS so far. Their findings highlight that this cooperative approach resulted in noteworthy improvements in efficiency, both in terms of monetary and environmental impact.

2.3.2 Fairness in cooperative transportation

In general, the efficiency gains resulting from modern algorithmic solutions have led to increased attention being paid to the fair distribution of these efficiency gains among the groups involved. (De-Arteaga et al. 2022). In the context of cooperation between LSPs, a distinction can be made between centralized and decentralized cooperation (Gansterer and Hartl 2018). In centralized cooperations, the decisions of the coalitions are made by a central authority. In contrast, in decentralized cooperations, the participants cooperate individually or are supported by a central authority that does not have full information. In decentralized cooperations, auction models are mainly used to manage the exchange of demands. Thereby, the exchange of demands is ensured by a shared pool in which each player submits the demands it wants to trade. From this pool, the individual demands are then traded among the players by means of auctions (e.g., Elting et al. 2025, Gansterer and Hartl 2016, Gansterer et al. 2020b, Li et al. 2015, Karels et al. 2020). However, our study considers centralized planning, where a central authority has full information about all LSPs. The literature already investigates the fact that centralized cooperation can reduce overall system costs in other routing and transportation problems (e.g., Cruijssen et al. 2007a, Fernández et al. 2018, Irannezhad et al. 2018, Krajewska et al. 2008, Padmanabhan et al. 2022, Stellingwerf et al. 2018). In their survey about collaborative vehicle routing Gansterer and Hartl (2018) concluded that in most studies on centralized collaborations potential benefits of 20-30% are identified. With regard to cooperation at satellite depots, Bruni et al. (2024) investigated the benefits of vertical and horizontal cooperation and found significant cost savings for both forms of cooperation. In the context of service network design models, Zhang et al. (2024) investigated the benefits of cooperation between carriers with uncertainty in demands. However, existing studies rarely consider how to ensure that the cooperation is perceived as fair for all involved participants. There are two main approaches to ensuring fairness within centralized cooperation. First, cost allocation methods can be used to allocate the total costs incurred in the system to the individual LSPs according to a previously defined method (Guajardo and Rönnqvist 2016). The simplest methods are proportional cost allocation methods, in which the costs are allocated proportionately to a previously defined reference value. Furthermore, more advanced methods from cooperative game theory are introduced in the literature like the Shapley Value (Shapley 1953), the Nucleolus (Schmeidler 1969) and the Equal Profit Method (Frisk et al. 2010). In the context of 2T-CLSs, Gückel et al. (2024) compared different cost allocation methods and outlined the diverging interests of different sized LSPs in selecting a cost allocation method. For a general review of cost allocation methods, we refer to Guajardo and Rönnqvist (2016).

However, since cost allocation methods focus exclusively on a financial aspect, and other essential aspects (e.g. changes in market shares or the fulfillment of predominantly own demands) are disregarded, we pursue a different approach. This includes the introduction of constraints directly within the decision support system that balance the workload and/or cost of each coalition member. Within this context, Mancini et al. (2021) considered workload- and profit balance constraints for a vehicle routing problem with cooperative carriers. They concluded that workload balance constraints and minimum profit constraints over the whole planning horizon only slightly increase total system cost. In Crainic et al. (2020), constraints are added that ensure lower and upper bounds for the service costs and the workload of each LSP in the coalition. They reported significant impacts on the total cost, especially when the capacity constraints are tight. Further Matl et al. (2019) analyzed different equity functions for workload resources in a vehicle routing context. Bonku et al. (2025) investigated different objective functions, such as effectiveness and equity, in a collaborative vehicle routing model for food allocation in nonprofit settings. Recently, Soriano et al. (2023) considered a multi-depot vehicle routing problem in which they added a fairness objective function. This function aims to maximize, among all depots, the worst profit variation concerning the stand-alone solution to the classical cost minimization function. Sánchez et al. (2022) considered models quantifying different fairness and efficiency criteria of the individual routes.

Besides the existing research in cooperative transportation, the impact of fairness constraints on the total system is relatively unexplored, especially in a more realistic multi-day context. While fairness is typically considered at a single point in time, real-world scenarios require fairness over time to ensure that no LSP is disadvantaged across multiple days. This dynamic perspective introduces challenges, as the distribution of workload and costs must be adjusted across multiple planning periods to maintain fairness and sustain long-term cooperation.

2.3.3 Research gap

As highlighted, numerous studies have explored various planning problems across different levels within 2T-CLSs. However, research explicitly focusing on cooperation within 2T-CLSs remains notably limited. While the impact of cooperation on total costs in different contexts has already been explored, there is a gap in terms of integration and the impact of fairness constraints in multi-day planning to ensure that no LSP feels unfairly treated. Furthermore, current research lacks evidence on the effects of multi-day planning in which certain constraints related to both service regularity and fairness must be adhered to over several days. We close this research gap and explicitly model cooperation in a multi-day setting. This includes incorporating fairness constraints as well as considering the limited daily flexibility in the selection of services. This approach provides new insights into the design of cooperative 2T-CLSs, thereby contributing significantly to the existing body of literature.

2.4 Mathematical problem formulation

Section 2.4.1 introduces the notation. Section 2.4.2 presents the problem formulation, including the base model for full cooperation and additional constraints on cooperation fairness and service regularity. While various equity functions exist in the literature, we focus on constraints related to workload (market share and own demands served) and relative cost distribution among LSPs. These metrics are particularly relevant as they allow LSPs to compare their standing within the cooperation.

2.4.1 Notation

This section establishes the key sets and parameters for our base model. Additional notation that is only used for the additional constraints regarding workload and cost balance as well as regarding the service regularity is introduced in the respective sections for better readability. For a comprehensive overview, the notation is also provided in Table B.1 in Appendix B.1.

Sets

We consider a set of LSPs, denoted as $\mathcal{N}$. The physical infrastructure is represented by a set of CDCs, denoted as $\mathcal{E}$, and satellite facilities, denoted as $\mathcal{Z}$. The first-tier delivery from CDCs to satellites is operated by a set of urban vehicle services $\mathcal{R}$. Each service in this set can potentially be provided on every day within the schedule length. On the contrary, the set $\mathcal{D}$ represents the demands within the whole schedule length but can be further divided into subsets of the demands for each individual day. Each service is operated by one of the urban vehicles. Thereby, $\mathcal{M}$ represents the set of vehicle types that differs by capacity and cost components. The time dimension is considered through the set $\mathcal{T} = \{1, 2, ..., |\mathcal{T}|\}$ that represents the days in the schedule length and the set $\mathcal{P} = \{1, 2, ..., |\mathcal{P}|\}$ that represents the time periods during each day. In general, the set $\mathcal{T}$ does not necessarily have to refer to days but can also be understood as a consistency interval within which several sets of periods are connected. However, in the course of our study, driven by the problem definition, we will continue to refer to days.

Subsets

Each LSP n contributes its own subset of demands $\mathcal{D}(n)$ and its own subset of services, denoted as $\mathcal{R}(n)$, to the coalition. Additionally, the demands associated with day t are represented by the set $\mathcal{D}(t)$. These demands need to be fulfilled on the exact same day on which they occur.

Concerning services, distinct subsets are created: $\mathcal{R}^D(d, z)$ represents services that meet the release and due date requirements for demand d at satellite z. The release and due dates are periods during the day. Note that the due date of each demand is transferred to the satellites depending on the time-distance to the satellites (e.g., Fontaine et al. 2021). $\mathcal{R}^P(p, z)$ covers services operating during period p at satellite z, which is determined by the arrival time plus service time of the service at the satellite. The superscripts are introduced to separate the subsets, as they have the same number of indices. $\mathcal{R}(p, e, m)$ encompasses services that operate during period p, originating from CDC e, and utilizing urban vehicle type m. Please note that some parameters of the services, e.g., service times or start and end times, do not appear in the optimization model but are only used in advance as a pre-processing step to generate these subsets, which is why they are not explicitly introduced in the following. Further, in contrast to the demands, the same services are offered daily, and therefore, there is no explicit subset of services for a specific day.

Parameters

Each service r is associated with a specific vehicle type m_r, originates from a specific CDC e_r, and follows a sequential pattern of satellite visits. Each vehicle type m is associated with a specific capacity u_m. The volume of each demand d is denoted by v_d. For every satellite z, the following capacity-related parameters are introduced: a_{zpn} denotes the maximum number of services that LSP n can handle during period p at satellite z, and b_{zpn} represents the maximum demand volume that LSP n can handle during period p at satellite z. Additionally, h_{emn} reflects the number of urban vehicles owned by LSP n of vehicle type m available at CDC e.

Cost considerations are integrated as follows: Each service r is defined by its operating costs c_r, encompassing financial and environmental expenses linked to negative external effects. Like other studies in tactical planning, we approximate second-tier routing costs rather than explicitly modeling each delivery route. The cost associated with assigning demand d to satellite z and service r, denoted as s_{dzr}, captures the estimated last-mile routing expenses. While a full routing model is beyond the scope of this study, our approach aligns with established methods in the literature (e.g., Crainic et al. 2009, Fontaine et al. 2021) and provides a practical means of evaluating second-tier costs. Furthermore, the costs associated with the initial delivery of demand d from its origin to CDC e are represented by f_{de}.

2.4.2 Mathematical problem formulation

We introduce our base model for full cooperation where resources and demands are shared without restriction (Section 2.4.2.1). Subsequently, we incorporate fairness constraints related to workload (Section 2.4.2.2) and costs (Section 2.4.2.3). Then, we introduce constraints related to the regularity in selecting services (Section 2.4.2.4).

2.4.2.1 Base model

Our model utilizes the following binary decision variables: y_{rt}, which equals 1 if service $r \in \mathcal{R}$ operates on day $t \in \mathcal{T}$, and 0 otherwise. Additionally, x_{dzr} equals 1 if demand $d \in \mathcal{D}$ is assigned to satellite $z \in \mathcal{Z}$ and service $r \in \mathcal{R}$, and 0 otherwise. We do not need to include the day index t for the assignment variable x_{dzr} because our model formulation ensures that demands are assigned on the exact day they arise.

We formulate the problem as a service network design formulation as follows:

$$min \qquad \sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} c_r \cdot y_{rt} + \sum_{d\in\mathcal{D}}\sum_{z\in\mathcal{Z}}\sum_{r\in\mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr} \tag{2.1}$$

$$\sum_{z\in\mathcal{Z}}\sum_{r\in\mathcal{R}} x_{dzr} = 1 \qquad \forall d \in \mathcal{D} \tag{2.2}$$

$$\sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R} \setminus \mathcal{R}^D(d,z)} x_{dzr} = 0 \qquad \forall d \in \mathcal{D} \tag{2.3}$$

$$\sum_{z \in \mathcal{Z}} x_{dzr} \leq y_{rt} \qquad \forall d \in \mathcal{D}(t), r \in \mathcal{R}, t \in \mathcal{T} \tag{2.4}$$

$$\sum_{d \in \mathcal{D}(t)} \sum_{z \in \mathcal{Z}} v_d \cdot x_{dzr} \leq u_{m_r} \cdot y_{rt} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \tag{2.5}$$

$$\sum_{r \in \mathcal{R}(p,e,m)} y_{rt} \leq \sum_{n \in \mathcal{N}} h_{emn} \qquad \forall e \in \mathcal{E}, m \in \mathcal{M}, p \in \mathcal{P}, t \in \mathcal{T} \tag{2.6}$$

$$\sum_{d \in \mathcal{D}(t)} \sum_{r \in \mathcal{R}^P(p,z)} v_d \cdot x_{dzr} \leq \sum_{n \in \mathcal{N}} b_{zpn} \qquad \forall z \in \mathcal{Z}, p \in \mathcal{P}, t \in \mathcal{T} \tag{2.7}$$

$$\sum_{r \in \mathcal{R}^P(p,z)} y_{rt} \leq \sum_{n \in \mathcal{N}} a_{zpn} \qquad \forall z \in \mathcal{Z}, p \in \mathcal{P}, t \in \mathcal{T} \tag{2.8}$$

$$x_{dzr} \in \{0,1\} \qquad \forall d \in \mathcal{D}, z \in \mathcal{Z}, r \in \mathcal{R} \tag{2.9}$$

$$y_{rt} \in \{0,1\} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \tag{2.10}$$

The objective function (2.1) minimizes the total costs, which consist of the service operating costs (first sum) and the cost of assigning each demand to a service-satellite combination (second sum). The latter term explicitly accounts for the cost of transporting each demand from its origin to the corresponding CDC, where the assigned service begins. Please note that these costs extend beyond monetary aspects, as city logistics aims to reduce the external impacts of freight movements, such as congestion, emissions, and pollution (Crainic et al. 2009).

Constraints (2.2) ensure that each demand is assigned to exactly one service, while Constraints (2.3) make sure that demands can only be assigned to services that fulfill the release and due dates for the demand. We need two sets of constraints for this instead of one, as otherwise, multiple assignments would make it possible to circumvent the fairness constraints introduced later. Constraints (2.4) state that on each day, demands can only be assigned to a service if the service is selected on this day. Constraints (2.5) ensure that the capacity of the services is not exceeded. Additionally, these constraints guarantee that if a demand is assigned to a service, that service must operate on the day the demand arises. This ensures that demands match the available service capacities for that specific day on which they occur, maintaining an accurate daily alignment between services and demands. Constraints (2.6) limit the number of urban vehicles that are available at each CDC. Constraints (2.7) and (2.8) are the capacity constraints of the satellites. Thereby, Constraints (2.7) limit the demand volume that operates at satellite z in period p on day t while Constraints (2.8) limit the number of urban vehicles that can be handled at satellite z in period p on day t. These capacity constraints result from the available capacities across all LSPs. Constraints (2.9) and (2.10) specify the binary value domain of the decision variables.

2.4.2.2 Workload balance constraints

We follow the terminology of Mancini et al. (2021) by referring to constraints that limit the shifts in workload among the LSPs compared to the initial situation without cooperation as workload balance constraints. Let β be a parameter that sets a lower limit for the number of demands assigned to the services of each LSP, relative to the number of demands the LSP brings into the coalition. Specifically, $\beta = 1$ enforces that each LSP fulfills exactly as many demands by its own services as it brings into the coalition. Let α be a parameter that determines the share of each LSP's own demand volume that its own services must fulfill. At $\alpha = 1$, there is no demand sharing between the LSPs in the coalition as each LSP must fulfill 100% of its own demand volume by its own services, while $\alpha = 0$ allows for complete demand sharing. For further illustration, we consider a simple case with two LSPs. LSP 1 contributes 5 demands ($|\mathcal{D}_1| = 5$), and LSP 2 contributes 15 demands ($|\mathcal{D}_2| = 15$), where each demand has a volume of 5 ($v_d = 5 \ \forall d \in \mathcal{D}$). This results in a total demand volume of 25 for LSP 1 ($\sum_{d \in \mathcal{D}_1} v_d = 25$) and 75 for LSP 2 ($\sum_{d \in \mathcal{D}_2} v_d = 75$). Table 2.1 shows the minimum number of demands each LSP must fulfill using its own services for varying values of β. Table 2.2 shows the corresponding minimum own demand volume that must be fulfilled by each LSP's own services for varying values of α.

Table 2.1: Minimum number of demands each LSP must fulfill with own services under varying β

LSP	$\beta = 0$	$\beta = 0.25$	$\beta = 0.5$	$\beta = 0.75$	$\beta = 1.0$
LSP 1 ($\lvert D_n \rvert = 5$)	0	1.25	2.5	3.75	5
LSP 2 ($\lvert D_n \rvert = 15$)	0	3.75	7.5	11.25	15

Table 2.2: Minimum own demand volume each LSP must fulfill under varying α

LSP	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1.0$
LSP 1 ($\sum_{d \in \mathcal{D}_1} v_d = 25$)	0	6.25	12.5	18.75	25
LSP 2 ($\sum_{d \in \mathcal{D}_2} v_d = 75$)	0	18.75	37.5	56.25	75

For balancing the workload of the LSPs over the schedule length, we consider the following constraints regarding the number of demands that each LSP must fulfill and the share of own demand volume that each LSPs must fulfill by its own services.

$$\beta \cdot |D_n| \leq \sum_{d \in \mathcal{D}} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}(n)} x_{dzr} \qquad \forall n \in \mathcal{N} \tag{2.11}$$

$$\alpha \cdot \sum_{d \in \mathcal{D}(n)} v_d \leq \sum_{d \in \mathcal{D}(n)} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}(n)} x_{dzr} \cdot v_d \qquad \forall n \in \mathcal{N} \tag{2.12}$$

Constraints (2.11) ensure a lower limit for the number of demands assigned to the services of each LSP. They guarantee that the number of demands assigned to each LSP's services is greater than β times the number of demands of that LSP. However, the fulfilled demands do not have to correspond to the demands brought into the coalition by the respective LSP but can also represent demands from other LSPs. This ensures that each LSP does not lose too many demands and thereby too much market share compared to the initial situation without cooperation. Constraints (2.12) represent an even stricter restriction. They ensure that each LSP satisfies at least a particular share α of their own demand volume through their own services.

So far, these balance constraints hold on average over all days in the schedule length and ensure workload balance on average over these days. To ensure that on each individual day, the workload balance constraints are met, we can replace them as follows:

$$\beta \cdot |\mathcal{D}(n) \cap \mathcal{D}(t)| \leq \sum_{d \in \mathcal{D}(t)} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}(n)} x_{dzr} \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \tag{2.13}$$

$$\alpha \cdot \sum_{d \in \mathcal{D}(n) \cap \mathcal{D}(t)} v_d \leq \sum_{d \in \mathcal{D}(n) \cap \mathcal{D}(t)} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}(n)} x_{dzr} \cdot v_d \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \tag{2.14}$$

2.4.2.3 Cost balance constraints

Further, we consider constraints related to the total cost of each LSP in order to have a fair distribution of cost savings among the LSPs. The costs that an LSP n bears are made up of the operating costs of its services ($\sum_{r \in \mathcal{R}(n)} \sum_{t \in \mathcal{T}} c_r \cdot y_{rt}$) as well as the cost to deliver own demands to a CDC and the final delivery cost to deliver the demands from the satellite to their final location ($\sum_{d \in \mathcal{D}(n)} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr}$).

Let C_n be the standalone cost for each LSP n when acting independently without cooperating with other LSPs in the coalition. Thereby, C_n must be determined beforehand by solving the model for the sets and parameters of LSP n in isolation. The total standalone cost for all LSPs acting independently $C_\mathcal{N}$ is defined by $C_\mathcal{N} = \sum_{n \in \mathcal{N}} C_n$.

Let γ be a parameter that provides flexibility in the cost distribution, allowing a maximum percentage markup on the costs that would be allocated to the LSP in the case of exact equal relative cost savings among all LSPs. Setting γ to 0 could lead to no feasible solution, as it might not be possible to have a solution in which each LSP has the exact same relative cost savings.

$$\sum_{r \in \mathcal{R}(n)} \sum_{t \in \mathcal{T}} c_r \cdot y_{rt} + \sum_{d \in \mathcal{D}(n)} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr} \leq C_n \qquad \forall n \in \mathcal{N} \tag{2.15}$$

$$\begin{aligned} &\sum_{r \in \mathcal{R}(n)} \sum_{t \in \mathcal{T}} c_r \cdot y_{rt} + \sum_{d \in \mathcal{D}(n)} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr} \leq \\ &(1+\gamma) \cdot \frac{C_n}{C_\mathcal{N}} \cdot (\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}} c_r \cdot y_{rt} + \sum_{d \in \mathcal{D}} \sum_{z \in \mathcal{Z}} \sum_{r \in \mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr}) \qquad \forall n \in \mathcal{N} \end{aligned} \tag{2.16}$$

Constraints (2.15) ensure that each LSP bears at most the cost that would incur if the LSP would act independently without cooperating with other LSPs in the coalition. Constraints (2.16) are stricter. They ensure that the costs for LSP n must be smaller or equal to the costs $(1+\gamma) \cdot C_n / C_\mathcal{N}$ times the total cost of the coalition.

The presented constraints hold over all days in the schedule length and ensure cost balance on average over these days. To ensure that on each individual day, these balances hold, we replace them by Constraints (2.17) and (2.18). Thereby, C_{nt} represents the standalone cost of LSP n on day t, while $C_{\mathcal{N}t} = \sum_{n \in N} C_{nt} \forall t \in \mathcal{T}$ represents the cumulative standalone cost of all LSPs of the coalition $\mathcal{N}$ on day t.

$$\sum_{r\in\mathcal{R}(n)} c_r \cdot y_{rt} + \sum_{d\in\mathcal{D}(n)\cup\mathcal{D}(t)} \sum_{z\in\mathcal{Z}} \sum_{r\in\mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr} \leq C_{nt} \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \tag{2.17}$$

$$\sum_{r\in\mathcal{R}(n)} c_r \cdot y_{rt} + \sum_{d\in\mathcal{D}_n\cup\mathcal{D}(t)} \sum_{z\in\mathcal{Z}} \sum_{r\in\mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr} \leq$$
$$(1+\gamma) \cdot \frac{C_{nt}}{C_{\mathcal{N}t}} \cdot (\sum_{r\in\mathcal{R}} c_r \cdot y_{rt} + \sum_{d\in\mathcal{D}(t)} \sum_{z\in\mathcal{Z}} \sum_{r\in\mathcal{R}} (s_{dzr} + f_{de_r}) \cdot x_{dzr}) \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \tag{2.18}$$

2.4.2.4 Service regularity constraints

To prevent the execution of entirely different services each day, we include constraints on the regular operation of services. For this purpose, we introduce the binary auxiliary decision variable $\hat{Y}_r$, set to 1 if service $r \in \mathcal{R}$ operates on every day $t \in \mathcal{T}$.

We incorporate a service regularity parameter, θ, to specify the proportion of selected services that should operate each day. For example, $\theta = 0.7$ indicates that 70 percent of all selected services should be those that operate every day. To enforce this, we introduce the following constraints:

$$\hat{Y}_r \leq y_{tr} \qquad \forall r \in \mathcal{R}, t \in \mathcal{T} \tag{2.19}$$

$$\theta \cdot \sum_{t\in\mathcal{T}} \sum_{r\in\mathcal{R}} y_{tr} \leq \sum_{r\in\mathcal{R}} \hat{Y}_r \cdot |\mathcal{T}| \tag{2.20}$$

$$\hat{Y}_r \in \{0,1\} \qquad \forall r \in \mathcal{R} \tag{2.21}$$

Constraints (2.19) enforce that if a service is defined as a permanent service ($\hat{Y}_r = 1$), then it must be selected on each day ($y_{tr} = 1 \forall t \in \mathcal{T}$). Constraint (2.20) ensures that at least a fraction θ of the selected services must be defined as permanent services. Accordingly, $\theta = 0.7$, for example, means that at least 70% of the services selected in the schedule length ($y_{tr} = 1$) must be services that operate on every single day ($\hat{Y}_r = 1$). Constraints (2.21) ensure the binary value domain.

2.5 Numerical study

Within this numerical study, our goal is to generate insights into both the general impacts of cooperation on monetary and environmental KPIs and to investigate the effects of the introduced fairness constraints in a multi-day setting on the overall system as well as on the individual LSPs. In addition, we want to quantify the effects of the service regularity. These numerical experiments provide the basis for the policy implications formulated in Section 2.6. We start by describing the problem-specific instances in Section 2.5.1 and the numerical setup in Section 2.5.2. Section 2.5.3 shows the impact of cooperation on major KPIs. Afterward, Section 2.5.4 analyzes the effects on total system costs when adding workload and cost balance constraints to the base model. Section 2.5.5 discusses the effects of these constraints from a financial and a workload perspective. In Section 2.5.6, we investigate the impact of integrating constraints regarding the service regularity.

Section 2.5.7 shows further experiments regarding sustainability objectives and the robustness of our obtained results.

2.5.1 Instance generation

As there are no publicly available realistic data sets in the area of 2T-CLSs, we are guided by similar publications in the field (e.g., Crainic et al. 2004, Fontaine et al. 2021, Lange et al. 2025) and generate our instances based on a real city.

We consider a schedule length of six days. During the days, we consider $|\mathcal{P}| = 72$ periods, each having 10 minutes, representing one business day (12 operating hours). We generate a network containing three CDCs and six satellites. The Euclidean distance determines distances between demand locations, satellites, and CDCs.

We consider four different urban vehicle types: small tram, large tram, small truck, and large truck, with capacities of 50 for small and 75 for large vehicles. The cost of the services c_r depends on the vehicle type and the travel distance. We set the fixed cost for the vehicles as follows: 10 for small trams, 15 for large trams, 13 for small trucks, and 18 for large trucks. The variable costs are determined by multiplying the travel distance by 1.2 for small trams, 1.5 for small trucks, 1.7 for large trams, and 2.0 for large trucks. The lower cost of trams is justified by their lower environmental impact cost. Additionally, the travel speed for trams is set to 25 km/h and for trucks to 20 km/h, as trucks are more affected by congestion and traffic signals. Services are generated based on plausible routes that connect the CDCs with a sequence of satellites. Tram-operated services can only reach satellites located at tram stops, which account for half of all satellites. Each service is replicated three times within the day. The first starting period is randomly set between periods 5 and 15. The following replication starts 15 periods later. We assume a service time of two periods for each satellite.

Similar to other studies on cooperative problems, we focus on a small coalition to draw better conclusions for individual cooperation partners (e.g., Fernández et al. 2018, Gansterer and Hartl 2016, Gansterer et al. 2019, Padmanabhan et al. 2022, Zhang et al. 2024). We consider three LSPs operating in this system with different numbers of demands. Specifically, LSP 1 has between 12 and 15 demands per day, LSP 2 has between 17 and 20 demands per day, and LSP 3 has between 22 and 25 demands per day, respectively, leading to $51 \leq |\mathcal{D}(t)| \leq 60 \forall t \in \mathcal{T}$ and consequently $306 \leq |\mathcal{D}| \leq 360$ in each instance. This setup allows us to analyze cooperation among LSPs of different sizes—small, medium, and large—providing insights into how cooperation affects LSPs of different sizes. Note that the demands vary from day to day and differ in number, volume, and location. All demand volumes are randomly set between 5 and 10. Thereby, each value is uniformly distributed across the specified range. Each demand's release date (RD) is randomly set between periods 0 and 50. The due date (DD) is then determined by $DD = RD + \text{randint}(12, 22)$ to ensure time feasibility. Demands beyond this period are not considered to maintain a feasible scheduling framework within the given planning horizon. The tight delivery time windows reflect the fast-paced nature of city logistics, where timely deliveries are essential. Demand locations are sampled for all instances based on publicly available data on population density across different city districts.

LSP 1 provides 21 services, while LSP 2 provides 27 services, and LSP 3 provides 33 services. The services are generated independently of each other and the different vehicle types occur with approximately the same frequency. The assignment cost s_{dzr} is determined using the Euclidean distance between the satellite z and demand location d, a commonly used proxy for second-tier routing costs (Fontaine et al. 2021, Crainic et al. 2020). The cost for delivering demands to CDCs f_{de} are randomly set between 1 and 5 to reflect cost variability across CDCs, given the complexity of real-world cost estimation. For capacity constraints, we assume that each LSP is allowed to operate one vehicle per period at each satellite (a_{zpn}) and a demand volume of 30 units (b_{zpn}). Furthermore, we limit the number of vehicles of each type m available at CDC e for all LSPs to one (h_{emn}). Within this framework, we generate ten different instances, which we will use for all our numerical experiments in the upcoming sections.

2.5.2 Numerical setup

We implemented the mathematical model in Python 3.12 using Gurobi 11 as a solver. All experiments are carried out on an AMD Ryzen 9 5950X 16-Core Processor, 3.40 GHz with 128 GB RAM. All experiments were performed on six threads. We set a time limit of 60 minutes for Gurobi. With this setting, Gurobi could solve some of the experiments in the upcoming sections exactly, while minor GAPs remained in others. The average GAP across all experiments was 0.28%, and the maximum GAP was 1.4%. Particularly the experiments in which constraints were in force, which apply over the entire period and thus connect the individual days with each other, were hard to solve.

2.5.3 Cooperation without fairness consideration

In this section, we identify the impact of cooperation on various KPIs. We have included cost-based KPIs, which are particularly relevant for LSPs and their customers, as they want their deliveries to be as cheap as possible, and environment-based KPIs, which are particularly relevant for the municipal authority and the city's residents, as they are primarily interested in minimizing negative impacts.

We achieve this by solving the base model (Section 2.4.2.1) for each LSP separately and aggregating the results (No Coop.), as well as by solving the base model collectively for all LSPs without considering fairness (Coop.). To calculate CO2 equivalent emissions, we assume an Emission Factor (EF) of 303.3 g per kilometer for services with small trucks (Umweltbundesamt Österreich 2024). For large trucks, we apply a 50% increase, bringing the emissions to 454.95 g per kilometer. In contrast, trams generally have lower CO2 equivalent emissions compared to trucks. However, these values can vary significantly depending on the local electricity mix. For the purposes of this study, we assume emissions of 40 g per kilometer for small trams and 80 g per kilometer for large trams.

As shown in Table 2.3, cooperation leads to substantial cost reductions, with total system costs decreasing by 22.4%, mainly due to a 34.9% reduction in service operating costs. Additionally, cooperation results in a 28.9% reduction in the number of selected services while utilization increases by 23.4 percentage points, highlighting a more efficient use of available vehicle capacity. Notably,

Table 2.3: Impact of cooperation on major KPIs

KPI	**Definition**	**No Coop.**	**Coop.**	Δ [%]
Total system cost	Objective function (2.1)	4518.2	3507.7	-22.4
Service operating cost	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} c_r \cdot y_{rt}$	2330.9	1516.6	-34.9
CDC assignment cost	$\sum_{d\in\mathcal{D}}\sum_{z\in\mathcal{Z}}\sum_{r\in\mathcal{R}} f_{de_r} \cdot x_{dzr}$	893.2	869.0	-2.7
Second tier cost	$\sum_{d\in\mathcal{D}}\sum_{z\in\mathcal{Z}}\sum_{r\in\mathcal{R}} s_{dzr} \cdot x_{dzr}$	1294.2	1122.1	-13.3
No. of selected services	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} y_{rt}$	57.4	40.8	-28.9
No. of stops at satellites*	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} z_r \cdot y_{rt}$	113.4	61.2	-46.0
Utilization of operated services [%]	$\frac{\sum_{d\in\mathcal{D}} v_d}{\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} u_{m_r}\cdot y_{rt}}$	78.7	97.1	23.4
Share of large vehicles* [%]	$\frac{\sum_{r\in\acute{\mathcal{R}}}\sum_{t\in\mathcal{T}} u_{m_r}\cdot y_{rt}}{\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} u_{m_r}\cdot y_{rt}}$	28.5	61.4	115.4
Share of trams* [%]	$\frac{\sum_{r\in\hat{\mathcal{R}}}\sum_{t\in\mathcal{T}} u_{m_r}\cdot y_{rt}}{\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} u_{m_r}\cdot y_{rt}}$	55.3	69.5	25.7
Driven distance of services [km]	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} distance(r) \cdot y_{rt}$	889.5	466.5	-47.6
CO2 emissions [kg CO2]	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} distance(r) \cdot EF_r \cdot y_{rt}$	184.9	97.8	-47.0

*$\acute{\mathcal{R}}$ represents the subset of services operated by large urban vehicles, $\hat{\mathcal{R}}$ the subset of services operated by trams and z_r the number of stops at satellites of service r.

the number of stops at satellites reduces even more by 46.0%.

Beyond cost savings, cooperation also leads to notable environmental benefits. The total driven distance decreases by 47.6%, leading to a 47.0% reduction in CO2 emissions. At the same time, the share of large vehicles has doubled, and the share of trams has increased by 25.7 percentage points, reflecting a shift towards higher-capacity and more sustainable transport modes. These results underscore the dual benefits of cooperation: cost efficiency for logistics service providers and reduced externalities for cities. The impact on these KPIs shows that cooperation not only leads to monetary savings but also significantly reduces the negative impact on the environment, such as noise, CO2 emission, traffic congestion, etc., through the more efficient use of vehicles, which is particularly important in the context of city logistics.

Despite the clear positive effects of cooperation, the outcomes can also lead to unfair distribution of costs and market shares, and without considering that each LSP should fulfill its own demand volume primarily through its services. In Table 2.4, we show the average minimum and average maximum values for the relative cost savings, the relative deviation in market share, and the share of own demand volume allocated to own services across the LSPs. The average minimum value, therefore, indicates how low this KPI is on average for the LSPs with the lowest value.

As can be seen, there are large deviations in all three KPIs. This indicates clear unfairness and instability within the coalition of LSPs, highlighting the urgent need for additional fairness constraints. Further, each service provider only satisfies a small fraction of its own demands, as the base model does not take this allocation into account. These results show that full cooperation without taking the before-introduced fairness constraints into account does not lead to acceptable

Table 2.4: Average minimum and maximum value of the LSPs for various KPIs

KPI	Avg. minimum	Avg. maximum
Relative cost savings [%]	6.3	37.6
Relative deviation in market shares [%]	-38.8	47.1
Share of own demand volume [%]	17.8	50.7

results for every LSP and that embedding appropriate constraints is necessary.

2.5.4 Impacts of fairness constraints on total system cost

We examine the impact of the workload (Section 2.4.2.2) and cost balance (Section 2.4.2.3) constraints on the total system costs by adding them to the base model with different parameter settings. For this, we distinguish between three scenarios in the following experiments:

1. The constraints that ensure daily workload/cost balance are enforced (referred to as *Individual day*).
2. We divide the schedule length into two and three-day segments. The model is solved individually for each segment, whereby the constraints that ensure average workload/cost balance over these segments are enforced. We report the results as a sum over the segments (referred to as *Two days* and *Three days*).
3. The constraints that ensure average workload/cost balance over the entire schedule length of six days are enforced (referred to as *All days*).

2.5.4.1 Workload balance

We start our investigation by analyzing the influence of Constraints (2.11) and (2.13) that limit the loss of market share, as they are less restrictive and focus exclusively on the number of demands and do not consider their volume or affiliation to an LSP. In our analysis, we gradually increase β from 0.7 (each LSP must be assigned at least 70% as many demands as it contributes to the coalition) to 1. Figure 2.3 shows the results.

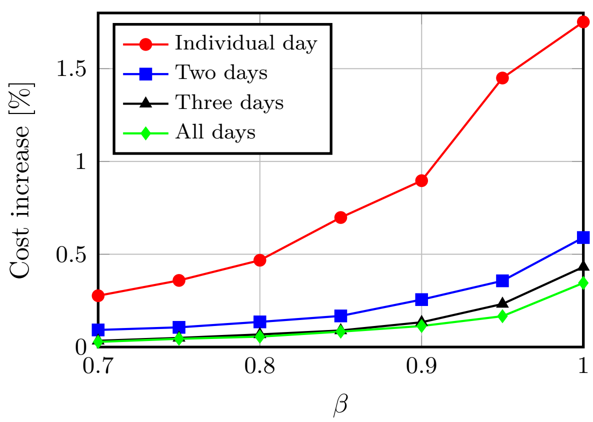

Figure 2.3: Cost impact of varying β

We find that, on average, the inclusion of these constraints leads to an increase in total system costs of up to 1.75% when β is set to 1, and the constraints are enforced individually on each day. These results are consistent with the results obtained by Gansterer and Hartl (2018) with the introduction of similar workload balance constraints in a vehicle routing context, as they also observe only minor cost increases. In addition, it further shows that the more days over which the constraints hold on average, the lower the increase in costs. Furthermore, we find that a large part of the increase only occurs with a very restrictive $\beta \geq 0.9$.

In the next step, we determine the impact on the total system cost of Constraints (2.12) and (2.14), which restrict the demand sharing of LSPs. These are more restrictive as they consider both the demand volume and the LSP to which the demands belong.

We increase α gradually from 0.7 (each LSP must satisfy at least 70 percent of its own demand volume with its own services) to 1 (no demand sharing possible). Figure 2.4 shows the results. In contrast to the previous analysis, we now see a very high impact on total system costs. When demand sharing is totally restricted ($\alpha = 1$), the total cost across all instances increases by 21.9% on average. Furthermore, we find that for all α values less than one, it is significantly cheaper to enforce these constraints over a more extended period of time rather than on each individual day. Even stretching the constraints over two days leads to significant savings. We also find that by sharing only a small portion of the demand volume, a large portion of the potential cost savings that would be possible by full demand sharing can be realized. This is demonstrated by the significant increase in total system costs for $\alpha > 0.8$.

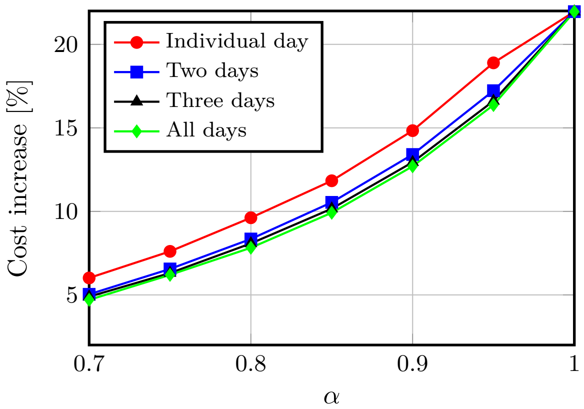

Figure 2.4: Cost impact of varying α

Overall, these two analyses show that the inclusion of constraints that only affect the number of demands leads merely to a slight increase in total system costs, while constraints that limit demand sharing are associated with a high increase in costs. Furthermore, we find that for both constraints, a large part of the cost increase only happens when the constraints are enforced with strict α and β parameter settings.

2.5.4.2 Cost balance

We start our investigation by including the less restrictive cost-balance Constraints (2.15) and (2.17) that ensure that the cost of each LSP is not greater than its stand-alone cost. Table 2.5

shows the cost increase compared to the base model without these constraints.

Table 2.5: Cost impact of stand-alone cost constraints

Individual day	**Two days**	**Three days**	**All days**
0.36%	0.14%	0.11%	0.08%

We find that including these constraints leads to a marginal increase in total system costs close to zero. Even if these constraints are enforced every single day, the total cost only increases by 0.36%. This shows that including this simple fairness criterion, namely that every LSP must benefit financially from the cooperation, causes almost no additional costs.

In the next step, we include the more restrictive constraints on the relative cost savings (Constraints (2.16) and (2.18). We gradually increase γ from 1 to 6 percent. The results are shown in Figure 2.5.

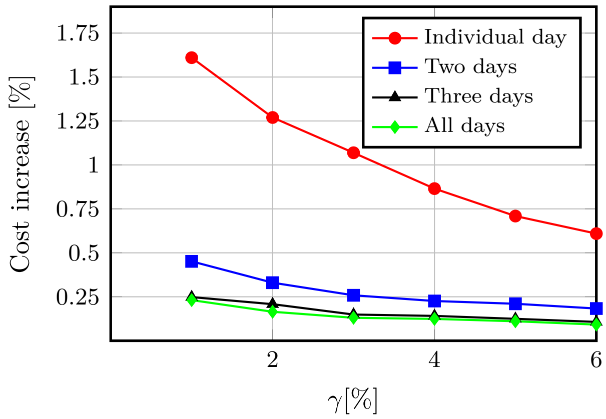

Figure 2.5: Cost impact of varying γ

We find that even for $\gamma = 1\%$ with daily enforcement of the constraints, the total system cost only increases by 1.6%. As soon as the constraints are stretched over segments of two or more days, resulting in balancing effects, the cost increases are close to zero.

Overall, these two experiments show that the inclusion of constraints regarding the cost distribution of LSPs only leads to marginal cost increases that are close to zero as soon as they are stretched over several days or as soon as the constraints are provided with less strict parameter settings.

2.5.5 Impact of fairness constraints on financial and workload aspects

In this section, we discuss the impact of the fairness constraints under both financial (Section 2.5.5.1) and workload (Section 2.5.5.2) aspects. The analysis aims to highlight the relationship between workload and cost balance constraints and to examine whether enforcing one type of fairness also contributes to achieving the other. Further, we investigate the daily variation effects on workload and costs when constraints are met over the entire period rather than on each individual day (Section 2.5.5.3). In addition to the differently sized LSPs used in the main analysis, we created a second, *patterned-demand instance set* where the LSPs are equal in size but differ in seasonal demand profiles. We generated ten instances in which LSP 1 faces a front-loaded pattern (25, 25, 20, 10, 5 demands across days 1–5), while LSP 2 and LSP 3 each have constant 15 demands per day. We re-ran all experiments on this instance set; to avoid overloading the main text, detailed results and figures are provided in Appendix B.2. Key observations from that analysis are referenced where relevant below.

2.5.5.1 Cost aspects

In this section, we take fairness considerations regarding the individual relative cost savings of the LSPs into account. We do this by comparing the individual cost savings of the LSPs for three cases: In Case 1, we solve the base model without any additional fairness constraints. In Case 2, we include Constraints (2.11) with β set to 1 and Constraints (2.12) with α set to a moderate value of 0.8. In Case 3, we include the Constraints (2.16) with γ set to 1%. Figure 2.6 depicts the results as boxplots. These boxplots show the distribution of the individual relative cost savings of the three LSPs across all instances (LSP 1 is the smallest, LSP 2 the medium, LSP 3 the largest).

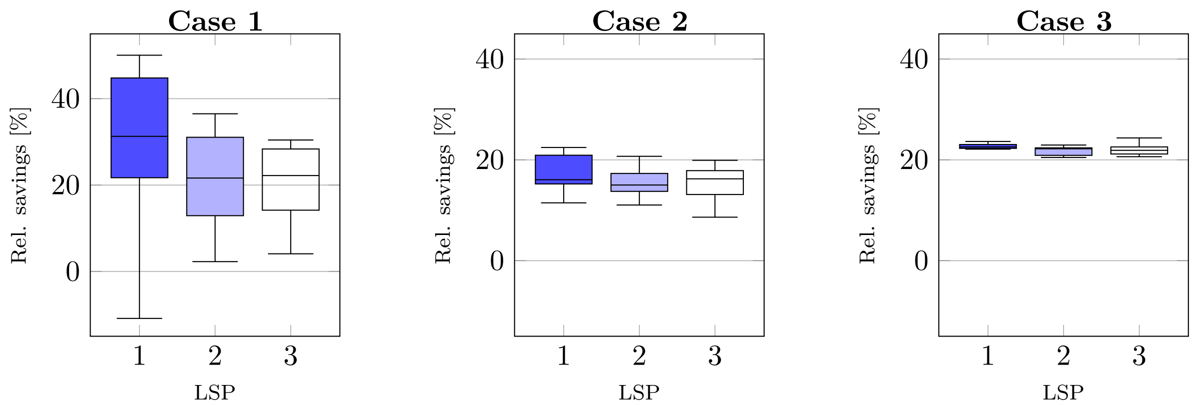

Figure 2.6: Relative cost savings of LSPs for the different cases [%]

As can be seen, the individual cost savings in Case 1 are subject to extreme fluctuations. In some cases, the individual cost savings are close to 0, or even in the negative range, which represents an extreme violation of the most straightforward fairness criterion that each LSPs should have financial benefits. In Case 2, individual savings are no longer subject to such strong fluctuations, as it is ensured that each LSP bears at least a particular share of the costs since each LSP must satisfy as many demands as it contributes to the coalition. However, the mean values of the savings are lower overall, as these constraints significantly increase the total system costs, as already shown. Further, there are still large deviations in the relative cost savings among the LSPs on individual instances. Only in Case 3 is it ensured that each LSP has a relative financial advantage of almost the same size without the total system cost rising sharply. These results indicate that enforcing workload balance (Case 2) does reduce dispersion in individual cost savings and thus contributes to more equitable outcomes, yet at the expense of higher system cost. However, the resulting cost balance remains markedly weaker than when cost balance is enforced directly (Case 3). A qualitatively identical pattern holds in the patterned-demand instance set (Appendix B.2).

2.5.5.2 Workload aspects

In this section, we examine the percentage deviations in the number of assigned demands of the individual LSPs for the exact same three cases. Figure 2.7 shows the results. A value of, e.g., 20% means that the LSP has 20% more demands assigned than it contributes to the coalition.

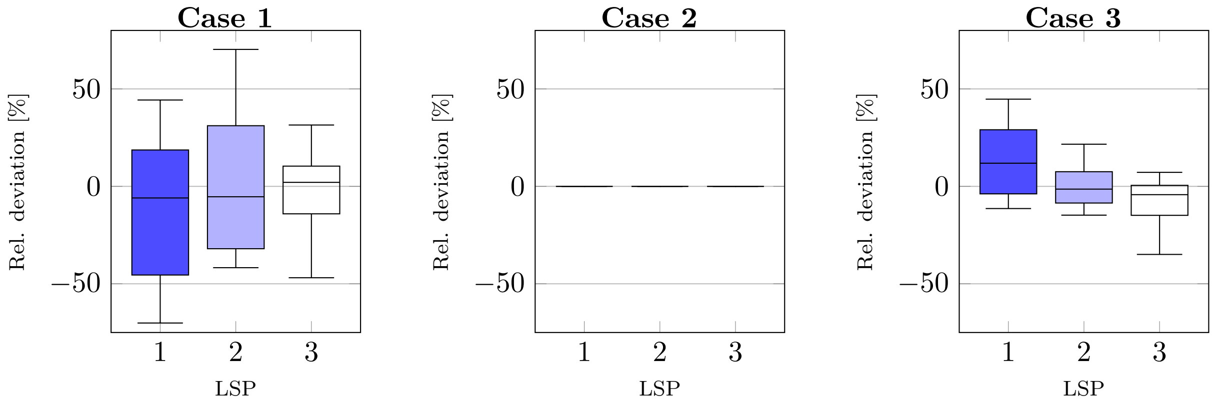

Figure 2.7: Relative deviation in the number of assigned demands for the different cases [%]

In Case 1, the relative changes in the number of demands for all three LSPs are subject to huge fluctuations as no single constraint limits the workload for any of the LSPs.

In Case 3, the fluctuations due to the introduction of cost balance constraints are much smaller but still significant. This shows that introducing the cost balance constraints also leads to a certain balance in the workload but still allows deviations in the number of demands. In addition, no consideration is given to whether the demands assigned to an LSP are also the respective own demands of this LSP. This further underscores that cost and workload balance constraints are related to some extent: enforcing one type of fairness also partially restricts the other. However, the remaining fluctuations indicate that such indirect effects may still lead to outcomes that can be perceived as unfair. Only in Case 2 does the introduction of constraints result in absolutely no change in market shares, as changes in market shares are forbidden due to the constraints.

Overall, our results show that both workload and cost balance constraints are necessary to enable a fair coalition in both aspects. We illustrated that cost-based fairness constraints only insufficiently consider workload aspects implicitly and vice versa. The same conclusion holds under patterned seasonal demands (Appendix B.2): workload constraints alone do not guarantee cost fairness, and cost constraints alone do not guarantee workload fairness. In practice, constraints on both metrics are needed to jointly achieve equitable outcomes.

2.5.5.3 Daily fluctuations under multi-day fairness constraints

The inclusion of constraints that apply on average over a horizon of multiple days rather than on each individual day allows daily fluctuations. Therefore, we examine the daily fluctuations for the three LSPs if these fairness constraints are enforced on average over the whole schedule length. To do this, we solve the base model, including the respective constraints regarding costs and workload, with the parameter settings described below.

Figure 2.8 shows the distribution of the number of daily demands that the respective LSPs satisfy with their own services relative to the number of demands they bring into the coalition on the respective day. A value of 1, therefore, means that exactly as many demands are fulfilled by services of an LSP as the respective LSP contributes to the coalition on that day. For this analysis, we include Constraints (2.11) and set $\beta = 1$ to ensure that each LSP satisfies exactly as many demands with its services over the entire schedule length as it contributes to the coalition.

Figure 2.9 shows the distribution of the daily own share of demand volume, which the LSPs satisfy with their own services. For this analysis, we include Constraints (2.12) and set $\alpha = 0.8$.

Figure 2.10 shows the distribution of the daily cost savings of the coalition as a whole ($\mathcal{N}$) and for the respective LSPs. For this analysis, we include Constraints (2.16) and set $\gamma = 1\%$.

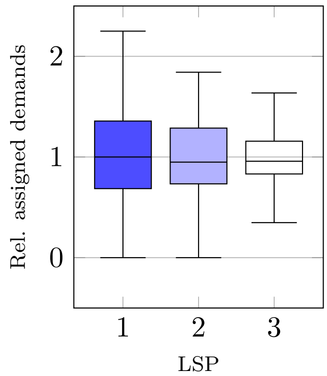

Figure 2.8: Distribution of the daily number of demands assigned to own services relative to the own number of demands for $\beta = 1$

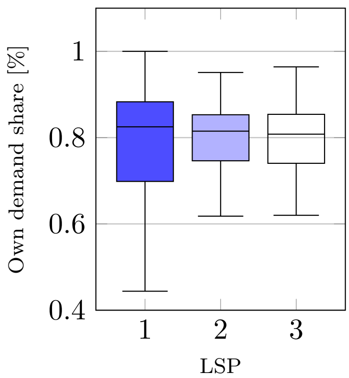

Figure 2.9: Distribution of the daily share of own demand volume assigned to own services for $\alpha = 0.8$

Overall, it can be seen that enforcing the constraints over a period of several days results in high deviations on a daily basis. In all three analyses, these are particularly strong for the smallest LSP (LSP 1), as the balancing effects are most pronounced for it in relative terms. In the patterned-demand instance set, the LSP with the front-loaded demand pattern (LSP 1) exhibits larger relative day-to-day fluctuations in both workload and cost-related metrics than the LSPs with constant demand.

In the case of LSPs with different seasonal demand patterns, it can be observed that the LSP

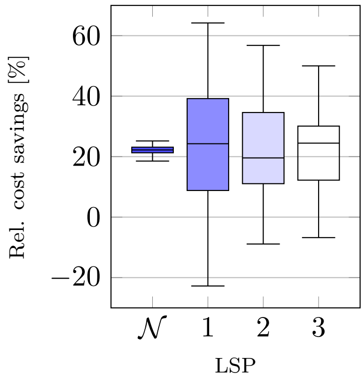

Figure 2.10: Distribution of daily relative cost savings [%] for $\gamma = 1\%$

with a high deviation of number of demands within the planning horizon, face higher relative daily fluctuations in these metrics compared to LSPs with constant demand pattern. If these effects are undesirable or pose a problem, both the constraints that hold on average over the schedule length and the constraints relating to individual days can be enforced simultaneously so that even within the individual days, the fluctuations only occur in a certain interval.

2.5.6 Impact of service regularity

In this section, we examine the effects of regularly operated services over the entire schedule length. To do this, we add the Constraints (2.19) and (2.20) to the base model and gradually increase θ from 0 (no regularity) to 1 (complete regularity). We distinguish between two cases. In the first case (moderate fluctuation), we take the instances we have used for previous experiments. In the second case (high fluctuation), we modify these instances by randomly removing between 0 and $3 \cdot (60 - |\mathcal{D}(t)|)$ demands each day t. This approach increases the fluctuation of daily demand, removing more demands on days with fewer already and fewer on days close to the maximum of 60 demands. Additionally, we multiply the demand volumes v_d by a different random number between 0.7 and 1.3 each day. With this approach, we achieve a higher daily fluctuation in both the number of demands and the demand volume, allowing us to evaluate whether this has an effect on the costs of service regularity. The results are shown in Figure 2.11.

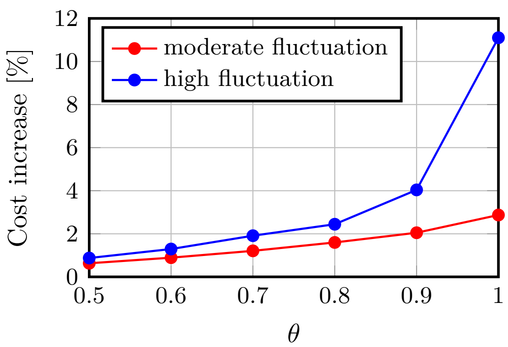

Figure 2.11: Cost impact of varying θ [%]

We see that in case of moderate fluctuation, the cost increase with complete service regularity amounts to almost 3% on average. In addition, we observe that for all values of θ, the cost increase in case of high fluctuation is significantly higher. With complete regularity, the costs increase by over 11%. In particular, a very high increase between $\theta = 0.9$ and $\theta = 1$ can be seen, which is due to the fact that there are no compensation options for fluctuating demand if all services have to be operated regularly. This clearly indicates that the regularity of the services is particularly detrimental to the system if there are strong fluctuations in the demand structures between days, as the service capacity provided cannot then adapt to the demand fluctuations between days. This shows that service regularity makes particular sense if there is also a corresponding demand regularity. Detailed analysis further revealed the following environmental impact in case of high fluctuation when comparing $\theta = 0$ and $\theta = 1$: the average number of selected services increases by 27.4%, the average utilization of the services decreases by 24.1%, and the driven distance of the services increases by 10.8%.

Overall, the results show that complete service regularity leads to both monetary cost increases and higher environmental impact costs. However, most of the cost increases only occur with high parameter settings for θ, and this effect is stronger the greater the daily fluctuation in the demand structure is.

2.5.7 Further experiments

In further experiments, we first analyse the impact of cooperation as well as the impact of the before introduced constraints on different sustainability oriented objectives (Section 2.5.7.1). Then, we derive insights on how our obtained results are dependent on the ratio between demand volumes and capacities (Section 2.5.7.2).

2.5.7.1 Sustainability objectives

In this section, we evaluate the impact of alternative sustainability-oriented objectives by replacing the main objective function (2.1) with other measures directly linked to environmental performance. Specifically, we consider the number of urban vehicle stops at satellites, the total distance traveled by urban vehicles, and the resulting CO_2 emissions within the city. These objectives are closely aligned with the United Nations Sustainable Development Goals, particularly *Sustainable Cities and Communities* and *Climate Action* (United Nations 2015). In this experiment, we distinguish between full cooperation and the case of constrained cooperation. For the constrained case, we use the previously introduced constraints and set the parameters for the constraints to very strict values as follows: $\alpha = 0.95$, $\beta = 1$, and $\gamma = 0.01$. Thereby, the constraints hold on average over the whole planning horizon (Constraints (2.11), (2.12), and (2.16)). Table 2.6 reports the results of this analysis. The reported values indicate the percentage change in each sustainability objective relative to a no-cooperation baseline; negative values correspond to reductions (i.e., savings) and thus improvements in the respective objective.

As the results show, cooperation substantially advances these sustainability goals. Even under fairness constraints that are enforced on average across the entire planning horizon, a significant share of the potential savings in stops, distance, and emissions can still be realized. This demon-

Table 2.6: Cost impact of stand-alone cost constraints

Objective	Expression	Full cooperation	Constrained cooperation
No. of stops at satellites	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} z_r \cdot y_{rt}$	-60.47%	-53.29%
Driven distance of services [km]	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} distance(r) \cdot y_{rt}$	- 64.22%	-45.07%
CO2 emissions [kg CO2]	$\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} distance(r) \cdot EF_r \cdot y_{rt}$	-67.23%	-47.71%

strates that cooperative two-tier city logistics not only reduces operational costs but also provides a robust pathway toward more sustainable urban freight systems, offering actionable insights for both logistics providers and municipal policymakers.

2.5.7.2 Impact of demand volumes

In this section, we analyse how cooperative savings are affected by different ratios between demand volumes and vehicle capacities. As this relationship was not derived from real-world data, it is important to assess its impact on the reported cooperative savings. To this end, we scale the demand volumes v_d by multiplicative factors of 0.25 and 0.5, using 1 as the unchanged baseline, and evaluate the effect on total cost, service operating cost, and the number of selected services. These are the components most affected by demand scaling; assignment costs to satellites and CDCs are unaffected by demand volumes. This setup also reflects alternative plausible real-world situations in which vehicles operate with more spare capacity, allowing demands to be consolidated more effectively. Table 2.7 reports the savings obtained under these settings by comparing the case without cooperation to the case with full cooperation.

Table 2.7: Savings [%] for different demand scaling factors

KPI	0.25	0.5	1
Total cost	28.1	25.0	22.4
Service operating cost	49.8	37.8	34.9
No. of selected services	49.7	37.8	28.7

As shown, savings increase as demand volumes decrease, since cooperative capacity can be utilized more effectively. This suggests that in real-world applications with lower demand-to-capacity ratios, cooperative savings may be even larger.

2.6 Policy implications

Based on the results of our numerical experiments, we derive the following policy implications that are relevant for LSPs as well as for policymakers.

Cooperation leads to monetary and environmental cost savings: Our results show that cooperation leads to huge improvements in monetary as well as non-monetary KPIs. These result from more efficient use of services, higher utilization of urban vehicles, and economies of scale through cooperation. It is particularly important that cooperation is not limited to the use of shared infrastructure but also includes demand sharing, as this can lead to considerable cost savings even when established on a small scale.

Balance fairness over time: Our numerical experiments consistently demonstrate that adhering to fairness criteria on average over an extended period of several days yields significant cost savings compared to enforcing them on a daily basis. Even an extension of the constraints to a segment of two days enables balancing effects that mitigate the impact of the constraints on the total system cost. Although these balancing effects reduce overall costs, they lead to considerable daily fluctuations in workload and cost distributions for LSPs, particularly for the smaller LSPs in the coalition. This holds for both workload and cost-related fairness criteria.

Mitigate the impact of too strict constraints: We show that too strict constraints increase the total system cost and thus harm all LSPs of the coalition. Fairness constraints should, therefore, not be too strict to reduce negative effects on the coalition as a whole. Across all experiments, moderate settings of these constraints have little to almost no impact on the total system cost.

Consider cost and market shares: Focusing solely on costs or solely on market shares does not lead to acceptable coalitions for all LSPs. In order to ensure a stable coalition, both aspects must therefore be taken into account in the decision support system so that none of the LSPs is disadvantaged in terms of cost and workload aspects.

Allow flexibility over time in the selection of services: We demonstrate that cooperative city logistics systems need flexibility in selecting services over time. Regularly operated services might be wishful, but too strict constraints on this lead to a dramatic increase in total system cost and therefore harm the entire system as well. Especially when there are high fluctuations in the daily demand structure, flexibility is necessary as otherwise there will be immense additional costs as the selection of services can no longer be adapted to the specific demand structures of each day. Therefore, service regularity relies on demand regularity to prevent significant increases in total system costs. In this context, too, the majority of cost increases occur when flexibility is severely restricted. Conversely, a moderate restriction, where some services have to be operated regularly, only leads to a slight increase in costs.

2.7 Conclusion

In the existing body of literature on city logistics, numerous models and problem settings have been explored that implicitly assume a cooperative logistics system. However, our study is the first to explicitly address the area of cooperation between LSPs in a more realistic multi-day scenario, considering both fairness and service flexibility constraints.

Throughout our investigation, we analyzed the impact of different constraints regarding the fair design of a cooperative city logistics system. The crux of our findings underscores the importance of integrating both cost-based and workload-based constraints for the establishment of fair and sustainable cooperations in city logistics. Our numerical experiments clearly revealed that fairness should be considered on average over a horizon of several days rather than on a daily basis. Furthermore, it became evident from our results that excessively stringent constraints not only compromise total costs but also negatively impact all LSPs engaged in the coalition. Based on these results, we derived policy implications that are of great importance from the perspective of both the LSPs and the public policymakers in order to realize successful city logistics projects.

Looking forward, our study opens up several avenues for future research. In particular, there is considerable potential for extending the scope beyond financial and workload equity, exploring aspects such as joint risk-sharing mechanisms or reciprocal compensation when an LSP serves another's demand. Future research could also examine cooperative infrastructure planning for resilient transportation networks (Zhang et al. 2023), as collaborative strategies may enhance both resilience and efficiency in the face of disruptive events. Further, cooperative and fairness aspects at the operational level could be explored, such as the impact on routing decisions, travel times, and operational costs. Beyond the cooperative aspects, the city logistics literature still lacks comprehensive real-world instance datasets that could significantly advance research and practical applications in the field.

In conclusion, we encourage further research in the domain of cooperative city logistics. By addressing these challenges and offering actionable recommendations to both LSPs and public policymakers, we can contribute to more sustainable, efficient, and ultimately more livable cities.

Acknowledgments

We thank the anonymous reviewers for their valuable recommendations, which have significantly improved our paper. The paper was written as part of a project funded by the German Research Foundation. We would like to express our gratitude to the German Research Foundation (project number: ROCOCO FO1468/1-1) for the financial support of our research project. Additionally, our gratitude goes to the Bavarian Research Alliance for their support of the research stay in Montreal, which contributed significantly to the development of this paper. While working on this paper, the second author was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréal, Canada. We gratefully acknowledge the financial support provided by the Canadian Natural Sciences and Engineering Research Council (NSERC) through its Discovery grant program, as well as by the Fonds de recherche du Québec through their Teams and CIRRELT infrastructure grants.

Contribution 3 - Fast Shapley value approximation through machine learning with application in routing problems

3 Fast Shapley value approximation through machine learning with application in routing problems

Johannes Gückel, Pirmin Fontaine

Abstract For many routing applications, it is not only necessary to minimize total costs, but also to allocate them to individual customers. In this context, the allocation according to the Shapley value is a well-known method highly regarded for its fulfillment of major fairness criteria. However, its computational complexity restricts its applicability, especially for NP-hard underlying optimization problems like routing problems, since the exact computability of the Shapley values is no longer possible within a reasonable time.

In this study, we propose a general Machine Learning-based Shapley Value Approximator (MLSVA) and apply it to routing problems based on the exploitation of routing-specific problem structures as features, enabling real-time approximations.

Within an extensive numerical study, we show that our MLSVA outperforms current state-of-the-art approximation results for the Traveling Salesman Problem (TSP) and, at the same time, achieves very good results for the Capacitated Vehicle Routing Problem (CVRP), for which it is the first efficient method that quickly delivers high-quality approximations. On average, we approximate Shapley values with a mean absolute percentage error of 2.4% for the TSP and 3.5% for the CVRP. Further, the MLSVA achieves strong approximation results even when trained on biased labels. This shows the scalability of the MLSVA and allows high-quality approximations even for large routing problems. Finally, we demonstrate the generalizability of our approach by successfully approximating Shapley values for items in a variant of the bin packing problem.

Published: Networks 86 (4)
URL: https://doi.org/10.1002/net.70003

3.1 Introduction

While the literature offers many exact and heuristic methods for solving diverse route planning problems to determine optimal routes and total cost (e.g., Elshaer and Awad 2020, Costa et al. 2019), there has been relatively scarce research on allocating the resulting cost to individual customers. Allocation methods are not only used for monetary cost allocation but also for the allocation of environmental emissions, which is likely to become increasingly relevant in the future (Kellner and Schneiderbauer 2019).

The problem of cost allocation in vehicle routing was introduced several decades ago in the context of cooperative game theory (Göthe-Lundgren et al. 1996). Engevall et al. (2004) explored the application of game-theoretic allocation methods, including the Shapley value, in routing problems with heterogeneous vehicle fleets, though computational limits restricted exact calculations to small instances. Similarly, Sun et al. (2015) considered the Shapley value for cost allocation on fixed routes. Özener et al. (2013) studied an inventory routing application using an approximation approach based on random subset sampling to circumvent computational challenges. Further real-life applications include the work by Engevall et al. (1998), who applied the Shapley value to allocate costs among customers of a gas and oil company, highlighting both practical relevance and computational limitations. These studies collectively underline the necessity for efficient approximation methods capable of delivering accurate Shapley value estimates for larger problem instances. Although some cooperative game settings allow efficient computation of Shapley values, this is not the case for routing problems. In such settings, the main computational challenge lies not in the Shapley formula itself, but in evaluating the cost of each subcoalition, which requires solving a complex optimization problem. Due to this combinatorial nature, exact Shapley value computation becomes intractable even for moderately sized instances.

Inspired by concepts from cooperative game theory, the Shapley value has emerged as a method that ensures fair cost allocation to individual customers. Despite its alignment with various fairness criteria, the practical application of the Shapley value in routing problems remains limited to instances with only a small number of customers. This limitation stems from the need to solve the routing problem for every subcoalition of customers to compute the Shapley value exactly, resulting in significant computational expenses. Therefore, there is a need for approximation methods that can provide fast and accurate estimates of the Shapley value while remaining computationally efficient.

Previous approximation methods for the Shapley value can be categorized into two types. The first type includes statistical sampling methods (e.g., Touati et al. 2021) that are computationally intensive and do not exploit the underlying problem structure of routing problems. The second type involves approximations explicitly designed for the Traveling Salesman Problem (TSP) (e.g., Popescu and Kilby 2020), and is not adaptable to other routing problems.

In this study, we propose a general machine learning-based approach to approximate Shapley values and demonstrate its application to the TSP and the Capacitated Vehicle Routing Problem (CVRP). Through this approach, we exploit the particular problem structure of the underlying routing problem and create problem-specific features on which the models can learn and then approximate Shapley values for unseen problem instances rapidly. Although Shapley value approximations can generally only be tested on small instances, as exact Shapley values can only be computed for instances with a limited number of nodes, we propose methods to effectively train our approach for larger instances as well. These methods include training on quickly generated but biased Shapley values, as well as training on smaller instances than those used for later application. Furthermore, we can provide managerial insights about the influential factors determining each customer's cost share. As a result, we contribute to the literature in five ways:

1. We propose a new machine learning-based approach for approximating Shapley values, which utilizes both routing problem-specific features and the efficiency property of the Shapley value.

2. We show options that can be used to create quickly generated but biased labels. This is an important feature, especially for larger instances, as exact labels may no longer be created there.

3. Through an extensive numerical study, we demonstrate the performance of our method compared to simple proxies and current state-of-the-art approximation methods for the TSP and CVRP. Further, we show that this machine learning-based approach also leads to good approximations even with biased labels. As a result, we are the first efficient method that delivers high-quality solutions not only for the TSP but also for the CVRP.

4. Unlike other methods, our approach provides managerial insights into the influential factors on the Shapley value, which is highly relevant for logistics service providers.

5. We show that the methodology that we propose can be generalized by applying it to a bin packing problem.

The structure of this study is organized as follows: Section 3.2 details the problem description by presenting the problem of cost allocation, the Shapley value and its properties as well as the necessity of an approximation method for Shapley values. Section 3.3 provides a comprehensive literature review on applications and approximations of Shapley values as well as on machine learning applications in routing problems. Section 3.4 introduces our machine learning-based approximation approach and discusses the features, the machine learning models, and the generation of biased labels. Section 3.5 presents an extensive numerical study on the performance of our approach and analyzes the impact of biased labels in the training process, which is critical for larger instances where exact labeling is computationally prohibitive. Section 3.6 provides managerial insights by analyzing the feature's importance. Section 3.7 demonstrates the generalizability of our approach by applying it to the allocation of an item's cost in a bin packing variant. Section 3.8 concludes our study and provides further research directions.

3.2 Problem description

In this section, we first present the general problem description of allocating cost to customers in the context of routing problems (Section 3.2.1). Then, we discuss the Shapley value and its

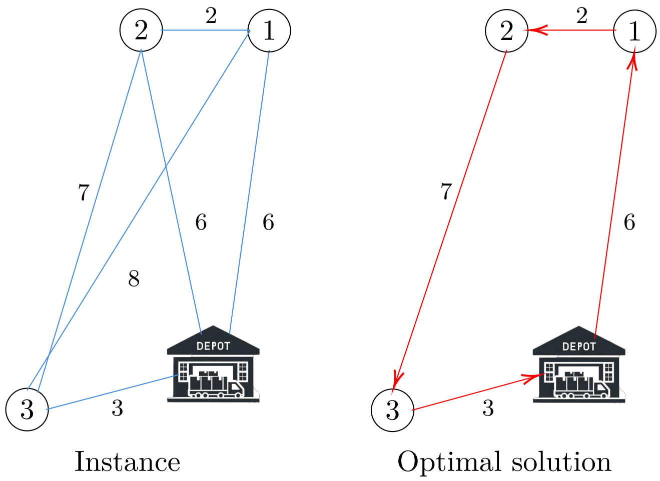

Figure 3.1: TSP instance solved

properties as well as its computational limitations and highlight the importance of approximated Shapley values (Section 3.2.2).

3.2.1 Allocating cost to customers in routing problems

Cost allocation in routing problems presents a significant challenge. The general problem is explained using the example of a TSP instance shown in Figure 3.1. Nevertheless, the resulting implications also hold for other routing problems.

Let $\mathcal{N}$ be the set of all customers in a TSP instance. $C(\mathcal{N})$ denotes the total cost of the optimal TSP solution. The goal is to determine a cost allocation ϕ where each customer $n \in \mathcal{N}$ is assigned a cost ϕ_n. We also include a depot node 0, which incurs no cost. Therefore, the total cost $C(\mathcal{N})$ must be fully allocated to the customers, i.e., $C(\mathcal{N}) = \sum_{n \in \mathcal{N}} \phi_n$, which in the example in Figure 3.1 results in $\phi_1 + \phi_2 + \phi_3 = 18$. Even in this simple example instance, it is not straightforward to determine which customer is responsible for which share of the total cost. For instance, if the cost were distributed depending on the distance to the depot, customer 3 would bear significantly less than customers 1 and 2. However, this might be perceived as unfair because customers 1 and 2 only incur low marginal cost due to their geographical proximity. As an alternative, the cost could be allocated in proportion to the marginal cost $C(\mathcal{N}) - C(\mathcal{N} \setminus \{n\})$ of each customer n. This might be unfair to customer 3 because he is much closer to the depot.

To address fairness concerns, cooperative game theory has introduced a variety of more sophisticated cost allocation methods that are particularly applicable to such allocation scenarios. These include core allocation (Gillies 1959), the Equal Profit Method (Frisk et al. 2010), and the nucleolus (Schmeidler 1969). For a comprehensive review of cost allocation methods in transportation problems, we refer to Guajardo and Rönnqvist (2016). Each of these approaches has distinct characteristics and emphasizes different aspects of fairness. Among these, the Shapley value, which will be discussed in detail in the following section, stands out as the most widely used allocation method due to its unique and well-established properties.

3.2.2 Shapley value cost allocation

The Shapley value was first introduced by Shapley (1953). Since then, it has been applied in many different contexts. Formally, the Shapley value is defined as follows:

$$\phi_n^{SV} = \sum_{\mathcal{S} \subsetneq \mathcal{N} \setminus \{n\}} \frac{|\mathcal{S}|!(|\mathcal{N}| - |\mathcal{S}| - 1)!}{|\mathcal{N}|!} [C(\mathcal{S} \cup \{n\}) - C(\mathcal{S})] \tag{3.1}$$

In this Equation (3.1), the sum is over all subcoalitions $\mathcal{S}$ within the set of all customers $\mathcal{N}$ that do not include customer n. The term $[C(\mathcal{S} \cup \{n\}) - C(\mathcal{S})]$ represents the marginal cost of the customer to the subcoalition S. The fraction $\frac{|\mathcal{S}|!(|\mathcal{N}|-|\mathcal{S}|-1)!}{|\mathcal{N}|!}$ represents the weighting of subcoalitions within all possible permutations. Additionally, especially important for the understanding of sampling approximation methods, the formula can be expressed as an average over all possible permutations, as shown in Equation (3.2) (Mitchell et al. 2022):

$$\phi_n^{SV} = \frac{1}{|\mathcal{N}|!} \sum_{p \in \mathcal{P}} (C(p_{n-1} \cup \{n\}) - C(p_{n-1})) \tag{3.2}$$

Thereby, $\mathcal{P}$ represents the set of all permutations, and p_{n-1} represents the set of customers ranked lower than customer n in the order of permutation p. Consequently, the Shapley value allocates cost according to each customer's average marginal cost to each possible permutation. The cost allocation of the Shapley value is unique due to the fulfillment of the following properties:

- **Efficiency:** $\sum_{n \in \mathcal{N}} \phi_n^{SV} = C(\mathcal{N})$. The sum of the Shapley values is equivalent to the total cost.
- **Symmetries:** If $C(\mathcal{S} \cup \{n\}) = C(\mathcal{S} \cup \{\tilde{n}\}) \forall S \subseteq \mathcal{N} \setminus \{n, \tilde{n}\}$, then $\phi_n^{SV} = \phi_{\tilde{n}}^{SV}$. Two customers with the same marginal cost have the same Shapley value.
- **Additivity:** For any two games C and C', the Shapley value satisfies $\phi_n^{SV}(C+C') = \phi_n^{SV}(C) + \phi_n^{SV}(C')$. That is, the Shapley value of the sum of two games equals the sum of their individual Shapley values.
- **Dummy:** If $C(\mathcal{S} \cup \{n\}) = C(\mathcal{S}) \forall \mathcal{S} \subseteq \mathcal{N} \setminus \{n\}$, then $\phi_n^{SV} = 0$. A customer that has no marginal impacts has a Shapley value of zero.

In addition to these four most frequently mentioned properties, the Shapley value also fulfills the property of monotonicity, according to which, if a game changes in a way that increases or maintains a player's contribution to all coalitions, their allocated cost should not decrease (Young 1985). Further, the Shapley value also satisfies the balanced contribution property (Myerson 1980), which ensures that for any pair of players, one player's claim against another is always balanced by a corresponding counterclaim from the other player. Consequently, the Shapley value embodies numerous well-established properties of cooperative game theory, each emphasizing different aspects of fairness in cost allocation. As a drawback, it is important to note that, unlike other cost allocation methods such as the nucleolus (Schmeidler 1969) or the core allocation (Gillies 1959), the Shapley value does not necessarily lie within the core of a cooperative game. This means that there may

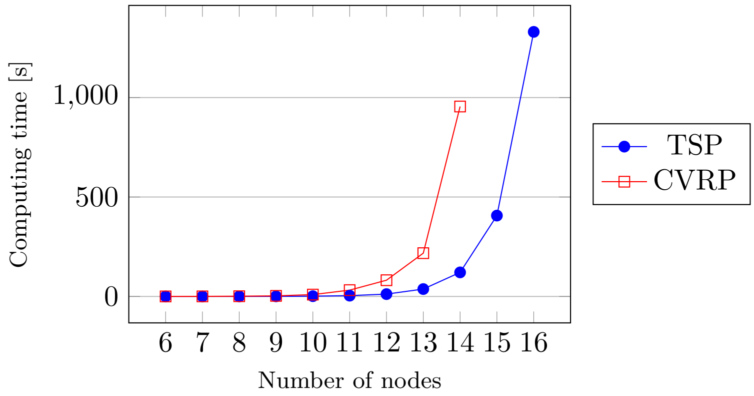

Figure 3.2: Computing time [s] per number of nodes required to compute a single instance of the Shapley value (averaged over ten randomly generated instances with the numerical setup described in Section 3.5)

be cases where a subcoalition can find a more favorable cost allocation, violating the rationality property, which ensures that no subcoalition has an incentive to deviate from the allocation (Shapley 1971). Computing the Shapley value requires evaluating every possible subcoalition, represented by the characteristic function $C(\cdot) : 2^{\mathcal{N}} \rightarrow \mathbb{R}$. The characteristic function assigns a value to each subset $\mathcal{S} \subseteq \mathcal{N}$, which in this context represents the routing cost of serving the customers in $\mathcal{S}$. For a problem instance with $|\mathcal{N}|$ customers, this requires solving $2^{|\mathcal{N}|}$ routing problems. For 15 customers, this already amounts to $2^{15} = 32,768$ problems. Figure 3.2 illustrates the computing time for a single instance as a function of the number of nodes. These times include the exact solution of the corresponding routing problem for every subcoalition required to compute the Shapley value. It shows that the computing time exponentially increases with the number of nodes. This computational burden highlights that there is a need for efficient approximation methods for Shapley values within a short computational time.

3.3 Literature review

This section is divided into four parts. First, we review the existing literature on the applications of the Shapley value (Section 3.3.1). Then, we examine the literature on approximations of the Shapley value (Section 3.3.2). Further, we review the current state of the literature on machine learning in routing and transportation (Section 3.3.3). We conclude by outlining research gaps (Section 3.3.4).

3.3.1 Shapley value applications

The Shapley value was initially introduced in the literature on cooperative game theory for allocating costs to players of a cooperative game (Shapley 1953). Since then, it has been used in many different contexts to allocate total costs. Beyond allocating costs to customers in routing problems, which has been discussed in the introduction, the Shapley value has also been applied to many other applications in the transportation sector. For instance, it can be used to allocate cost to carriers in cooperative routing problems. In this context, Krajewska et al. (2008) used the Shapley value to distribute profits to cooperating carriers. While allocating costs and distributing profits are conceptually distinct, both processes can be modeled using the Shapley value to fairly allocate the total amount based on each participant's contribution. In another application, Kimms and Kozeletskyi (2016b) employed the Shapley value for cost allocation in the context of a TSP with a rolling horizon and cooperative salesmen. Additionally, Gückel et al. (2024) studied cost allocation among cooperative logistics service providers in a city logistics context, where they compared the Shapley value with other cost allocation methods. Further, the Shapley value has been used as a cost allocation method for distributing highway construction and maintenance costs among different user groups (Gómez-Rodríguez et al. 2024, Kuipers et al. 2013).

Besides its use in transportation and routing, the Shapley value has been applied in a variety of other domains. For instance, in cooperative queueing games, Bendel and Haviv (2018) demonstrated the Shapley value's utility for allocating costs when servers pool resources to optimize service rates. Similarly, Timmer et al. (2013) applied the Shapley value in the context of inventory management among cooperating companies, ensuring that the cost allocation incentivizes collaboration. In the realm of peer-to-peer networks, Altan and Özener (2019) proposed an approximation of the Shapley value to achieve fair cost allocation between service providers and users working together to minimize joint costs. Moreover, Zolezzi and Rudnick (2002) utilized the Shapley value to design a cooperative cost allocation method for transmission systems in the energy sector. Additionally, Choudhury et al. (2021) introduced the Generalized Egalitarian Shapley value, which provides more flexibility in determining the level of marginality based on coalition size.

Beyond cost allocation, the Shapley value has found widespread applications in other fields. In machine learning, it has become a widely used tool for assessing feature importance in model performance (e.g., Cohen et al. 2007, Lundberg and Lee 2017, Borgonovo et al. 2024). Additionally, it has been used in sports analytics to evaluate player importance in basketball (Metulini and Gnecco 2023). In healthcare, Roshanaei et al. (2017) used the Shapley value to distribute cost savings from cooperating hospitals. Gnecco et al. (2019) applied the Shapley value to measure the contribution of each node in cooperative transportation network games to assess centrality. Notably, the Shapley value has also been investigated in the context of environmental cost allocation, as seen in studies such as Ciardiello et al. (2020) and Kellner and Schneiderbauer (2019).

Despite its broad applicability, the Shapley value is seldom used in applications involving NP-hard problems with a large number of participants due to significant computational limitations.

3.3.2 Shapley value approximations

To date, the approximation methods found in the literature can be divided into two categories. The first category includes approximations that are independent of the underlying subproblem and that rely solely on the properties of the Shapley value. This category is represented by statistical sampling methods that aim to approximate the Shapley value through selective sampling of permutations or sub-coalitions while reducing computing time. The most straightforward sampling method is the Monte Carlo sampling introduced by Shapley and Mann (1960) for electoral college voting games. In this sampling method, random permutations are sampled and evaluated until a predefined number is reached. Then, the average marginal contribution of each participant over all sampled permutations is determined and serves as the approximation. Schopka and Kopfer (2017) used a so-called α-sampling procedure to approximate Shapley values in horizontal carrier coalitions. Horizontal carrier coalitions involve collaboration between competing carriers at the same level of the supply chain to reduce costs through joint operations, in contrast to vertical cooperation, which occurs between different supply chain levels (e.g., carriers and suppliers). Unlike our work, where costs are allocated to customers, the cost allocation here occurs at the carrier level. This procedure does not sample random permutations but only subcoalitions with a small number α of carriers, which leads to a much lower computing time. Touati et al. (2021) recently showed a new approach by combining Monte Carlo sampling with Bayesian methods. Further, Castro et al. (2009) introduced a polynomial calculation approach for the Shapley value on the basis of sampling. However, the limitation here is that the value of each coalition in the subproblem must be computed polynomially. With the goal of attributing predictions to features in a machine learning context, Mitchell et al. (2022) introduced a new approximation method of Shapley values on the basis of a specific selection of permutation samples.

The second category includes problem-specific approximation methods that explicitly exploit properties of the underlying routing problem. Aziz et al. (2016) investigated several proxies for Shapley values in TSPs. Both simple rules of thumb, such as distance to depot or distribution based on marginal cost, and more advanced methods, such as linear regression or sampling methods, were used as proxies. In their research outlook, it became clear that approximation methods that also perform on more general vehicle routing problems are needed. Popescu and Kilby (2020) presented two approximation approaches for the Euclidean TSP, which build on an extension of the one-dimensional TSP. These approximations are on the basis of shared distances, where the cost associated with any two locations is based on the Euclidean distance between their coordinates and the depot. These require only polynomial computing time. They showed that they outperform all other proxies from the literature and are only outperformed by sampling methods with very high iteration numbers and computing time. The current state-of-the-art method is from Levinger et al. (2021). Their approximation method, SHAPO, is based on a fixed order of customers. However, they showed that this approximation method can be generalized by considering the customer sequence of a TSP as a fixed order. In a numerical study, they achieved better results than the previous state-of-the-art method by Popescu and Kilby (2020).

3.3.3 Machine learning in routing and transportation

Machine learning methods have been used in the fields of routing and transportation for prediction or decision support in various areas. Li et al. (2019) used decision tree ensemble methods for the travel time prediction. Gmira et al. (2020) predicted the travel speed to get from one location to another in a road network with the use of different machine learning and data mining methods. Basso et al. (2021) proposed a probabilistic Bayesian machine learning approach for predicting the expected energy consumption and variance in the electric vehicle routing problem.

In addition to actual predictions, machine learning models are also used to support decision-making or adapt policies. Ghiani et al. (2022) used a neural network to adjust online policy parameters in a pickup and delivery problem learned offline by using simulation experiments. Van der Hagen et al. (2024) used supervised machine learning approaches to support time slot decisions for the feasibility check to offer time slots to passengers in an online booking system. In the context of hierarchical vehicle routing problems, Sobhanan et al. (2024) integrated a neural network for predicting the objective function value of the underlying vehicle routing problem without actually solving it into a genetic algorithm. Another possible application is the reduction of problems. For example, by using a Support Vector Machine in the TSP, Sun et al. (2021) predicted which decision variables could be removed from the problem without significantly affecting the quality of the solution. Bertsimas and Kim (2023) developed a machine learning approach to accelerate the solution process for the mixed-integer convex optimization problem and applied it to problems from the field of transport optimization. In the context of load planning for rail transportation, Larsen et al. (2022) developed a two-stage stochastic programming model, where the expected solution of the second-stage problem is predicted based on the first-stage variables. Further, machine learning models have been successfully applied in heuristics (e.g., Bruni et al. 2023, Ermağan et al. 2022) and exact solution methods (e.g., Zhang et al. 2022). In recent years, there has been a growing number of contributions showcasing new machine learning-based methodologies applied conceptually to fundamental routing problems. Asín-Achá et al. (2024) demonstrated how machine learning models can be used to select a best performing fast algorithm and parameterize it at the same time by exploiting routing-specific features for the CVRP. Morabit et al. (2024) proposed a machine learning-based heuristic for the reoptimization of repeatedly solved routing problems after minor changes in the problem data. Chamurally and Rieck (2024) integrated a machine learning approach into an adaptive large neighborhood solution method and used historical data to predict customer demands for a vehicle routing problem with stochastic demands. For a literature review about machine learning methods in a vehicle routing context, we refer to Bai et al. (2023). For a more general review of the use of machine learning in combinatorial optimization, we refer to Bengio et al. (2021). Overall, machine learning is already used to predict outcomes and enhance solution approaches by leveraging problem-specific features, making it a suitable choice for the case at hand.

3.3.4 Research gaps

While the Shapley value is well-regarded for cost allocation in literature, its resource-intensive computations usually limit its application to instances with a small number of customers.

The literature lacks a general approximation method that can be applied to different routing problems and utilizes the specific problem components in each case. Machine learning models seem ideally suited for this problem setting as there are a lot of influential factors on the Shapley value that have not been explored in any of the approximation methods before. In addition, none of the current approximation methods provides managerial insights about the main cost drivers for customers in routing problems. We fill this research gap with our machine learning-based approximation approach.

3.4 Methodology

This section starts describing the general outline of our Machine Learning-based Shapley Value Approximator (MLSVA) (Section 3.4.1). Then, we describe our methodology step by step. We start with the machine learning models (Section 3.4.2). Next, we continue with the feature generation (Section 3.4.3). After that, we show the scaling of our approximations to ensure the efficiency property (Section 3.4.4). Finally, we describe how to create fast but biased labels to make our approach feasible for larger instances, since exact labels for them cannot be created in acceptable time (Section 3.4.5).

3.4.1 General outline

Our MLSVA is based on supervised machine learning, utilizing true Shapley values and influential features to learn and predict Shapley values for unseen instances and customers. An instance refers to a specific realization of a routing problem (TSP, CVRP, etc.), characterized by a given number of customers, their locations, and other relevant parameters. The primary distinction between instances lies in their unique combination of problem-specific parameters, such as the number of customers, their positions, and additional factors like customer demands or vehicle capacity. Let $\mathcal{T}^{all}$ denote the set of all generated instances, which serve as the basis for training and testing our models, $\mathcal{N}(t)$ represent the set of customers in instance t, $\hat{\phi}_n^{ML}$ denote the original predictions of our machine learning models, and $\hat{\phi}_n$ denote our final approximation of the Shapley value for customer n. The cost function $C(\cdot)$ defines the total routing cost for any subset of customers within an instance, i.e., it represents the characteristic function of the cost game. For example, $C(S)$ denotes the cost of the optimal routing solution for a coalition $S \subseteq \mathcal{N}(t)$, and $C(\mathcal{N}(t))$ corresponds to the total cost of serving all customers in instance t. Thereby, no predefined customer order is assumed, as costs are based on (optimal) routing solutions. Using this notation, our general methodology is outlined in Algorithm 4.

We begin by generating a set of instances, denoted as $\mathcal{T}^{\text{all}}$. For each instance t, the first step is to compute the characteristic function, i.e., to evaluate the routing cost for all subcoalitions of customers within the instance. This allows us to assign the true Shapley value ϕ_n^{SV} to each customer observation n, and to prepare the corresponding feature vector $f(n)$ that captures the factors influencing the Shapley value (see lines 2–6). These features characterize both the individual customer as well as the instance in which the customer is present, thereby exploiting the specific routing problem structure. It is important to note that, while predictions are made for individual customers, instance-specific data and costs are crucial for generating the features and scaling the predictions.

We then split the instances into training instances $\mathcal{T}^{train}$ and test instances $\mathcal{T}^{test}$ (line 7). For the training instances, we use the Shapley values as labels (line 8). The selected supervised machine learning models are trained exclusively on the training instances (line 9) based on the features and labels. These trained models are then used to generate predicted Shapley values $\hat{\phi}_n^{ML}$ for each observation (lines 10 to 12). Next, these predictions are scaled for each instance so that the sum of the approximated Shapley values corresponds to the total routing costs for that instance, producing the final Shapley value approximations $\hat{\phi}_n$ (lines 13 to 15). This scaling is a critical aspect of our methodology, as the sum of the approximations for all customers in an instance must match the total routing costs that are known in advance (see efficiency property in Section 3.2).

In the end, the performance of the method is evaluated exclusively on the test data $\mathcal{T}^{test}$ and the best performing machine learning model can be selected. Once the machine learning models are trained, our approximation approach can generate approximations rapidly without high computational effort.

Algorithm 4: MLSVA

1 **Generate** set of instances $\mathcal{T}^{all}$;
2 **for** $t \in \mathcal{T}^{all}$ **do**
3     $C(\cdot) \leftarrow$ **determineCharacteristicFunction**$(\mathcal{N}(t))$;
4   **for** $n \in \mathcal{N}(t)$ **do**
5         $\phi_n^{SV} \leftarrow$ **computeShapley**$(n, C(\cdot))$;
6         $f(n) \leftarrow$ **prepareFeatures**(n, t); `// Section 3.4.3`
7 **Separate** $\mathcal{T}^{all}$ **into** $\mathcal{T}^{train}$ **and** $\mathcal{T}^{test}$;
8 Use ϕ_n^{SV} as label $\forall n \in \mathcal{N}(t),\ t \in \mathcal{T}^{train}$;
9 **trainMlModels**$(f(n), \phi_n^{SV}\ \forall n \in \mathcal{N}(t),\ t \in \mathcal{T}^{train})$; `// Section 3.4.2`
10 **for** $t \in \mathcal{T}^{test}$ **do**
11   **for** $n \in \mathcal{N}(t)$ **do**
12         $\hat{\phi}_n^{ML} \leftarrow$ **predictShapley**$(f(n), \phi_n^{SV})$;
13 **for** $t \in \mathcal{T}^{test}$ **do**
14   **for** $n \in \mathcal{N}(t)$ **do**
15         $\hat{\phi}_n \leftarrow$ **scalePrediction**$(\hat{\phi}_n^{ML}, C(\mathcal{N}(t)))$; `// Section 3.4.4`

3.4.2 Machine learning models

In this framework, we use machine learning models that are able to provide an approximation for a numerical value. This is supervised machine learning as we use labeled observations (Hastie et al. 2009).

Neural Networks: Neural networks (NN) are especially known for recognizing highly non-linear relationships and can generate numerical, categorical, and binary outputs (Hastie et al. 2009). A NN essentially consist of a sequence of layers in which data is transferred non-linearly to the layer behind. Each layer's output represents the next layer's input in the sequence. Each layer consists of several neurons that apply transformations to the input data (LeCun et al. 2015). These transformations are controlled by learnable parameters and activation functions determining how the input signal is modified. The sequence's last layer determines the final output. In our study, we use a feedforward architecture in which each layer receives input from the previous layer.

Random Forest: Random Forest (RF) is an ensemble learning method in which a large collection of de-correlated trees is formed, the results of which are then averaged. The basic idea behind it is that aggregating multiple different decision trees reduces the variance of the approximation. This technique can be used for both classification and regression of numerical data (Hastie et al. 2009). As the number of trees in a forest increases, the generalization error converges (Breiman 2001).

XGBoost: XGBoost is short for eXtreme Gradient Boosting (XGB). In contrast to the RF, new trees are created sequentially, which learn from the approximation errors of the previous trees and thus generate better approximations (Hastie et al. 2009).

Polynomial regression: Polynomial regression (PR) extends linear regression by incorporating polynomial terms, capturing complex, non-linear relationships between variables (Weisberg 2005). A PR model of degree d is given by Equation (3.3):

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_d X^d, \tag{3.3}$$

where $\beta_0, \beta_1, \ldots, \beta_d$ are regression coefficients. The degree d controls model flexibility, with $d = 2$ for quadratic and $d = 3$ for cubic regression. Higher degrees enable more complex curves but risk overfitting.

Support Vector Regression: Support Vector Regression (SVR) extends the concepts of Support Vector Machines to regression problems by fitting a hyperplane in a high-dimensional space that best approximates the given training data within a specified margin (Drucker et al. 1996).

k-nearest-neighbor: k-nearest-neighbor (KNN) is a straightforward machine learning algorithm that is used for both classification and regression. We include this algorithm to have a simple machine learning method that serves as a benchmark for the more advanced methods introduced before. The basic idea is to approximate the target variable based on the k nearest neighbors. Since our use case is a regression with numerical output, the approximation is determined by the mean value of the k nearest neighbors. Let $\mathcal{K}(n)$ be the set of k nearest neighbors of customer n; the approximation is defined with Equation (3.4).

$$\hat{\phi}_n^{KNN} = \frac{\sum_{n' \in \mathcal{K}(n)} \phi_{n'}^{SV}}{|\mathcal{K}(n)|} \tag{3.4}$$

The hyperparameter $k = |\mathcal{K}(n)|$ must be tuned in advance.

The selection of machine learning models in this study covers different highly recognized algorithms, each representing distinct paradigms. RF and XGB showcase the strengths of tree-based ensemble techniques through bagging and boosting, respectively. PR captures non-linear relationships effectively, while an NN manages complex data structures via deep learning. SVR extends Support Vector Machines principles to high-dimensional regression tasks. KNN is included as the simplest machine learning model, serving as a baseline for comparing the performance of more advanced approaches. Together, these models provide a thorough evaluation of diverse approaches, enhancing the robustness and applicability of our results.

3.4.3 Feature generation

To accurately approximate the Shapley value using regression models, we design features that capture both individual customer characteristics and their interactions within the instance. These features are grouped into four categories: (1) **Distance-based features**, describing spatial relationships that impact cost distribution; (2) **Instance-based features**, summarizing global structural properties of the problem instance; (3) **Cost-based features**, incorporating marginal and local cost effects that directly influence allocation; and (4) **Cluster-based features**, reflecting structural dependencies among customers that impact cost allocation. Additionally, for the CVRP, we introduce **CVRP-specific features**, which account for demand distributions and vehicle capacity constraints, both of which play a crucial role in shaping cost allocation. The clusters were created using the DBSCAN algorithm from Ester et al. (1996). This algorithm is based on a density-based notion of clusters, which identifies clusters as areas of high density separated by areas of low density. Further, only for the CVRP instances, we use CVRP-specific features. Please note that we often use features in relation to the mean characteristics of the other customers of an instance, as the absolute value of a feature is often not very meaningful, but rather, the ratio to the other feature characteristics of the customers of an instance is relevant. This is indicated with (r). The distance between two nodes $i, j \in \mathcal{N}$ is denoted as d_{ij}, which refers to the distance between the nodes in the original TSP instance, as specified in the distance matrix. This distance remains fixed regardless of the solution to the instance. A detailed overview of the features for a customer n, including their mathematical expressions where applicable, is provided in Table 3.1.

Table 3.1: Feature description

Feature name	**Description**	**Expression**
Distance-based features		
Distance to Depot (r)	The distance of the customer to the depot node 0	$\frac{d_{0n}}{\frac{1}{\lvert\mathcal{N}\rvert}\sum_{\hat{n}\in\mathcal{N}} d_{0\hat{n}}}$
Distance to m closest customer (r)	Distance to the m closest customer. We include this feature for $m = 1, .., 5$	$\frac{d_{nm}}{\frac{1}{\lvert\mathcal{N}\rvert}\sum\limits_{\hat{n}\in\mathcal{N}} d_{\hat{n}m}}$
Distance to gravity center (r)	Distance of the customer to the gravity center g of all customers	$\frac{d_{ng}}{\frac{1}{\lvert\mathcal{N}\rvert}\sum\limits_{\hat{n}\in\mathcal{N}} d_{\hat{n}g}}$
Mean distance to other customers (r)	Mean distance of the customer to other customers	$\frac{\frac{1}{\lvert\mathcal{N}\rvert-1}\sum\limits_{i\in\mathcal{N}\setminus\{n\}} d_{ni}}{\frac{1}{\lvert\mathcal{N}\rvert}(\sum\limits_{\hat{n}\in\mathcal{N}}(\frac{1}{\lvert\mathcal{N}\rvert-1}\sum\limits_{j\in\mathcal{N}\setminus\{\hat{n}\}} d_{\hat{n}j}))}$
Instance-based features		
Number of customers	Number of customers in the instance	$\lvert\mathcal{N}\rvert$
Total cost	Total cost of the (optimal) solved routing problem	$C(\mathcal{N})$
Y/X-coordinate	The X and Y coordinates of the customer location	-
Y/X-coordinate depot	The X and Y coordinates of the depot location	-
Descriptive statistics	Descriptive statistics of the coordinates. We consider the min, max, mean, standard deviation, and the skewness of X and Y coordinates	-
Cost-based-features		
Marginal costs (r)	Marginal costs of a customer	$\frac{C(\mathcal{N})-C(\mathcal{N}\setminus\{n\})}{\frac{1}{\lvert\mathcal{N}\rvert}\sum\limits_{\hat{n}\in\mathcal{N}}(C(\mathcal{N})-C(\mathcal{N}\setminus\{\hat{n}\}))}$
Shortcut cost (r)	Cost savings when skipping a customer in the given optimal tour by directly connecting its predecessor i and successor j. Unlike marginal cost, this only accounts for local cost changes without re-optimizing the solution	$\frac{d_{in}+d_{nj}-d_{ij}}{\frac{1}{\lvert\mathcal{N}\rvert}\sum\limits_{\hat{n}\in\mathcal{N}}(d_{i\hat{n}}+d_{\hat{n}j}-d_{ij})}$
Cluster-based-features		
Number of clusters	The number of different clusters including outliers as own cluster after application of DBSCAN	-
Number of customers in cluster	The number of customers in the same cluster	-

Continued on next page

Table 3.1 – continued from previous page

Feature name	Description	Expression
Distance to cluster center (r)	Distance of the customer to the center of the own cluster $z(n)$	$\frac{d_{nz(n)}}{\frac{1}{\lvert\mathcal{N}\rvert}\sum_{\hat{n}\in\mathcal{N}} d_{\hat{n}z(\hat{n})}}$
Cluster distance to depot	Distance of the cluster center $z(n)$ to the depot 0	$\frac{d_{0z(n)}}{\frac{1}{\lvert\mathcal{N}\rvert}\sum_{\hat{n}\in\mathcal{N}} d_{0z(\hat{n})}}$
Distance to closest other cluster centroid (r)	Distance of the customer to the center of the closest other cluster $\hat{z}(n)$	$\frac{d_{n\hat{z}(n)}}{\frac{1}{\lvert\mathcal{N}\rvert}\sum_{\hat{n}\in\mathcal{N}} d_{\hat{n}\hat{z}(\hat{n})}}$
Cluster area (r)	The area of the cluster customer n is assigned to, defined as the smallest possible square that fully encloses all customers in the cluster. Let A_n denote the area of customer n's cluster	$\frac{A_n}{\frac{1}{\lvert\mathcal{N}\rvert}\sum_{\hat{n}\in\mathcal{N}} A_{\hat{n}}}$
CVRP-specific features		
Vehicle capacity - demand ratio	The capacity of the vehicles divided by the total demand	$\frac{Q}{\sum_{\hat{n}\in\mathcal{N}} u_{\hat{n}}}$
Demand (r)	Demand volume of customer n	$\frac{u_n}{\sum_{\hat{n}\in\mathcal{N}} u_{\hat{n}}}$
Customer cluster demand	Demand volume of customer n divided by the average demand volume of all customers in the same cluster. The set $\mathcal{A}$ represents the customers in the own cluster	$\frac{u_n}{\frac{1}{\lvert\mathcal{A}\rvert}\sum_{\hat{n}\in\mathcal{A}} u_{\hat{n}}}$
Cluster demand share	Cumulative demand volume of customers in the same cluster divided by the cumulative demand volume of all customers	$\frac{\sum_{i\in\mathcal{A}} u_i}{\sum_{\hat{n}\in\mathcal{N}} u_{\hat{n}}}$

3.4.4 Scaling the output

The machine learning models provide numerical approximations of the Shapley value for each customer. However, the sum of these approximated Shapley values for all customers in a given instance may not exactly equal the total cost. To preserve the efficiency property (see Section 3.2), we scale the approximations for each instance in accordance with its total cost.

The Shapley value predictions generated by the machine learning models are denoted by $\hat{\phi}_n^{ML}$. To ensure that the total cost for each instance is fully allocated, the condition $\sum_{n\in\mathcal{N}(t)} \hat{\phi}_n = C(\mathcal{N}(t))\ \forall t \in \mathcal{T}$ must be satisfied. To achieve this, we scale the machine learning predictions in proportion to the total cost of each instance, yielding the final approximation according to Equation (3.5):

$$\hat{\phi}_n = \frac{\hat{\phi}_n^{ML}}{\sum\limits_{i\in\mathcal{N}(t)} \hat{\phi}_i^{ML}} \cdot C(\mathcal{N}(t)) \qquad \forall n \in \mathcal{N}(t),\, t \in \mathcal{T} \tag{3.5}$$

3.4.5 Generating fast but biased labels

Since the exact determination of Shapley values in this labeling process is only possible for relatively small instances and thus train data with exact Shapley values can only be generated for relatively small instances, we propose three methods for generating faster but biased labels.

The first option is to use approximated Shapley values as labels that are much faster to generate than exact Shapley values. We use Monte Carlo sampling, an unbiased estimator where the errors are identical and independently distributed (Shapley and Mann 1960). In Monte Carlo sampling, the Shapley value is not calculated as an average over all permutations (see Equation (3.2)) but only over a small proportion of all permutations. Mathematically, it is expressed in Equation (3.6):

$$\hat{\phi}_n^{MC} = \frac{1}{s} \sum_{p \in \mathcal{P}^s} C(p_{n-1} \cup \{n\}) - C(p_{n-1}) \tag{3.6}$$

$\mathcal{P}^s$ represents s random samples from the set $\mathcal{P}$ of all permutations of customers $\mathcal{N}$. The set of customers that appear in the order of permutation before customers n is represented by p_{n-1}. We use the same set of random permutations for each customer within the same instance. The great advantage of using this approximation method is that the bias is not subject to any systematic over- or underestimation of the Shapley value. Still, the approximation error through the bias corresponds to a distribution around the mean Value of 0. The larger s, the better the approximations. In the following, we refer to this modification as MLSVA-MC. The second option to speed up the label generation process is to evaluate the characteristic function not with an exact solver but with a heuristic that is very fast to evaluate and, therefore, speeds up the labeling process (MLSVA-H). This can also be done within the Monte Carlo sampling to have very fast labels generated by Monte Carlo sampling based on evaluating the routing cost of the subcoalitions through a heuristic (MLSVA-H&MC), which further speeds up the labeling process.

The idea behind this approach is that machine learning models, provided they are correctly calibrated and do not overfit the training data, do not learn the noise caused by the approximation of the labels as long as the deviation of the approximated Shapley values from the true Shapley values is not systematic but random.

3.5 Numerical study

First, we describe our numerical setup (Section 3.5.1). Then, we show the evaluation metrics (Section 3.5.2). After that, we detail the implementation of our machine learning models (Section 3.5.3), describe benchmark approaches (Section 3.5.4), and show our approximation results compared to these approaches for the TSP and CVRP (Section 3.5.5). In further experiments, we investigate how the approximation performance changes when training our models on biased labels or only on smaller instances (Section 3.5.6). Additionally, we report the computing times for generating labels and executing predictions (Section 3.5.7). Finally, we conduct further experiments on instances with different node distributions and on larger instances to further underscore the robustness of our approach (Section 3.5.8).

3.5.1 Numerical setup and instance generation

Since the exact determination of the Shapley values for large instances represents an extreme computational effort (see Figure 3.2), we measure the performance on new generated instances with up to 15 nodes, as in other works on Shapley value approximations (e.g., Levinger et al. 2021).

We create a total of 5000 new instances each for instance sizes from 7 to 15 nodes for the TSP and 7 to 13 nodes for the CVRP. For the CVRP, we assume a random vehicle capacity between

10 and 18 per instance and a demand volume per customer between 1 and 5 to include different demand structures in the instances. Each instance is generated by randomly setting X and Y coordinates in the interval 0 to 100. The depot is also set randomly in this interval. This allows us to achieve a broad mix of instances that does not follow a specific structure. However, if a special structure exists in the real-world environment, the training data should be generated to match this structure. Of these instances, we use 80% as training data and 20% as test data to evaluate and benchmark our machine learning framework against other methods. We use Gurobi 9.7 as the solver to compute the characteristic function by exactly solving the routing problems, limiting it to four threads. We implemented the TSP and the CVRP with classical formulations that are provided in Appendix C.1. Although advanced exact solution methods could potentially slightly reduce the time to calculate the exact Shapley values, even with state-of-the-art methods, exact Shapley values for larger instances remain computationally infeasible due to the exponential growth in subcoalitions. The data processing, the addressing of the solver, and the implementation of the machine learning models are conducted in Python 3.12. All experiments are carried out on an AMD Ryzen 9 5950X 16-Core Processor, 3.40 GHz with a 128 Gb RAM.

3.5.2 Evaluation metrics

We use the following evaluation metrics to evaluate our approximations for Shapley values.

- **MAPE:** The mean absolute percentage error from the true Shapley value, illustrated in Expression (3.7).

$$\text{MAPE} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \frac{|\hat{\phi}_n - \phi_n^{SV}|}{\phi_n^{SV}} \times 100 \tag{3.7}$$

- **MMAXPE:** The mean maximum absolute percentage error from the true Shapley value related to an instance, illustrated in Expression (3.8).

$$\text{MMAXPE} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \max_{n \in \mathcal{N}(t)} \frac{|\hat{\phi}_n - \phi_n^{SV}|}{\phi_n^{SV}} \times 100 \tag{3.8}$$

  where $\mathcal{T}$ is the set of routing problem instances and $\mathcal{N}(t)$ represents the customers in instance $t \in \mathcal{T}$. We introduce this unusual metric because it is particularly important for evaluating the worst deviation of approximations from the real Shapley value for each instance.

3.5.3 Implementation and hyperparameter tuning

We utilized various implementation environments and hyperparameter tuning methods to minimize the MAPE after scaling for our machine learning models, which are described individually below. Therefore, we conducted hyperparameter tuning separately for the TSP and CVRP. The resulting configurations with the hyperparameters are presented in Appendix C.2.

- **NN**: We implement the NN with TensorFlow Keras API. For hyperparameter tuning, we use the Bayesian optimization of the Keras-tuner. This probabilistic model-based approach constructs a posterior distribution to approximate and finds the optimal hyperparameters, balancing exploration and exploitation through an acquisition function (O'Malley et al. 2019). We use Adam Optimizer (Kinga et al. 2015) to train the weights.

- **XGB and RF**: We implement the RF algorithm with the Scikit-learn Python library, while XGB is implemented using the XGBoost Python library (Chen and Guestrin 2016). For both methods, we use Bayesian search for hyperparameter tuning.
- **PR, SVR and KNN**: We implement the algorithms with the Scikit-learn Python library (Pedregosa et al. 2011) and employ a grid search for hyperparameter tuning.

To standardize the features, we apply the StandardScaler from the Scikit-learn Python library, which transforms the data to have a mean (μ) of 0 and a standard deviation (σ) of 1.

3.5.4 Benchmark approaches

We use three proportional methods, along with the current state-of-the-art method SHAPO (Levinger et al. 2021), and Monte Carlo sampling, as described in Section 3.4.5, as benchmark approaches. SHAPO stands for SHapley Approximation based on a fixed Order and is an efficient method for approximating the Shapley value in our context. It simplifies computations by assuming a fixed servicing order for customers, estimating each customer's marginal contribution to the total cost, and allowing for polynomial-time computation. For a comprehensive treatment of the mathematical foundations and implementation of SHAPO, we refer the reader to Levinger et al. (2021). Note that SHAPO can only be used for the TSP as a benchmark, as it is not designed for the CVRP. To implement SHAPO, we reimplemented their algorithm according to the pseudo-code provided in their paper. For Monte Carlo sampling, we ensured fairness in the benchmark by using a high number of sampled permutations (150) to reduce variance and improve accuracy. As simple proportional methods, we use the depot distance and the reroute margin. Mathematically, these are expressed with Equations (3.9) and (3.10):

$$\hat{\phi}_n^{Depot} = C(\mathcal{N}) \cdot \frac{d_{0n}}{\sum\limits_{i \in \mathcal{N}} d_{0i}} \tag{3.9}$$

$$\hat{\phi}_n^{Reroute} = C(\mathcal{N}) \cdot \frac{C(\mathcal{N}) - C(\mathcal{N} \setminus \{n\})}{\sum\limits_{i \in \mathcal{N}} (C(\mathcal{N}) - C(\mathcal{N} \setminus \{i\}))} \tag{3.10}$$

A more complex proportional benchmark is the Alternative Cost Avoided Method (ACAM). This method, originally introduced by Tijs and Driessen (1986), first assigns each customer their separable cost, defined as their marginal contribution to the total cost. The remaining non-separable cost is then distributed proportionally based on the difference between each customer's standalone cost and their separable cost, reflecting the savings they achieve by joining the grand coalition. Mathematically, this is expressed with Equation (3.11):

$$\hat{\phi}_n^{ACAM} = m_n + \frac{(C(\{n\}) - m_n)}{\sum\limits_{i \in \mathcal{N}} (C(\{i\}) - m_i)} \cdot g(\mathcal{N}) \tag{3.11}$$

where $m_n = C(\mathcal{N}) - C(\mathcal{N} \setminus \{n\})$ is the marginal cost of customer n and $g(\mathcal{N}) = C(\mathcal{N}) - \sum_{n \in \mathcal{N}} m_n$ is the total non-separable cost.

3.5.5 Approximation performance

We present the approximation performance for the TSP (Section 3.5.5.1) and the CVRP (Section 3.5.5.2). Additionally, the 95th Percentile Absolute Percentage Error is reported in Appendix C.3, along with a detailed illustration of a 7-node TSP and CVRP instance for improved interpretability in Appendix C.4.

3.5.5.1 TSP results

The results of our benchmark are presented in Tables 3.2 and 3.3.

Table 3.2: MAPE [%] for MLSVA differentiated by ML Models and Benchmarks (TSP)

	MLSVA						Benchmarks				
No. of nodes	**NN**	**RF**	**XGB**	**PR**	**SVR**	**KNN**	**SHAPO**	**Depot**	**Reroute**	**ACAM**	**MCS**
7	1.61	5.87	3.60	2.94	4.00	22.64	**1.22**	26.40	60.09	10.38	6.35
8	**1.71**	6.05	3.65	2.75	3.04	20.01	1.79	27.65	62.29	12.33	6.83
9	**1.82**	6.60	3.73	2.83	3.42	19.44	2.30	28.58	64.72	14.37	7.26
10	**1.99**	6.59	3.89	3.03	3.41	17.77	2.84	28.60	66.30	15.46	7.63
11	**2.32**	6.72	4.21	3.31	4.02	20.02	3.42	29.21	67.88	16.94	7.78
12	**2.64**	6.84	4.34	3.85	4.42	18.02	4.00	29.30	68.59	17.98	8.02
13	**2.90**	7.09	4.56	4.50	4.72	19.04	4.46	29.34	69.69	18.67	8.29
14	**3.15**	7.38	4.97	4.51	4.80	17.80	5.02	30.10	68.51	19.12	8.38
15	**3.45**	7.65	5.07	4.77	5.03	17.58	5.56	29.98	69.44	19.83	8.70
avg.	**2.40**	6.75	4.22	3.61	4.10	19.15	3.40	28.80	66.39	16.12	7.69

Table 3.3: MMAXPE [%] for MLSVA differentiated by ML Models and Benchmarks (TSP)

	MLSVA						Benchmarks				
No. of nodes	**NN**	**RF**	**XGB**	**PR**	**SVR**	**KNN**	**SHAPO**	**Depot**	**Reroute**	**ACAM**	**MCS**
7	4.25	14.35	9.70	9.22	13.72	86.72	**3.08**	52.83	113.93	27.83	14.40
8	**4.79**	16.29	10.18	8.86	9.94	81.00	4.96	56.17	121.26	34.40	16.33
9	**5.13**	19.36	11.34	9.41	12.00	85.38	6.73	58.45	131.19	40.66	17.69
10	**5.87**	20.21	12.23	10.89	12.51	80.05	8.79	59.99	144.67	45.15	19.43
11	**7.14**	21.26	15.19	12.38	17.30	112.07	10.84	61.47	151.87	48.97	20.62
12	**8.51**	22.45	14.77	16.09	20.43	99.70	13.34	62.68	161.00	52.33	21.70
13	**9.77**	24.76	17.19	21.25	22.90	117.46	15.51	64.01	171.32	53.86	22.69
14	**10.21**	26.97	19.74	18.96	21.35	111.81	17.75	65.99	168.88	55.29	23.23
15	**11.93**	28.96	19.42	20.26	23.29	111.95	20.12	66.17	177.30	57.50	24.75
avg.	**7.51**	21.62	14.42	14.15	17.05	98.46	11.24	60.86	149.05	46.22	20.09

We find that for the TSP, the NN delivers by far the best performance among the machine learning models. With an average MAPE of 2.40%, it also clearly outperforms the current state-of-the-art approximation approach SHAPO, especially for larger instances. For the MMAXPE, the NN outperforms SHAPO even more significantly, which makes it a particularly suitable approach when cost allocations are desired where the worst approximation is not far away from the real Shapley value.

We further observe that the advanced machine learning models (all except KNN) significantly outperform the simple proportional proxies, although they do not reach the approximation quality achieved by the NN. Among the proportional methods, ACAM clearly delivers the best performance, yet it remains far behind the advanced machine learning models. Additionally, all advanced machine learning models outperform the MCS benchmark in MAPE. We also observe that as the number of customers in an instance increases, the approximation error tends to rise. However, this effect is even more pronounced for SHAPO and MCS, emphasizing the inherent difficulty of accurately approximating Shapley values. While a clear trend is visible, it remains uncertain whether this pattern would persist with larger instances or eventually stabilize.

3.5.5.2 CVRP results

Our benchmark results are presented in Tables 3.4 and 3.5 for the respective evaluation metrics.

Table 3.4: MAPE [%] for MLSVA differentiated by ML Models and Benchmarks (CVRP)

	MLSVA						Benchmarks			
No. of nodes	NN	RF	XGB	PR	SVR	KNN	Depot	Reroute	ACAM	MCS
7	**2.93**	6.43	4.49	4.61	5.31	20.71	21.70	45.54	22.72	4.98
8	**3.21**	6.51	4.47	4.17	4.64	19.76	21.78	46.89	24.26	5.09
9	**3.42**	6.67	4.90	4.30	4.57	18.44	22.53	45.41	22.82	5.19
10	**3.47**	6.62	4.88	4.06	4.41	16.87	23.00	45.08	22.81	5.16
11	**3.69**	6.99	5.08	4.29	4.62	16.53	23.66	45.00	23.91	5.30
12	**3.86**	6.90	5.09	4.40	4.77	17.53	24.04	43.78	24.18	5.34
13	**3.98**	6.97	5.21	4.49	5.24	17.38	24.32	43.52	25.14	5.42
avg.	**3.51**	6.73	4.87	4.33	4.79	18.17	23.00	45.03	23.69	5.21

Table 3.5: MMAXPE [%] for MLSVA differentiated by ML Models and Benchmarks (CVRP)

	MLSVA						Benchmarks			
No. of nodes	NN	RF	XGB	PR	SVR	KNN	Depot	Reroute	ACAM	MCS
7	**7.70**	15.63	11.59	13.48	16.08	76.41	47.97	98.22	56.92	11.25
8	**8.82**	16.19	12.80	12.32	14.36	81.41	49.65	102.16	58.70	12.20
9	**9.57**	17.19	13.70	13.35	14.15	83.73	55.79	103.91	58.10	12.84
10	**9.90**	17.93	13.93	12.34	13.60	78.68	58.28	103.88	59.10	13.23
11	**10.82**	20.54	15.59	14.49	16.10	89.76	62.67	108.50	63.60	14.45
12	**11.76**	21.23	16.20	15.61	17.78	95.43	67.02	109.74	66.80	14.74
13	**12.59**	21.67	16.63	15.64	20.93	94.60	70.77	112.23	71.10	14.82
avg.	**10.17**	18.63	14.35	13.89	16.14	85.72	58.88	105.52	62.05	13.36

Consistent with the TSP results, the NN achieves the best approximations in both MAPE and MMAXPE, although the overall approximation quality is slightly lower compared to the TSP. All advanced machine learning models again clearly outperform the proportional methods in both metrics. The MCS benchmark achieves comparable results to some advanced machine learning models in terms of MAPE and is only surpassed by the NN in MMAXPE; however, it incurs a significantly higher computational cost (see Section 3.5.7.2). Additionally, we observe the same trade-off as in the TSP: the approximation error increases with instance size, though this effect is less pronounced compared to the TSP.

All reported values represent scaled predictions. Detailed analysis revealed that scaling the approximations to match the total routing cost consistently improves performance across all machine learning models. However, this improvement remains small (e.g., 0.16 percentage points for the NN in the TSP), except for KNN, as the cumulative predictions of the machine learning models already closely match the actual total instance costs. Similarly, improvements in the MMAXPE metric resulting from scaling are also minor (e.g., 0.31 percentage points for the NN in the TSP).

While our MLSVA approach does not explicitly enforce the axiomatic properties of the Shapley value, the resulting approximations closely align with the true values. Consequently, the fairness properties of the Shapley value are nearly preserved, providing a more game-theoretically justified cost allocation compared to simple proportional methods and other practical cost allocation methods.

3.5.6 Training with approximated labels

This section shows the effects of fast-to-generate but biased labels (see Section 3.4.5) in the training process. We show this for Monte Carlo sampled Shapley values (MLSVA-MC) (Section 3.5.6.1), for the labels generated by evaluating the routing problems with a very fast but simple heuristic (MLSVA-H), as well as for the combination of both biases (MLSVA-H&MC) (Section 3.5.6.2). We further show the effects of only using smaller instances as training data (Section 3.5.6.3).

If these experiments demonstrate that the approximation performance decreases only minimally, this shows that we can also apply our methodology to larger instances with multiple customers, as we can generate the necessary biased labels very quickly and still achieve good approximation results. In the following, we conduct all our experiments using only the best-performing model, the NN. Note that we have individually tuned the number of epochs in the training process of the NN for these modifications, as we have found that for biased labels, a lower number of epochs has a better effect on the performance of our approach.

3.5.6.1 Monte Carlo approximated labels

In this section, we evaluate the effect of labels generated by Monte Carlo sampling. To avoid memory errors, we generate the random permutations using the Fisher-Yates shuffle algorithm, which runs in complexity of $O(n)$. Further, we take advantage of memoization (caching the value of already evaluated subcoalitions) as this significantly reduces computing times for generating the labels. We use a small number of $s = 50$ random samples of permutations for our experiments. We deliberately use this small number to investigate the effect of a strong bias in the training data on the performance of the NN. The approximation quality of the labels (MC performance), as well as the approximation quality of the NN trained with these labels (MLSVA-MC), is shown in Table 3.6 for both the TSP and the CVRP. The performance of Monte Carlo sampling is reported in terms of MAPE and MMAXPE, while the performance of the MLSVA-MC is evaluated based on the increase in MAPE and MMAXPE compared to the results in Sections 3.5.5.1 and 3.5.5.2 (expressed in percentage points as Δ MAPE / Δ MMAXPE). For instance, a value of 1.5% indicates that MAPE increases from, e.g., 3.0% to 4.5%.

We observe that the labels generated through Monte Carlo sampling exhibit a strong bias, with MAPE values of 13.31% for the TSP and 9.13% for the CVRP, and MMAXPE values reaching 34.74% and 23.27%, respectively. However, our MLSVA-MC achieves excellent results, performing only slightly worse than MLSVA trained on exact Shapley values. For instance, in the CVRP, the

Table 3.6: Performance of MC-Sampling and MLSVA-MC in %

	TSP				CVRP			
	MC		MLSVA-MC		MC		MLSVA-MC	
Nodes	**MAPE**	**MMAXPE**	**Δ MAPE**	**Δ MMAXPE**	**MAPE**	**MMAXPE**	**Δ MAPE**	**Δ MMAXPE**
7	10.91	24.71	1.84	4.47	8.80	19.91	1.58	3.33
8	11.87	28.17	1.59	4.04	8.84	21.11	1.35	2.95
9	12.45	30.71	1.74	4.86	9.07	22.39	1.11	2.62
10	13.01	33.03	1.73	4.73	9.24	23.72	0.97	2.16
11	13.45	35.04	1.75	4.93	9.27	24.54	0.94	2.74
12	13.97	37.49	1.75	5.24	9.37	25.28	0.85	2.54
13	14.34	39.23	1.82	5.33	9.34	25.97	0.83	2.78
14	14.66	41.14	1.74	4.98	-	-	-	-
15	15.11	43.18	1.75	4.90	-	-	-	-
avg.	13.31	34.74	1.75	4.83	9.13	23.27	1.09	2.73

average MAPE increases only marginally from 3.51% to 4.60%. Additionally, while MC produces high MMAXPE values, the impact on MMAXPE when using these labels remains relatively minor.

We could thus show that it is possible to approximate Shapley values with high precision, although using approximated Shapley values with very strong biases as labels. This shows that the performance of our approach only slightly declines when the machine learning models are not trained with exact labels but with biased labels that are much faster to generate. This insight is particularly important for larger instance sizes, for which exact Shapley values can no longer be generated.

3.5.6.2 Trained with heuristically solved routing problems

In this section, we analyze the impact on performance if the routing problems required for labeling are not solved exactly but instead with a very fast but straightforward heuristic. We deliberately use a heuristic that can be solved quickly but no longer finds the optimal solution, even for the smallest instances, in order to check the effects of routing problems that have not been solved optimally. Therefore, we use a simple *nearest neighbor heuristic* as a starting solution and then improve this solution with a simple *remove and best insert heuristic.* The pseudo-code is shown in Algorithm 5.

Algorithm 5: Remove and best insert heuristic

$\mathcal{S} \leftarrow$ **nearestNeighborHeuristic()**; // Generate initial solution $\mathcal{S}$
$it \leftarrow 0$;
while $it < \tau$ **do**
  $\hat{\mathcal{N}} \leftarrow$ **randomSelect**$(\mathcal{N}, 1, \lambda)$; // Select between 1 and λ customers
  $\mathcal{S} \leftarrow$ **remove**$(\mathcal{S}, \hat{\mathcal{N}})$; // Remove customers from solution
  for $n \in \hat{\mathcal{N}}$ **do**
    $\mathcal{S} \leftarrow$ **bestInsertion**$(\mathcal{S}, n)$; // Insert customer in solution
  $it \leftarrow it + 1$;

We provide two different configurations of the heuristic referred to as H1 and H2 to evaluate the effect of two different performing heuristics. We set the parameters as follows: H1: $\lambda = 2$ and $\tau = 2|\mathcal{N}|$, H2: $\lambda = 4$ and $\tau = 4|\mathcal{N}|$. The parameter λ defines the maximum number of customers that can be randomly selected and removed from the solution in each iteration. Table 3.7 shows the TSP and CVRP performance of the heuristic compared to the optimal routing costs.

Table 3.7: Mean GAP between the optimal solution and heuristic solutions for different instance sizes averaged over 100 instances

	H1		**H2**	
No. of nodes	**GAP TSP [%]**	**GAP CVRP [%]**	**GAP TSP [%]**	**GAP CVRP [%]**
7	0.13	2.94	0.00	0.03
8	0.56	3.89	0.00	0.02
9	1.20	6.67	0.00	1.03
10	1.62	5.98	0.17	1.78
11	2.22	6.88	0.22	3.27
12	2.78	8.66	0.40	3.86
13	3.43	8.58	0.64	3.47
14	4.20	-	1.07	-
15	4.42	-	1.63	-
avg.	2.28	6.23	0.46	1.92

As can be seen, the heuristic leads to deviations from the optimal solution even with our small instance sizes. This effect is intentional, as it allows us to evaluate the influence of this source of bias on the approximation performance of our NN. Notably, the heuristic performs slightly worse for the CVRP than for the TSP, and H1 is significantly less effective than H2.

In order to evaluate the effects on the approximation performance, we distinguish between two cases. In the first case (MLSVA-H), we use the heuristic to determine the characteristic function used for the Shapley values. In the second case (MLSVA-H&MC), we generate the labels by Monte Carlo sampling with $s = 250$ samples and use the heuristic to evaluate the routing costs of the subcoalitions. As a result, we have two sources of bias in the labels. The first one is by sampling permutations using the Monte Carlo sampling method. The second one is by the heuristic evaluation of the subcoalitions cost. The approximation performance of the NN for these two cases is presented in Table 3.8 (Δ MAPE [%]) and Table 3.9 (Δ MMAXPE [%]) as an increase in percentage points.

Table 3.8: Performance of MLSVA-H and MLSVA-H&MC in Δ MAPE [%]

	TSP				**CVRP**			
	MLSVA-H		**MLSVA-H&MC**		**MLSVA-H**		**MLSVA-H&MC**	
No. of nodes	**H1**	**H2**	**H1**	**H2**	**H1**	**H2**	**H1**	**H2**
7	1.13	0.50	1.65	0.84	2.32	0.58	2.73	0.96
8	0.95	0.46	1.47	0.63	2.49	0.46	2.63	0.76
9	0.93	0.52	1.59	0.75	2.56	0.54	3.18	0.90
10	1.06	0.53	1.61	0.71	2.74	0.60	3.20	0.93
11	1.38	0.48	1.83	0.72	2.85	0.86	3.61	1.07
12	1.51	0.56	2.00	0.82	3.23	0.95	4.01	1.20
13	1.69	0.58	2.16	0.89	3.33	1.31	3.87	1.53
14	2.11	0.77	2.31	1.02	-	-	-	-
15	2.14	0.81	2.25	1.03	-	-	-	-
avg.	1.43	0.58	1.87	0.82	2.79	0.76	3.32	1.05

As demonstrated, even with biased labels, our approach maintains strong approximation performance. Notably, even when both biases are present (MLSVA-H&MC), the decline in MAPE remains minimal compared to the original MLSVA without modifications. Moreover, the results clearly indicate that the stronger the heuristic, the smaller the performance loss. In contrast to MAPE, the impact on MMAXPE is significantly larger, particularly when using the weaker heuristic H1. A detailed analysis reveals that this is primarily driven by customers with very small

Table 3.9: Performance of MLSVA-H and MLSVA-H&MC in Δ MMAXPE [%]

	TSP				CVRP			
	MLSVA-H		MLSVA-H&MC		MLSVA-H		MLSVA-H&MC	
No. of nodes	**H1**	**H2**	**H1**	**H2**	**H1**	**H2**	**H1**	**H2**
7	3.59	1.20	4.63	2.19	7.81	2.07	8.65	3.31
8	3.48	1.24	4.41	1.50	9.96	1.92	10.27	2.94
9	3.88	1.82	5.92	2.25	11.58	2.37	14.38	3.70
10	4.41	2.02	6.20	2.28	13.84	2.99	15.69	4.33
11	6.90	2.13	8.51	2.84	17.01	5.13	19.66	5.60
12	7.65	2.90	9.42	3.54	21.83	6.44	21.27	6.53
13	8.53	3.05	10.23	4.24	23.08	8.18	22.44	8.38
14	11.57	4.79	12.91	5.27	-	-	-	-
15	11.05	4.77	11.53	4.99	-	-	-	-
avg.	6.78	2.66	8.20	3.23	15.02	4.16	16.05	4.97

Shapley values, as the heuristic evaluation of their marginal costs tends to overestimate their contribution to subcoalitions. The comparison between H1 and H2 highlights substantial differences in MMAXPE, reinforcing that a more accurate heuristic reduces this bias effectively.

Nevertheless, we were able to show that our approach achieves good approximation results even with large biases in the labels. This finding is important because it enables us to quickly generate labels for larger instances, allowing our machine learning-based approach to train efficiently without a substantial loss in performance. Overall, Sections 3.5.6.1 and 3.5.6.2 demonstrate that our method can effectively learn well-approximated Shapley values from poorly approximated ones, which highlights the applicability of our approach for larger instances. Furthermore, this is an important methodological insight, as it opens the way to improve further approximation algorithms in other contexts through machine learning algorithms.

3.5.6.3 Trained on smaller instances

In this experiment, we examine the effects of training the NN on smaller instances and then testing it on larger instances. To do this, we use the same training data as before, with the difference that all instances with more than 12 nodes for the TSP and more than 10 nodes for the CVRP are excluded. We then let our NN approximate Shapley values for instances with 13-15 nodes for the TSP and 11-13 nodes for the CVRP. The performance is shown in Table 3.10.

Table 3.10: Performance Improvement in Δ MAPE [%] and Δ MMAXPE [%]

TSP			CVRP		
No. of nodes	**Δ MAPE [%]**	**Δ MMAXPE [%]**	**No. of nodes**	**Δ MAPE [%]**	**Δ MMAXPE [%]**
13	0.37	1.80	11	0.67	2.40
14	0.66	2.61	12	0.99	3.69
15	1.02	3.15	13	1.40	4.84

As can be seen, the approximation performance only increases marginally if the NN is trained exclusively on smaller instances. For example, in the case of TSP instances with 15 nodes, the MAPE increases only slightly from 3.45% to 4.47%. Consequently, this experiment shows that our methodology also leads to good approximation results when we train on smaller instances than we test. Overall, the experiments in Section 3.5.6 show the scalability of our methodology, as it creates a framework that can be used to train machine learning models efficiently for larger instances.

3.5.7 Runtime analysis

In this section, we analyze the computational burden associated with different stages of our approach. First, in Section 3.5.7.1, we examine the time required to generate labeled training data, comparing the computational effort of our proposed biased labels to the exact computation of Shapley values. However, it is essential to emphasize that these labeling runtimes do not reflect the actual execution time of our MLSVA approach. Once the machine learning models are trained, only the computational burden for feature generation and the machine learning prediction time is relevant. These execution times are presented in Section 3.5.7.2.

3.5.7.1 Label generation

Figure 3.3 presents the runtimes required to generate labels for a single instance of different problem sizes. As already outlined, computing exact Shapley values for labeling (MLSVA) becomes infeasible for larger instances due to the factorial growth in complexity. In contrast, our proposed biased labels—particularly those incorporating heuristics—enable significantly faster labeling. For example, the MLSVA-H2&MC label requires only 4.45 seconds for a TSP instance with 15 nodes, demonstrating that our method can efficiently generate training data for much larger instances.

These results highlight the practicality of our approach in enabling scalable training datasets for machine learning models.

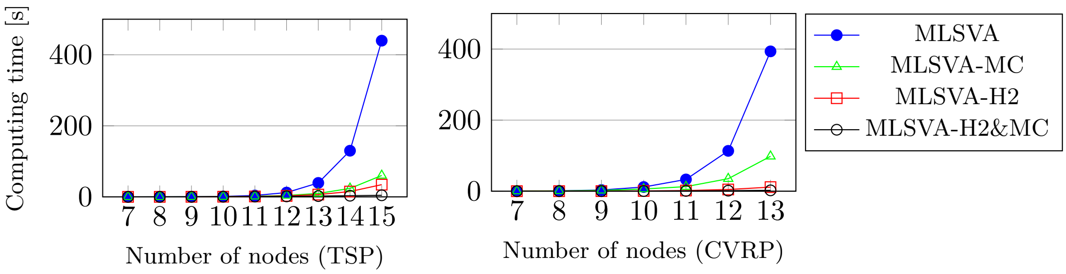

Figure 3.3: Computing time for labeling a single instance averaged over 100 instances

3.5.7.2 Execution times

Once the machine learning model is trained, execution time consists of two components: feature preparation and prediction. The latter is negligible, with the NN requiring only 0.000028 seconds per observation in the TSP, and similar times across other models and problems. The primary computational effort arises from feature generation, particularly the evaluation of marginal costs, which requires solving the instance while omitting each customer individually.

To address this, we report runtimes for two configurations: one with exact marginal cost calculations (Exact MC) and one where marginal costs are evaluated using the heuristic from Section 3.5.6.2 (Heuristic MC). Table 3.11 presents execution times for instances with 11–15 nodes.

Table 3.11: Runtime in seconds

	TSP		CVRP	
No. of nodes	Exact MC	Heuristic MC	Exact MC	Heuristic MC
< 11	0.18	0.07	0.6	0.06
12	1.05	0.09	2.50	0.07
13	2.63	0.09	4.16	0.08
14	6.57	0.10	–	–
15	16.49	0.11	–	–

While exact marginal cost evaluation remains feasible for the considered instance sizes, it becomes computationally expensive for larger instances. In contrast, the heuristic-based approach ensures consistently low runtimes, making our method scalable and well-suited for generating fast approximations even in large-scale applications. Notably, the heuristic evaluation of marginal costs incurs only a minor performance loss of 0.37% on average for TSP instances and 0.69% for CVRP instances in terms of Δ MAPE —despite relying on a relatively weak heuristic.

For comparison, the MC benchmark in Sections 3.5.5.1 and 3.5.5.2 already requires 117.54 seconds for 15-node TSP instances and 295.33 seconds for 13-node CVRP instances, whereas SHAPO, being a polynomial-time algorithm, runs almost instantly.

3.5.8 Further experiments

In this section, we conduct two experiments that further demonstrate the robustness of our approach. First, we examine the effects of testing our approach on instances with different characteristics from those it was trained on (Section 3.5.8.1). Next, we conduct experiments on significantly larger instances (Section 3.5.8.2).

3.5.8.1 Testing on different instances

We examine the effects of training the NN on the previously introduced training data and then applying it to newly generated instances with a different node location distribution. Specifically, we created 100 new instances for each instance size ranging from 7 to 15 nodes for the TSP, and from 7 to 13 nodes for the CVRP. These new instances differ from the original ones in that the customer coordinates follow a distinct spatial distribution. In this case, customer locations are drawn from a bimodal triangular distribution: with 50% probability, nodes are sampled from a triangular distribution over the left half of the coordinate space $[0, 50] \times [0, 100]$ with its peak at $(25, 50)$; the other 50% are drawn from the right half $[50, 100] \times [0, 100]$, peaking at $(75, 50)$. This results in two spatially distinct cluster centers. For a visual representation of the distribution, please see the probability heatmap in Figure C.3 in Appendix C.5. The depot node is still set randomly. Table 3.12 presents the increase in MAPE and MMAXPE for these newly generated instances.

As shown, the performance of MLSVA deteriorates only minimally when tested on instances that differ structurally from the training data. This demonstrates the robustness of our approach in handling different spatial distributions. Nevertheless, we recommend that for optimal performance, the training instances should be structurally similar to the instances on which the MLSVA is

Table 3.12: Performance on different characterized instances in Δ MAPE [%] and Δ MMAXPE [%]

	TSP		CVRP	
No. of nodes	**Δ MAPE [%]**	**Δ MMAXPE [%]**	**Δ MAPE [%]**	**Δ MMAXPE [%]**
7	0.81	1.53	0.78	2.08
8	0.86	1.54	1.30	2.80
9	0.96	1.86	1.16	2.47
10	0.76	1.13	1.31	2.71
11	0.65	0.83	1.58	3.93
12	0.39	0.83	1.18	2.67
13	0.25	0.58	1.60	4.41
14	0.51	1.06	-	-
15	0.38	0.39	-	-
avg.	0.62	1.08	1.27	3.01

subsequently applied.

3.5.8.2 Stability on larger instances

Validating our method on larger instances presents a challenge, as generating true labels becomes infeasible. To demonstrate the stability of our approach, we generated 1500 instances for each individual instance size ranging from 31 to 35 nodes for both the TSP and the CVRP, following the same settings as described in Section 3.5.1, and using an 80/20 train-test split. The observations were labeled using the biased labels of MLSVA-H2&MC from Section 3.5.6.2 and the marginal cost were heuristically evaluated.

Since both the biased labels and our approximations are estimates of the Shapley value, traditional evaluation metrics like MAPE and MMAXPE are not directly applicable. Instead, we focus on the relationship between the biased labels and our Shapley value approximations. Figure 3.4 illustrates this relationship by comparing our approximations with the biased labels for both the TSP and CVRP. The scatter plots use transparency to better visualize the density of data points.

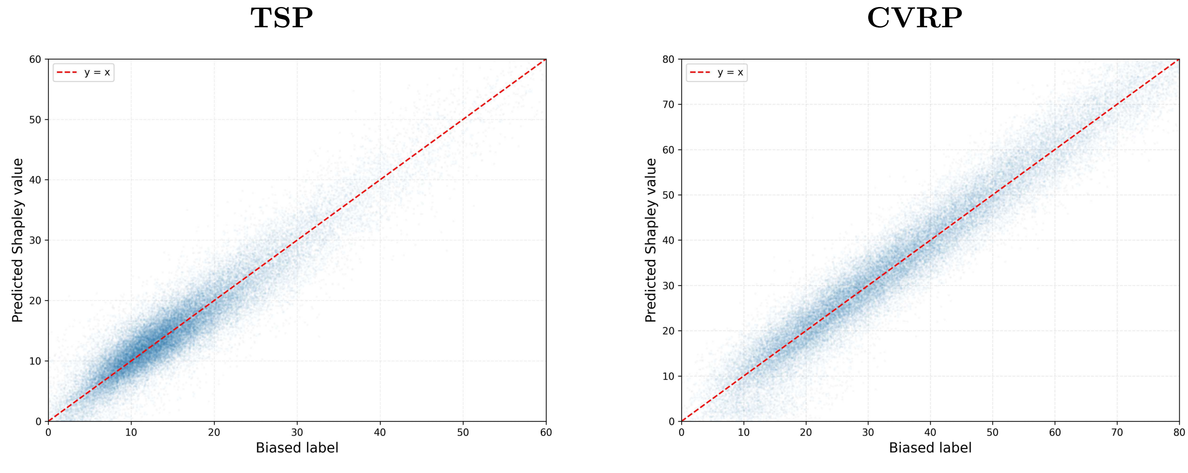

Figure 3.4: Comparison of Shapley approximations and biased labels

As shown in the figure, there is a clear correlation between the biased labels and the Shapley value approximations, as evidenced by a Pearson correlation coefficient of 0.94 for the TSP and 0.97 for the CVRP. The stronger performance for the CVRP can be attributed to the fact that the Monte-Carlo generated labels are closer to the true Shapley values for the CVRP than for the TSP (see Table 3.6 in Section 3.5.6.1). This suggests that our approximation approach remains stable even on larger instances. Moreover, as indicated by the experiments in Section 3.5.6, which show that we can generate well-approximated Shapley values even with highly biased labels, the deviation from the true Shapley value is expected to be much smaller than the deviation from the biased labels. Further, regarding runtime considerations, we observe that labeling a 35-node instance takes an average of 85.42 seconds, which makes it feasible to generate large labeled datasets for training the machine learning models. The execution time remains efficient, with an average feature generation time of 0.76 seconds per instance for the 35-node cases, enabling fast approximations even for larger instances.

3.6 Analysis of feature importance

In this section, we evaluate the importance of the features and thereby provide managerial insights about the main cost drivers in routing problems. We do this by using the Python package SHAP (SHapley Additive exPlanations) (Lundberg and Lee 2017). This package interprets the output of machine learning models by computing the contribution of each feature to the prediction inspired by Shapley values. Figure 3.5 depicts the SHAP Values of the ten features with the highest value for the TSP and CVRP.

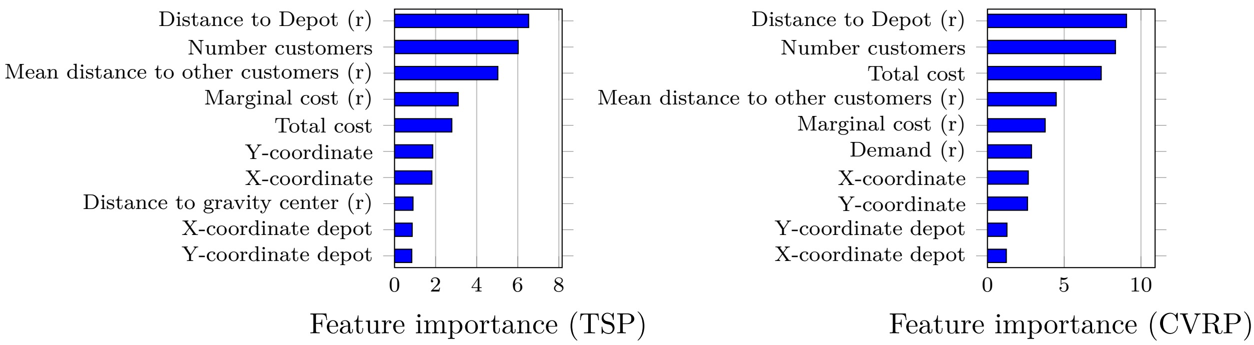

Figure 3.5: SHAP values of the ten most important features

We observe that the ten most important features are nearly identical for both the TSP and CVRP, with the only difference being the addition of the demand ratio in the CVRP. Interestingly, features related to the clustering of individual customers do not rank among the top ten in either case but still contribute to the approximation performance.

Table 3.13 shows the increase in MAPE of our MLSVA when trained with only these top ten features. For the TSP, the performance remains nearly comparable to the fully trained NN. For the CVRP, there is a slightly more noticeable decline in performance; however, the Shapley values are still approximated with an solid average MAPE. Detailed further analysis revealed the following effects:

- **Correlation among features:** While there is only minor correlation among the top ten features, some features exhibit high pairwise correlation, such as *Marginal Cost (r)* and *Shortcut cost (r)*. Despite this, their joint inclusion still improves the model's performance. Similarly, while distance-based features show some correlation, including all of them leads to the best overall performance.

- **Direction of influence:** For some features—such as *Mean Distance to Other Customers (r)* or *Marginal Cost (r)*—a consistent positive correlation with the model output is observed. In contrast, features like the coordinates exhibit less clear directional effects, as their influence is often shaped by interactions with other spatial attributes, such as the depot location or overall instance-based features.

- **Other machine learning models:** While the results presented here are for the NN, we also computed the SHAP values for RF, XGB, and PR. The obtained SHAP values across the different models are highly correlated, with the lowest Pearson correlation coefficient between any pair of models being 0.86. Note that for these models, a subset of the data was used for SHAP value computation.

Table 3.13: Performance of the MLSVA trained with the ten most important features in Δ MAPE [%] and Δ MMAXPE [%]

	TSP		CVRP	
No. of Nodes	**Δ MAPE [%]**	**Δ MMAXPE [%]**	**Δ MAPE [%]**	**Δ MMAXPE [%]**
7	0.71	2.36	1.49	3.88
8	0.89	2.73	1.57	3.61
9	1.12	3.63	1.87	4.53
10	1.11	3.21	1.78	3.95
11	1.18	3.75	1.88	4.12
12	1.18	2.86	1.79	4.02
13	1.27	3.53	1.88	4.04
14	1.34	3.69	-	-
15	1.28	2.96	-	-
avg.	1.12	3.19	1.75	4.02

3.7 Generalizability of the methodology

In this section, we show the generalizability of our approach by applying it to the approximation of Shapley values for items in a variant of the bin packing problem with size and weight constraints.

Let $\mathcal{I}$ be the set of items. Each item $i \in \mathcal{I}$ has a specific weight w_i and a specific size s_i. These items must be packed into bins, whereby the weight capacity W and the size capacity S must not be exceeded for each bin. The objective is to minimize the number of bins used. We assume a uniform cost of one per bin, allowing the number of bins to be interpreted as total cost. The mathematical model formulation and notation is available in Appendix C.1.3.

The specific problem characteristic of bin packing problems is that the marginal costs of the items have a binary structure where they are either 0 (no new bin must be opened) or 1 (exactly one new bin must be opened). We apply our MLSVA as described in Section 3.4 with exact Shapley values as labels. The only difference is the use of bin packing-specific features that are described in Table 3.14. These features capture the size and weight characteristics of items, bin utilization, and distribution-based statistics, ensuring the model accounts for key structural properties of bin packing instances.

Table 3.14: Features (bin packing)

Feature name	**Description**	**Expression**
Weight (r)	Weight of item i	$\frac{w_i}{\frac{1}{\lvert\mathcal{I}\rvert}\sum\limits_{j\in\mathcal{I}} w_j}$
Size (r)	Size of item i	$\frac{s_i}{\frac{1}{\lvert\mathcal{I}\rvert}\sum\limits_{j\in\mathcal{I}} s_j}$
Weight capacity ratio	Weight capacity of bins divided by total weight of all items	$\frac{W}{\sum\limits_{i\in\mathcal{I}} w_i}$
Size capacity ratio	Size capacity of bins divided by total size of all items	$\frac{S}{\sum\limits_{i\in\mathcal{I}} s_i}$
No. of bins	Number of bins in (optimal) solution	$\sum\limits_{b\in\mathcal{B}} y_b$
No. of items	Number of items in the instance	$\lvert\mathcal{I}\rvert$
$\sigma-$weight	Standard deviation of item weights	$\sigma(\{w_i\}_{i\in\mathcal{I}})$
$\sigma-$size	Standard deviation of item sizes	$\sigma(\{s_i\}_{i\in\mathcal{I}})$
Bin utilization size	Utilization of the bins by size	$\frac{\sum\limits_{i\in\mathcal{I}} s_i}{\sum\limits_{b\in\mathcal{B}} y_b \cdot S}$
Bin utilization weight	Utilization of the bins by weight	$\frac{\sum\limits_{i\in\mathcal{I}} w_i}{\sum\limits_{b\in\mathcal{B}} y_b \cdot W}$
Item-Bin size ratio	Items size divided by bins size capacity	$\frac{s_i}{S}$
Item-Bin weight ratio	Items weight divided by bins weight capacity	$\frac{w_i}{W}$
Weight x-percentile ratio	Weight of item i divided by the x-percentile of weight distribution. We include this feature for $x \in \{0.0, 0.25, 0.5, 0.75, 1\}$	$\frac{w_i}{P_x(w)}$
Size x-percentile ratio	Size of item i divided by the x-percentile of size distribution. We include this feature for $x \in \{0.0, 0.25, 0.5, 0.75, 1\}$	$\frac{s_i}{P_x(s)}$
Weight percentile	Percentile rank of item i's weight	$\text{Percentile}(w_i, w_j : j \in \mathcal{I})$
Size percentile	Percentile rank of item i's size	$\text{Percentile}(s_i, s_j : j \in \mathcal{I})$
Weight combinations	Combinations of items, including this one fitting in a bin by weight	-
Size combinations	Combinations of items, including this one fitting in a bin by size	-
Total combinations	Combinations of items, including this one fitting in a bin by weight and size	-
Marginal cost (r)	Marginal cost of item i	$\frac{C(\mathcal{I})-C(\mathcal{I}\setminus\{i\})}{\frac{1}{\lvert\mathcal{I}\rvert}\sum\limits_{i\in\mathcal{I}}(C(\mathcal{I})-C(\mathcal{I}\setminus\{\hat{i}\}))}$

We generated 3000 instances for each number of items, ranging from 7 to 15. The weights and sizes of the items are separately randomly set between 1 and 10. The bin size and weight are separately randomly set between 17 and 25. Once again, we used 80% of the instances for training and 20% for testing. In addition to the performance of the machine learning models, we also show the performance of the most plausible proportional approximation method by weight and size of the items. Formally, WS represents the approximation based on the equally weighted shares of weight and size and is expressed with Equation (3.12),

$$\hat{\phi}_i^{ws} = C(\mathcal{I}) \cdot (0.5 \cdot \frac{w_i}{\sum_{j \in \mathcal{I}} w_j} + 0.5 \cdot \frac{s_i}{\sum_{j \in \mathcal{I}} s_j}) \tag{3.12}$$

Table 3.15 displays the performance of the MLSVA applied to this bin packing problem. We observe that similar to the routing problems, the NN delivers the best approximation results. In contrast, the performance does not decrease with an increasing number of items in the instance. The results demonstrate that our methodology also leads to excellent approximations for Shapley values in an entirely different problem setting, like bin packing problems. All machine learning models deliver by far a better performance compared to the proportional benchmark.

Table 3.15: MAPE and MMAXPE [%] for MLSVA differentiated by ML Models (bin packing)

	MAPE							**MMAXPE**						
$\lvert\mathcal{I}\rvert$	**NN**	**RF**	**XGB**	**PR**	**SVR**	**KNN**	**WS**	**NN**	**RF**	**XGB**	**PR**	**SVR**	**KNN**	**WS**
7	**2.81**	3.26	3.27	4.03	3.73	6.88	23.49	**6.85**	8.36	8.38	9.99	9.25	16.97	55.70
8	**3.37**	4.00	3.69	4.16	3.88	6.16	22.62	**8.96**	10.71	9.77	10.81	10.23	15.93	57.94
9	**3.59**	4.36	3.90	4.32	4.04	6.06	22.39	**9.45**	11.65	10.40	11.15	10.67	15.86	57.91
10	**3.48**	4.24	3.67	4.12	3.86	5.39	21.17	**9.62**	11.93	10.26	11.37	10.59	14.84	56.78
11	3.50	4.03	**3.49**	3.86	3.62	5.15	21.68	**10.08**	11.61	10.10	11.19	10.53	14.53	61.29
12	**3.15**	3.90	3.29	3.69	3.43	4.79	22.01	**9.48**	11.49	9.63	10.68	9.87	13.87	63.90
13	**3.42**	4.19	3.51	4.10	3.79	4.94	21.97	**10.11**	12.81	10.81	12.18	11.06	14.74	66.43
14	**3.05**	3.78	3.13	3.70	3.32	4.37	21.83	9.84	11.48	**9.64**	11.29	9.82	12.96	65.84
15	**3.15**	3.90	3.19	3.72	3.33	4.41	21.72	**9.73**	12.21	9.89	11.63	10.02	13.46	68.54
avg.	**3.28**	3.96	3.46	3.97	3.67	5.35	22.10	**9.35**	11.36	9.88	11.14	10.23	14.80	61.59

3.8 Conclusion

Our study introduced a new machine learning-based approach for approximating Shapley values. Extensive numerical experiments showed that this approach leads to outstanding approximation results for Shapley values in the TSP and CVRP.

We demonstrated that the approximation through the exploitation of problem-specific knowledge leads to better approximations than simple proxies and current state-of-the-art approximation methods.

In addition, we showed through further experiments that this approach allows us to learn from fast but poorly generated labels and still achieve excellent approximation results. Furthermore, we could show the generalizability of our approximation approach by using it in the context of a bin packing problem to approximate Shapley values for the items. Overall, we present an approach that: (1) effectively utilizes the routing-specific problem structure, (2) enables real-time approximations, (3) can be trained on biased labels with only a moderate loss in performance, (4) identifies key factors influencing Shapley values, and (5) demonstrates versatility by being applicable to entirely different contexts, such as bin packing. Even though, questions about the scalability of Shapley value approximations in general remain an exciting field for future research as we still observed a trade of between instance size and approximation quality.

The results in the context of routing are particularly relevant for logistics service providers that need to allocate costs to individual customers. This approach enables them to generate cost allocations close to the real Shapley values immediately. Furthermore, if alternative game-theoretic cost allocation methods are preferred, which also require evaluating all possible subcoalitions, the same methodology can be applied, with the only distinction being the use of the selected cost allocation method as the label. Another promising direction would be to leverage machine learning techniques to directly predict cost allocations that satisfy specific game-theoretic properties, particularly in real-world applications where certain fairness criteria are essential or desired.

In general, we have demonstrated that machine learning models are an efficient tool for leveraging problem-specific knowledge in routing problems. We encourage further research in this direction by exploring similar approaches and assessing their applicability to other vehicle routing problem variants, such as those with time windows or additional constraints, as well as to other combinatorial optimization problems, such as bin packing problems with differently characterized bins (e.g., varying sizes or opening costs). Beyond Shapley value approximation, future research could also focus on broader applications of machine learning in routing, such as predicting feasibility in highly constrained routing problems, estimating marginal costs, forecasting partial routes, or even directly approximating total costs. We hope that our study serves as a foundation for further advancements in integrating machine learning with combinatorial optimization.

Acknowledgments

The paper was written as part of a project funded by the German Research Foundation. We would like to express our gratitude to the German Research Foundation (project number: ROCOCO FO1468/1-1) for the financial support of our research project. We would also like to thank Dennis Götz for his support with feature generation and implementation as part of his final thesis. In addition, we would like to thank the anonymous reviewers and the editor for their valuable recommendations, which have significantly improved our paper.

Contribution 4 - Approximating Shapley values with subcoalition Shapley values in routing problems

4 Approximating Shapley values with subcoalition Shapley values in routing problems

Johannes Gückel

Abstract The Shapley value is a widely recognized method for fair cost allocation in cooperative game theory. However, its computational intensity makes it challenging to apply in routing problems, particularly for allocating costs among customers. To address this issue, we propose a new approximation method called β Subcoalition Shapley Approximation, which is based on the mean weighted Shapley values of subcoalitions with exactly β customers. We test this approximation method for the traveling salesman problem as well as for the capacitated vehicle routing problem. Through an extensive numerical study, we demonstrate that our approximation provides excellent results, representing a good trade-off between approximation quality and computational intensity compared to other methods. Moreover, our approximation method is not limited to routing problems but can also be applied to other Shapley value applications, particularly suitable for NP-hard underlying problems where computation time increases dramatically with increasing instance size.

Accepted for publication in Transportation Research Procedia

4.1 Introduction

The cost allocation according to the Shapley value is highly recognized for its fairness in distributing costs within a coalition to individual participants. In routing problems, it can be utilized to allocate the total costs to individual customers (Engevall et al. 2004). However, the exact calculation of the Shapley value, which requires the costs for each possible subcoalition of customers - hereinafter referred to as the characteristic function - is only feasible for routing problems with a small number of customers. This limitation underscores the need for Shapley value approximations that deliver accurate results within acceptable computing times (Aziz et al. 2016). It is important to note that allocation methods are utilized not only for monetary cost distribution but also for allocating environmental emissions, a practice expected to gain more importance in the future (Kellner and Schneiderbauer 2019). Further applications of the Shapley value are for example the allocation of costs to carriers in horizontal cooperations (e.g., Gückel et al. 2024, Krajewska et al. 2008) or also in measuring feature importance in machine learning algorithms (e.g., Borgonovo et al. 2024, Cohen et al. 2007).

The literature on Shapley value approximation methods can essentially be divided into two main streams. The first involves statistical sampling methods that use intelligent sampling techniques to approximate Shapley values from samples. The most well-known statistical sampling method is Monte Carlo sampling, introduced by Shapley and Mann (1960), which approximates the Shapley value by averaging marginal contributions of each customer from randomly sampled permutations. Additional statistical sampling procedures are discussed in Castro et al. (2017) and Touati et al. (2021). Furthermore, Schopka and Kopfer (2017) approximated Shapley values by considering marginal costs in smaller subcoalitions. The second stream comprises problem-specific approximation methods developed exclusively for the Traveling Salesman Problem (TSP) within routing problems, which are not easily transferable to other contexts, such as the Capacitated Vehicle Routing Problem (CVRP) or other routing problems (Levinger et al. 2021, Popescu and Kilby 2020). For a comprehensive review of Shapley value approximations in the context of the TSP, we refer to Aziz et al. (2016).

In this study, we propose a novel method for approximating Shapley values by considering weighted Shapley values of smaller subcoalitions. This method is tested for both the TSP and the CVRP. Our contributions are threefold: 1) we develop a new approximation method based on the Shapley values of smaller subcoalitions with exactly β customers, 2) we demonstrate that this approximation method provides good results in both TSP and CVRP with significantly reduced computing times, and 3) we outline broader implications beyond routing problems, particularly in NP-hard problems where evaluating small subcoalitions is much faster than assessing larger instances.

The remainder of the paper is structured as follows: We begin with a description of the problem (Section 4.2), followed by the development of our approximation approach (Section 4.3). We then assess the performance of our method compared to other approximation methods for both the TSP and the CVRP (Section 4.4). Finally, we summarize our findings (Section 4.5).

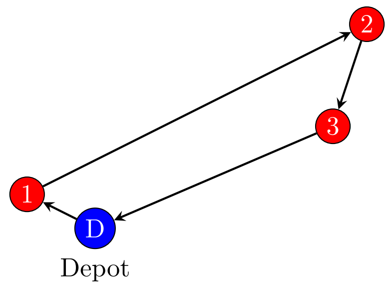

Figure 4.1: Solved TSP instance with three customers.

4.2 Problem description

First, we address the general problem of allocating costs to individual customers in routing problems, which is essential in understanding the complexities of cost allocation mechanisms (Section 4.2.1). Subsequently, we delve into the specifics of the Shapley value and its computational challenges, which are central to developing our proposed approximation method (Section 4.2.2).

4.2.1 Cost allocation in routing problems

Allocating cost to customers in a routing problem involves several complexities. The general problem is exemplified using a TSP instance, as shown in Figure 4.1. The implications discussed, however, are also applicable to other types of routing problems. Here, $\mathcal{N}$ represents the set of all customers of a given instance. The primary challenge in cost allocation lies in finding a fair allocation method, denoted as ϕ_n, that divides the total routing cost $c(\mathcal{N})$ among all customers such that $\sum_{n \in \mathcal{N}} \phi_n = c(\mathcal{N})$. For instance, if costs were allocated based on the distance to the depot, Customer 1 would incur significantly lower costs compared to Customers 2 and 3. This distribution could be perceived as unfair by Customers 2 and 3 due to their close proximity to each other, which results in lower marginal costs. Conversely, distributing costs based on marginal costs might disadvantage Customer 1, who is nearer to the depot, highlighting the disadvantages of simple proportional rules.

To address these challenges, more sophisticated methods within cooperative game theory have been developed. These include the nucleolus (Schmeidler 1969) and core allocations (Gillies 1959), which are designed to provide outcomes that ensure specific fairness criteria. For a comprehensive overview of cost allocation methods in transport logistics, we refer to Guajardo and Rönnqvist (2016). Among these methods, the Shapley value is particularly notable for satisfying numerous fairness properties established in game theory and is thus examined in greater detail in the following section. It is crucial to note that the computational constraints affecting the Shapley value are similarly applicable to these advanced cost allocation methods, as all require the computation of the characteristic function. Therefore, the methodology we develop for the Shapley value could also be beneficially adapted to these other cost allocation frameworks.

4.2.2 Shapley value

The Shapley value, introduced by Shapley (1953), has found diverse applications due to its unique properties compared to other allocation methods. Formally, the Shapley value for a customer n is defined with Equation (4.1):

$$\phi_n^{SV} = \sum_{\mathcal{S} \subset \mathcal{N}: n \in \mathcal{S}} \frac{(|\mathcal{S}|-1)!(|\mathcal{N}|-|\mathcal{S}|)!}{|\mathcal{N}|!}[c(\mathcal{S}) - c(\mathcal{S} - \{n\})] \tag{4.1}$$

The summation is over all subcoalitions $\mathcal{S}$ of the grand coalition $\mathcal{N}$ that include customer n. The term $[c(\mathcal{S}) - c(\mathcal{S} - \{n\})]$ represents the customer's marginal contribution to $c(\mathcal{S})$, and the fraction reflects the weighting of subcoalitions in all possible permutations. The Shapley value is the only cost allocation method that provides a unique cost allocation while simultaneously fulfilling four essential properties: efficiency, symmetry, additivity, and the dummy property.

In addition to these four widely recognized attributes, the Shapley value also enforces advanced properties such as monotonicity, which dictates that a player's cost allocation will not decrease if their contribution to any coalition remains unchanged or increases (Young 1985). Furthermore, it upholds the balanced contribution property, ensuring that claims between players are reciprocally balanced (Myerson 1980). Hence, the Shapley value embodies numerous foundational principles of cooperative game theory, emphasizing diverse fairness dimensions in cost allocation.

However, the exact calculation of the Shapley value requires the evaluation of the characteristic function. For a routing problem with $|\mathcal{N}|$ customers, $2^{|\mathcal{N}|}$ subcoalitions exist and must be evaluated, which is computationally very expensive. Even for routing problems with 15 customers, this necessitates the evaluation of 32,768 subcoalitions of customers, making it computationally intensive. Accordingly, this necessitates methods that can sufficiently approximate the real Shapley value within shorter computing times.

4.3 Methodology

We propose to approximate Shapley values by calculating Shapley values of smaller subcoalitions with exactly β customers. We name our method β Subcoalition Shapley Approximation (β-SSA).

Let $\mathcal{N}$ represent the set of all customers in a routing problem, and $\mathcal{S}_\beta$ the set of subcoalitions with exactly β customers. $\hat{\phi}_n^{SV}$ denotes the Shapley value approximation for customer n, shapley(s, n) is the Shapley value of customer n in subcoalition s, and $c(s)$ represents the total cost of subcoalition $s \in \mathcal{S}_\beta$.

For a routing problem with $\mathcal{N}$ customers, the number of subcoalitions $|\mathcal{S}_\beta|$ with exactly β customers is computed by Equation (4.2):

$$|\mathcal{S}_\beta| = \frac{\mathcal{N}!}{\beta! \cdot (|\mathcal{N}| - \beta)!} \tag{4.2}$$

To approximate Shapley values, we first solve the routing problems i.e., evaluate the cost of all subcoalitions with up to β customers (line 1-3). Then, based on the solved routing problems, we compute the characteristic function of all subcoalitions with exactly β customers and compute the

Shapley value for all of these subcoalitions. For each individual customer, these Shapley values among all of these subcoalitions is summed and weighted by the total routing cost of the respective subcoalitions (line 4-9). At the end, for each individual customer, the share of the cumulated Shapley values among all customers is determined and multiplied by the total routing cost of this instance yielding our final Shapley value approximation $\hat{\phi}_n^{SV}$ (line 10-12).

To conclude, our approximation relies on exactly computing the Shapley values for smaller subcoalitions of up to β customers, and then using their weighted average to approximate the Shapley values for each customer in the full coalition. Our approximation can be represented by Equation (4.3):

$$\hat{\phi}_n^{SV} = \frac{\sum\limits_{s \in \mathcal{S}_\beta} \text{shapley}(s,n) \cdot c(s)}{\sum\limits_{n' \in \mathcal{N}} \sum\limits_{s \in \mathcal{S}_\beta} \text{shapley}(s,n') \cdot c(s)} \cdot c(\mathcal{N}) \tag{4.3}$$

The procedure to obtain our approximations is depicted in Algorithm 6.

Algorithm 6: β-SSA Procedure

Input: Routing Problem with $|\mathcal{N}|$ customers

1 Determine all subcoalitions $\mathcal{S}_{1,\dots,\beta}$ of $\mathcal{N}$ with $1,\dots,\beta$ customers;
2 **for** $s \in \mathcal{S}_{1,\dots,\beta}$ **do**
3     $c(s) \leftarrow \text{solveRoutingProblem}(s)$; `// Evaluate cost of subcoalition` s
4 $\phi_n \leftarrow 0 \forall n \in \mathcal{N}$; `// Initialize cumulated Shapley values for each customer`
5 **for** $s \in \mathcal{S}_\beta$ **do**
6     $v(s) \leftarrow \text{computeCharacteristicFunction}(s)$;
7   **for** $n \in \mathcal{N}(s)$ **do**
8         $\text{shapley}(n,s) \leftarrow \text{computeShapleyValue}(v(s), n)$;
9         $\phi_n \leftarrow \phi_n + \text{shapley}(n,s) \cdot c(s)$; `// Add to cumulated weighted Shapley values`
10 **for** $n \in \mathcal{N}$ **do**
11     $\hat{\phi}_{n'}^{SV} \leftarrow \frac{\phi_n}{\sum\limits_{n' \in \mathcal{N}} \phi_n} \cdot c(\mathcal{N})$; `// Multiply share of cumulated Shapley values by total cost` $c(\mathcal{N})$
12 **return** $\hat{\phi}_n^{SV} \forall n \in \mathcal{N}$; `// Return Shapley value approximation for all customers`

For the special case where $\beta = 1$, the approximation reduces to simply approximating in proportion to the stand-alone cost of each customer. Conversely, for $\beta = |\mathcal{N}|$, it equals the determination of the real Shapley values. Consequently, by increasing β, we enhance the approximation performance but also the computation time as more subcoalitions must be evaluated.

4.4 Numerical study

We benchmark our approximation method in an extensive numerical study with two other sampling approximation methods and a proportional benchmark where the cost are allocated to the customers in proportion to their distance to the depot. We first describe the benchmark approximation methods (Section 4.4.1) and the evaluation metrics (Section 4.4.2). Then we outline the numerical setup (Section 4.4.3) before we report the results (Section 4.4.4).

4.4.1 Benchmark approaches

We consider two different sampling benchmark approaches. The first one is the classical Monte Carlo Sampling (MCS) procedure introduced by Shapley and Mann (1960). Mathematically it is expressed in Equation (4.4):

$$\hat{\phi}_n^{MC} = \frac{1}{s} \sum_{p \in \mathcal{P}^s} c(p_{n-1} \cup \{n\}) - c(p_{n-1}) \tag{4.4}$$

Thereby, $\mathcal{P}^s$ represents s random samples out of the set $\mathcal{P}$ of all permutations of the grand coalition $\mathcal{N}$, and p_{n-1} represents the set of customers that appear in the order of the permutation before customers n. We compute this for each customer by using the same set of random permutations $\mathcal{P}^s$. To avoid memory errors, we generate the random permutations with the Fisher-Yates shuffle algorithm that runs in complexity $O(n)$.
The second one is α-sampling as introduced by Schopka and Kopfer (2017). This is a similar approach to ours as the approximation also takes place via subcoalitions. In contrast to our approach, only the marginal costs of all subcoalitions with α customers are calculated and averaged for each customer and scaled up to the total costs. For a subcoalition s the marginal cost of a customer n is determined by Equation (4.5):

$$c_n(s) = \begin{cases} c(s) - c(s \setminus \{n\}), & \text{if } n \in s \\ c(s \cup \{n\}) - c(s), & \text{if } n \notin s \end{cases} \tag{4.5}$$

For $\mathcal{S}_\alpha$ representing the set of subcoalitions with exactly α customers, the approximation is derived by Equation (4.6):

$$\hat{\phi}_n^{\alpha} = \frac{\sum\limits_{s \in \mathcal{S}_\alpha} c_n(s)}{\sum\limits_{n' \in \mathcal{N}} \sum\limits_{s \in \mathcal{S}_\alpha} c_{n'}(s)} \cdot c(\mathcal{N}) \tag{4.6}$$

Further we consider a proportional benchmark where the cost are allocated in proportion to the distance to the depot. With d_{0i} representing the distance of customer i to the depot node 0, the depot benchmark can be expressed with Equation (4.7):

$$\hat{\phi}_n^{Depot} = c(\mathcal{N}) \cdot \frac{d_{0n}}{\sum\limits_{n' \in \mathcal{N}} d_{0n'}} \tag{4.7}$$

4.4.2 Evaluation metrics

We consider three different evaluation metrics. First, the Mean Absolute Percentage Error (MAPE), which provides a broad measure of the average error magnitude across observations. Second, we use the 95th Percentile Absolute Percentage Error (PAPE95) to assess the robustness of our method. This metric evaluates the performance stability of our approximation method under various conditions by focusing on the higher end of the error distribution, thus indicating how well the method performs even in less favorable scenarios. Lastly, we measure the computation time in seconds per instance for all customers within a routing problem instance. It is important to note that the depot benchmark is excluded from the computation time analysis, as it is computed almost instantaneously.

The MAPE over all customer observations $n \in \mathcal{N}$ is represented with Equation (4.8):

$$\text{MAPE} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \frac{|\hat{\phi}_n - \phi_n^{SV}|}{\phi_n^{SV}} \times 100 \tag{4.8}$$

The PAPE95 can be expressed with Equation (4.9):

$$\text{PAPE95} = \text{Percentile}_{95} \left(\left| \frac{\hat{\phi}_n - \phi_n^{SV}}{\phi_n^{SV}} \right| \times 100, \text{ for } n \in \mathcal{N} \right) \tag{4.9}$$

This selection of metrics, especially the inclusion of PAPE95, allows us to thoroughly evaluate the robustness of β-SSA.

4.4.3 Numerical setup

We generate 200 instances for both the TSP and the CVRP of every instance size (in terms of number of nodes). The customer and depot locations are uniformly randomly distributed, with X and Y coordinates ranging from 0 to 100. This setup yields Euclidean instances with customers randomly distributed in a two-dimensional space, resulting in spatially driven topologies. As instance size increases, so does the complexity, due to the combinatorial growth in the number of feasible routes. For instances with $|\mathcal{N}|$ customers, we then have $|\mathcal{N}| \cdot 200$ labeled observations to benchmark the performance of the approximation methods. All experiments were implemented using the Python programming language version 3.9. Gurobi was used as a solver for the exact evaluation of the routing problems for which we used classical arc-based formulations. To avoid multiple evaluations of the same subcoalition, all evaluated subcoalitions were stored independently in dictionaries for all approximation methods.

4.4.4 Results

We compare the approximations methods for α and β ranging from 2 to 6, and t from 25 to 200 random samples of permutations for the MC-sampling. Due to computational intensity, we report results for instances of up to 16 customers in the TSP and 14 customers in the CVRP.

TSP: The results for the TSP are presented in Tables 4.1 to 4.3. As observed, our proposed β-SSA achieves strong approximation performance, surpassed only by MCS when an extremely large number of permutations is used—resulting in significantly higher computational costs. Thus, β-SSA provides an effective balance between computational efficiency and approximation performance. This trade-off becomes even more pronounced for larger problem instances, where MCS must solve the full-scale problems directly, while our β-SSA concentrates on solving only the subproblems associated with smaller subcoalitions. Furthermore, it can be observed that the simple proportional depot benchmark is substantially outperformed by our approach across all tested instances.

CVRP: The results for the CVRP are shown in Tables 4.4 to 4.6. Similar to the findings for the TSP, our proposed β-SSA demonstrates a favorable trade-off between computational efficiency and

Table 4.1: Performance benchmark in MAPE [%] (TSP)

Nodes	β-SSA					α-sampling					MCS				Depot
	2	3	4	5	6	2	3	4	5	6	25	50	100	200	
8	14.99	10.70	7.82	5.19	2.61	13.43	23.82	33.62	41.41	50.59	16.99	11.68	7.99	5.63	28.45
10	17.70	13.69	11.14	8.88	6.70	15.18	22.26	29.02	33.78	38.87	18.55	12.88	8.88	6.64	28-58
12	19.55	15.72	13.38	11.42	9.56	16.91	21.22	26.12	29.37	32.57	20.91	13.77	10.12	7.02	28.60
14	22.74	18.82	16.49	14.53	12.70	18.77	21.08	24.68	27.23	29.32	21.01	14.46	10.25	7.26	30.67
16	23.77	19.96	17.81	16.02	14.36	20.07	21.98	24.44	25.96	26.93	21.82	15.09	10.41	7.64	30.40

Table 4.2: Performance benchmark in PAPE95 [%] (TSP)

Nodes	β-SSA					α-sampling					MCS				Depot
	2	3	4	5	6	2	3	4	5	6	25	50	100	200	
8	23.16	16.71	11.88	7.92	3.89	26.88	42.56	54.37	63.47	77.85	30.84	21.42	13.10	9.06	64.07
10	27.91	20.21	15.94	12.63	9.73	27.20	36.88	45.15	50.89	58.37	30.98	20.31	13.56	10.66	65.81
12	29.10	22.86	19.21	16.52	14.00	26.97	33.29	39.23	45.94	50.60	33.77	21.81	15.00	11.15	67.87
14	32.04	26.94	22.93	20.31	17.49	29.05	31.80	35.44	39.64	42.16	30.49	21.47	15.82	11.40	70.15
16	32.37	26.99	23.80	21.27	19.06	28.71	31.19	35.18	36.52	39.23	32.56	22.44	15.45	11.94	70.77

Table 4.3: Computation time [s] (TSP)

Nodes	SV	β-SSA					α-sampling					MCS			
		2	3	4	5	6	2	3	4	5	6	25	50	100	200
8	0.18	0.01	0.04	0.09	0.14	0.16	0.04	0.08	0.13	0.12	0.09	0.12	0.15	0.17	0.17
10	1.34	0.02	0.09	0.26	0.55	0.86	0.09	0.26	0.53	0.77	0.88	0.51	0.73	0.98	1.18
12	12.15	0.03	0.17	0.61	1.68	3.39	0.17	0.61	1.66	3.22	5.25	2.81	4.11	5.83	7.85
14	130.30	0.04	0.29	1.26	4.33	10.76	0.29	1.27	4.31	10.51	22.57	16.90	25.40	37.34	53.06
16	1489.30	0.06	0.47	2.36	9.72	28.92	0.48	2.45	9.71	28.62	77.07	106.69	162.90	243.20	355.65

approximation quality. It consistently delivers competitive results while maintaining significantly lower computational requirements.

Table 4.4: Performance benchmark in MAPE [%] (CVRP)

Nodes	β-SSA					α-sampling					MCS				Depot
	2	3	4	5	6	2	3	4	5	6	25	50	100	200	
8	17.87	12.54	9.80	7.40	4.30	15.53	24.60	28.55	32.30	34.74	10.99	9.89	6.83	4.21	24.46
10	19.15	14.70	12.43	10.53	7.93	17.86	25.59	31.14	33.79	36.44	14.79	11.81	8.53	4.39	22.42
12	20.41	17.00	15.13	12.67	9.24	23.51	29.47	26.88	19.19	19.21	17.77	9.08	7.41	5.16	23.47
14	25.07	21.55	18.27	14.95	12.29	24.04	22.44	22.83	20.41	18.34	13.85	10.91	6.91	5.41	23.82

Table 4.5: Performance benchmark in 95PAPE [%] (CVRP)

Nodes	β-SSA					α-sampling					MCS				Depot
	2	3	4	5	6	2	3	4	5	6	25	50	100	200	
8	25.28	19.60	16.87	13.24	7.27	28.51	44.34	44.62	42.50	50.79	15.95	15.12	11.10	6.69	62.22
10	26.88	20.58	17.51	14.99	11.68	27.72	44.15	49.02	54.70	63.78	20.03	15.57	12.69	6.78	57.23
12	30.65	24.95	21.51	18.51	14.93	36.10	41.94	37.77	29.09	28.68	21.84	13.40	10.31	8.55	59.45
14	33.39	26.80	23.27	20.65	17.54	36.04	33.12	32.72	26.74	27.49	19.39	15.96	9.39	6.87	63.04

It can be concluded, that our approach outperforms α-sampling in both approximation accuracy and computation time. While MCS surpasses our method in approximation accuracy with a high

Table 4.6: Computation time [s] (CVRP)

Nodes	SV	β-SSA					α-sampling					MCS			
		2	3	4	5	6	2	3	4	5	6	25	50	100	200
8	1.21	0.02	0.15	0.49	0.78	0.94	0.16	0.49	0.79	0.82	0.54	0.69	0.87	0.99	1.02
10	8.71	0.04	0.38	1.68	3.84	6.07	0.43	1.82	3.92	5.96	6.29	3.03	4.63	6.26	7.89
12	120.58	0.08	1.10	6.02	16.38	34.41	1.11	6.33	16.68	37.83	51.98	25.57	37.47	56.47	78.20
14	1357.49	0.17	3.36	21.61	70.38	169.66	3.62	22.61	77.58	183.05	365.90	168.38	232.64	372.07	544.61

sample count (100 and 200), it demands extensive computation time. In contrast, our approach delivers excellent results within much shorter computation times, making it especially suitable for routing problems with a large number of customers. Moreover, the advantages of our approach extend beyond routing problems. It offers a practical solution for other NP-hard problems where cost allocation to individual items is required. Since evaluating smaller sub-coalitions is computationally far more efficient than handling larger instances, our method represents a feasible and efficient tool for generating cost allocations in complex optimization settings.

4.5 Conclusion

In our study, we developed a novel approximation method for Shapley values based on the Shapley values of smaller subcoalitions. Within an extensive numerical study, we achieved excellent approximation results for routing problems with lower computation time compared to other approximation methods. Our approximation method is not only applicable for routing problems, but is especially suitable for NP-hard underlying subproblems, where the evaluation of small subcoalitions is much less computationally expensive than solving larger instances. Further research could investigate how this approximation method performs with other variants of the routing problems. In addition, the same methodology could also be used for approximations of the Shapley value in completely different contexts, especially when NP-hard problems are behind it, this approximation approach is useful. Finally, it could also be investigated whether other advanced game-theoretic cost allocation methods can be efficiently approximated by this methodology.

Acknowledgments

We would like to express our gratitude to the German Research Foundation (project number: ROCOCO FO1468/1-1) for the financial support of our research project.

Appendix A

Appendix for A two-step large neighborhood search for a collaborative two-tier city logistics system

A.1 MILP notation

Table A.1: Sets, parameters and decision variables

Sets and subsets	
$\mathcal{A}$	set of arcs
$\mathcal{A}(k)$	set of arcs city freighter k can use
$\mathcal{D}$	set of demands
$\mathcal{D}(n)$	set of demands assigned to LSP n
$\delta_k^-(i)$	set of arcs for city freighter k that start in node i
$\delta_k^-(i)$	set of arcs for city freighter k that end in node i
$\mathcal{E}$	set of CDCs
$\mathcal{K}$	set of city freighters
$\mathcal{K}(n)$	set of city freighters of LSP n
$\mathcal{K}(s)$	set of city freighters available at satellite s
$\mathcal{M}$	set of modes of transportation
$\mathcal{N}$	set of all LSPs
P	set of periods
$\mathcal{R}$	set of services
$\mathcal{R}(d,s)$	set of services fulfilling time windows for demand d and satellite s
$\mathcal{R}(n)$	set of services that LSP n operates
$\mathcal{R}(p,e,t)$	set of services of type t, starting from CDC e and operating during period p
$\mathcal{R}(p,s)$	set of services operating at satellite s during period p
$\mathcal{R}(p,s,m)$	set of services of mode m operating at satellite s during period p
$\mathcal{S}$	set of satellite platforms
$\mathcal{T}$	set of urban vehicle types
$\mathcal{T}(m)$	set of urban vehicle types for mode m
Parameters	
a_{spn}	maximum number of urban vehicles that LSP n is allowed to accommodate at satellite s in period p
$\overline{a}_{spmn}$	maximum number of urban vehicles of mode m that LSP n is allowed to accommodate at satellite s in period p
b_d	latest possible delivery time period for demand d
c_r	operating cost of service r
$\hat{c}_{ij}$	cost of city freighter using arc (i,j)
e_r	CDC of urban vehicle service r
f_{de}	assignment cost of demand d to CDC e
g_{spn}	total volume of freight that LSP n is allowed to accommodate at satellite s in period p
h_{etn}	fleet size of urban vehicles at CDC e of type t owned by LSP n
l	service time of city freighters at each demand location
M_1	Big M 1 for Constraints (1.17)
M_2	Big M 2 for Constraints (1.22)
m_r	transportation mode of service r
q	capacity of city freighters
ρ	handling time for each demand at each satellite
s_d	service time at demand location d.
s_k^+	satellite at which city freighter k starts
s_k^-	satellite at which city freighter k ends
$\bar{t}_{ij}$	second-tier travel time between node i and node j
t_r	urban vehicle type of service r
τ_{rs}	arrival time of service r at satellite s
u_t	urban vehicle capacity of type t
v_d	volume of demand d
w_r	service time of service r at each satellite
Decision variables	
p_{ik}	variable specifying the start of service time at vertex i serviced by city freighter k
x_{dsr}	taking the value one if demand d is assigned to satellite s and service r, zero otherwise
y_r	taking the value one if service r is selected, zero otherwise
z_{ijk}	taking the value one if city freighter k uses arc $(i,j) \in \mathcal{A}$, zero otherwise

A.2 Performance benchmark

Table A.2: Comparison of Gurobi and the I2S-LNS with varying $|D|$ and constant $|R| = 24$

Instance*	Costs	GAP [%]	Time [s]	Best costs	Avg. costs	σ [%]	Avg. time [s]	Δ[%]
N1-D5-1	94.55	0.00	1.25	94.55	94.55	0.00	10.24	0.00
N1-D5-2	77.15	0.00	0.50	77.15	77.15	0.00	17.19	0.00
N1-D5-3	121.71	0.00	4.97	121.71	121.71	0.00	19.56	0.00
N1-D5-4	123.86	0.00	2.09	123.86	123.86	0.00	17.95	0.00
N1-D5-5	126.97	0.00	1.41	126.97	126.97	0.00	18.95	0.00
N1-D10-1	173.41	0.00	209.70	173.41	173.41	0.00	48.73	0.00
N1-D10-2	177.29	4.92	3600	177.29	177.29	0.00	40.54	0.00
N1-D10-3	164.92	0.00	1045.30	164.92	164.92	0.00	31.87	0.00
N1-D10-4	166.96	0.93	3600	166.96	166.96	0.00	29.93	0.00
N1-D10-5	165.58	0.00	197.92	165.58	165.58	0.00	35.57	0.00
N1-D15-1	233.22	19.39	3600	228.43	228.96	0.23	102.20	-1.83
N1-D15-2	219.92	16.49	3600	211.45	211.67	0.10	111.50	-3.30
N1-D15-3	241.12	23.00	3600	228.08	228.08	0.00	144.20	-5.41
N1-D15-4	203.75	24.95	3600	203.75	203.75	0.00	94.96	0.00
N1-D15-5	273.66	40.92	3600	259.32	260.30	0.38	82.57	-4.52
N1-D20-1	307.00	32.73	3600	299.35	300.43	0.36	187.74	-2.15
N1-D20-2	268.18	16.76	3600	264.77	264.84	0.03	210.58	-1.25
N1-D20-3	306.51	24.92	3600	276.44	276.44	0.00	218.21	-9.80
N1-D20-4	258.46	28.23	3600	248.62	250.02	0.56	158.62	-3.26
N1-D20-5	287.34	18.48	3600	287.34	287.34	0.00	125.33	0.00
N2-D30-1	396.12	41.11	3600	342.06	343.05	0.29	273.63	-13.41
N2-D30-2	388.48	30.47	3600	382.67	386.09	0.89	486.02	-0.61
N2-D30-3	429.20	31.15	3600	386.41	390.57	1.08	297.51	-8.98
N2-D30-4	406.94	28.96	3600	370.28	371.59	0.35	500.25	-8.69
N2-D30-5	443.28	37.86	3600	376.48	377.10	0.16	467.68	-14.93
N2-D40-1	-	-	3600	487.97	491.03	0.75	600.00	-
N2-D40-2	-	-	3600	484.37	485.43	0.22	592.30	-
N2-D40-3	-	-	3600	492.01	499.20	1.46	573.82	-
N2-D40-4	-	-	3600	488.69	489.23	0.11	513.37	-
N2-D40-5	-	-	3600	480.22	485.74	0.73	442.63	-

*N1-D5-1 refers to the first out of five instances with network N1 and $|\mathcal{D}| = 5$.

Table A.3: Comparison of Gurobi and the I2S-LNS with varying $|R|$ and constant $|D| = 15$

	Gurobi			I2S-LNS				
Instance*	Costs	GAP [%]	Time [s]	Best costs	Avg. costs	σ [%]	Avg. time [s]	Δ [%]
R12-1	268.74	23.39	3600	268.74	268.74	0.00	107.00	0.00
R12-2	213.72	10.20	3600	209.17	209.17	0.00	63.50	-2.13
R12-3	234.14	12.53	3600	224.77	224.77	0.00	85.12	-4.00
R12-4	227.22	16.53	3600	224.20	224.20	0.00	106.96	-1.33
R12-5	206.22	13.40	3600	206.22	206.22	0.00	101.56	0.00
R36-1	203.11	14.76	3600	202.34	202.35	0.00	80.92	-0.37
R36-2	205.21	13.57	3600	205.21	205.21	0.00	95.83	0.00
R36-3	258.75	26.47	3600	236.47	236.48	0.00	165.57	-8.61
R36-4	201.09	13.88	3600	201.09	201.09	0.00	77.94	0.00
R36-5	245.40	34.10	3600	232.16	232.16	0.00	99.09	-5.40
R60-1	206.49	22.48	3600	197.38	197.50	0.06	96.84	-4.35
R60-2	198.17	21.86	3600	191.14	191.14	0.00	110.29	-3.55
R60-3	200.71	17.36	3600	193.26	193.36	0.05	89.83	-3.66
R60-4	179.82	31.15	3600	179.82	179.82	0.00	103.94	-0.00
R60-5	240.74	30.07	3600	232.75	232.75	0.00	145.33	-3.32

*R12-1 refers to the first out of five instances with $|\mathcal{R}| = 12$.

Table A.4: Performance benchmark against a single-step version on larger instances

		I2S-LNS				Single-step version			
$\|D\|$	$\|R\|$	Best costs	Avg. costs	σ [%]	Avg. time [s]	Best costs	Avg. costs	σ [%]	Δ [%]
50	36	552.32	555.85	0.64	1283.70	564.14	568.94	0.85	2.14
50	36	559.20	563.13	0.70	1258.43	569.61	578.61	1.58	1.86
50	36	613.62	615.01	0.23	1434.71	623.31	625.70	0.38	1.58
50	36	626.99	629.19	0.35	1230.52	629.64	644.14	2.30	0.42
50	36	583.67	589.24	0.95	1134.03	599.95	606.95	0.87	2.79
75	48	895.75	904.07	0.93	1800	911.71	919.82	0.89	1.78
75	48	844.22	848.36	0.49	1800	856.59	875.23	2.18	1.47
75	48	873.83	879.79	0.68	1800	894.13	907.89	0.63	2.32
75	48	894.47	898.29	0.43	1740.36	900.97	911.16	0.39	0.73
75	48	880.42	889.61	1.04	1751.60	902.06	913.90	1.31	2.46
100	60	1105.81	1111.72	0.53	1800	1120.08	1141.26	1.89	1.29
100	60	1110.46	1120.61	0.91	1800	1136.92	1144.45	0.66	2.38
100	60	1140.63	1147.24	0.58	1800	1146.18	1154.73	0.75	0.49
100	60	1092.63	1099.54	0.63	1800	1135.46	1138.17	0.24	3.92
100	60	1077.81	1089.62	1.10	1800	1096.16	1113.07	1.54	1.70

A.3 Single-step version

The single-step version of the heuristic is outlined in pseudocode in Algorithm 7. First, a continuous random number p between 0 and 1 is drawn in each iteration. If this random number falls below a threshold θ_3, a destroy-and-repair operation is performed using the operators $\mathcal{O}_1$. These operators correspond to the destroy-and-repair operators described in Section 1.5.4. Otherwise, a destroy-and-repair operation is carried out using the demand-based operators $\mathcal{O}_2$ from Section 1.5.5.

When applying the operators $\mathcal{O}_1$, we allow solutions with an objective value worse than the current solution. However, when using the operators $\mathcal{O}_2$, the current solution is only updated if the objective value falls within a predefined threshold, as is also the case in the second step described in Section 1.5.5. Additionally, the solution memory is incorporated by reverting to one of the stored solutions in memory if it_{max} iterations pass without an update to the global best solution $\mathcal{S}^*$. The functionality of the solution memory remains otherwise identical to that in the two-step version. All other parameters of the heuristic also remain unchanged. The probabilities for selecting operators are analogous to those in Sections 1.5.4 and 1.5.5. The parameters θ_3 and it_{max} were tuned in the same manner as described in Section 1.7.2.1 and are set to $\theta_3 = 0.015$ and $it_{max} = 1500$. Please note that we tune θ_3 as a fixed parameter, as adaptivity would strongly disadvantage service operators and would not be fair, since the modification of the service design often makes the solutions much worse in the short term, but can be advantageous in the long term. For a fair comparison, we allow the algorithm to run for the entire runtime without imposing an early stopping criterion.

Algorithm 7: Single-step version

$\mathcal{S}_c, \mathcal{S}^* \leftarrow$ Generate a feasible initial solution; `// Section 1.5.3`
$it \leftarrow 0$;
while *Time limit not reached* **do**
  $p \leftarrow \text{randomNumber}(0, 1)$;
  if $p < \theta_3$ **then**
    $\mathcal{S}_c \leftarrow \text{DestroyandRepair}(\mathcal{S}_c, \mathcal{O}_1)$; `// Apply destroy and repair operators on the service design`
  else
    $\mathcal{S}'_c \leftarrow \text{DestroyandRepair}(\mathcal{S}_c, \mathcal{O}_2)$; `// Apply destroy and repair operators on the demands`
    if $f(\mathcal{S}'_c) < (1 + \theta_w) \cdot f(\mathcal{S}_c)$ **then**
      $\mathcal{S}_c \leftarrow \mathcal{S}'_c$; `// Update current solution if threshold is met`
  if $f(\mathcal{S}_c) > f(\mathcal{S}^*)$ **then** $it \leftarrow it + 1$;
  else $it \leftarrow 0$; $\mathcal{S}^* \leftarrow \mathcal{S}_c$;
  if $it > it_{max}$ **then**
    $\mathcal{S}_c \leftarrow \text{drawRandomElement}(\Gamma)$; $it \leftarrow 0$; `// Go back to one of the solutions in Γ`
  $\Gamma \leftarrow \text{update}(\Gamma, \mathcal{S}_c)$;

Appendix B

Appendix for Tactical planning in cooperative two-tier city logistics systems with fairness constraints

B.1 Notation table

Table B.1: Summary of Notation

Symbol	Description
Sets	
$\mathcal{N}$	Set of logistics service providers
$\mathcal{E}$	Set of city distribution centers
$\mathcal{Z}$	Set of satellites
$\mathcal{R}$	Set of services from CDCs to satellites
$\mathcal{D}$	Set of demands
$\mathcal{M}$	Set of vehicle types
$\mathcal{T}$	Set of days in the schedule
$\mathcal{P}$	Set of time periods per day
Subsets	
$\mathcal{D}(n)$	Demands contributed by LSP n
$\mathcal{R}(n)$	Services contributed by LSP n
$\mathcal{D}(t)$	Demands occurring on day t
$\mathcal{R}^D(d, z)$	Services satisfying release/due date of demand d at satellite z
$\mathcal{R}^P(p, z)$	Services operating in period p at satellite z
$\mathcal{R}(p, e, m)$	Services in period p, from CDC e, using vehicle type m
Parameters	
m_r	Vehicle type assigned to service r
e_r	Origin CDC of service r
u_m	Capacity of vehicle type m
v_d	Volume of demand d
a_{zpn}	Max. number of services LSP n can operate at z in period p
b_{zpn}	Max. volume LSP n can handle at z in period p
h_{emn}	Number of vehicles of type m owned by LSP n at CDC e
c_r	Operating cost of service r (incl. environmental effects)
s_{dzr}	Approx. cost of assigning demand d to satellite z via service r
f_{de}	Cost of delivering demand d from its origin to CDC e
α	Min. share of an LSP's own demand volume to be fulfilled by its own services
β	Min. ratio of an LSP's assigned demands to its contributed demands
γ	Allowed deviation in cost allocation from equal relative savings
θ	Required proportion of services to operate on all days
C_n	Standalone cost of LSP n without cooperation
$C_{\mathcal{N}}$	Total standalone cost: $\sum_{n \in \mathcal{N}} C_n$
C_{nt}	Standalone cost of LSP n on day t
$C_{\mathcal{N}t}$	Total standalone cost on day t: $\sum_{n \in \mathcal{N}} C_{nt}$
Decision Variables	
y_{rt}	1 if service r operates on day t, 0 otherwise
x_{dzr}	1 if demand d is assigned to satellite z and service r, 0 otherwise
$\hat{Y}_r$	1 if service r operates on all days $t \in \mathcal{T}$, 0 otherwise

B.2 Impact of fairness constraints on financial and workload aspects for LSPs with different demand patterns

We replicate all experiments of Section 2.5.5 on a *patterned-demand instance set* comprising the same settings as the instances used before. In each instance, LSP 1 has a front-loaded demand profile (25, 25, 20, 10, 5 demands on days 1–5), while LSP 2 and LSP 3 each have constant 15 demands per day.

Figures B.1, B.2, B.3, B.4, and B.5 report the results. In the "2&3" label, LSP 2 and LSP 3 are grouped because they share identical demand profiles and exhibit very similar distributions.

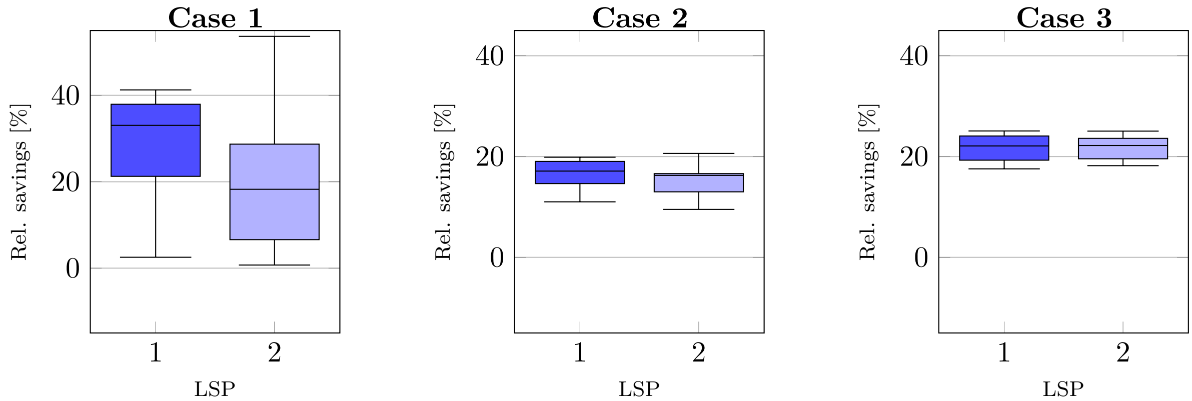

Figure B.1: Relative cost savings of LSPs for the different cases [%]

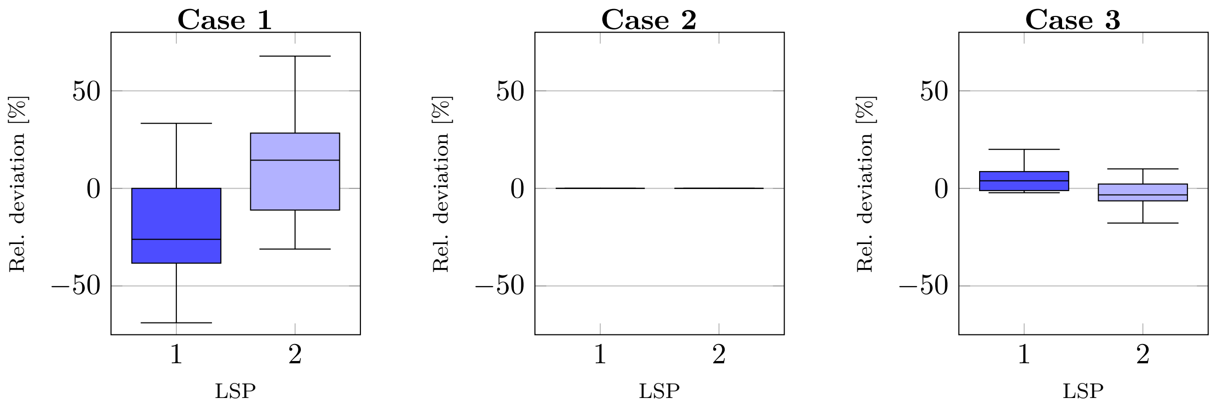

Figure B.2: Relative deviation in the number of assigned demands for the different cases [%]

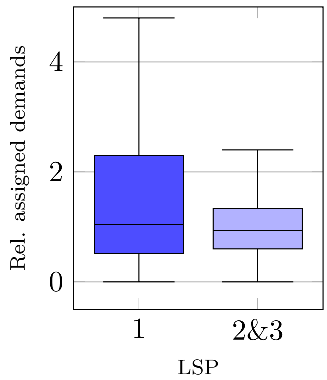

Figure B.3: Distribution of the daily number of demands assigned to own services relative to the own number of demands for $\beta = 1$

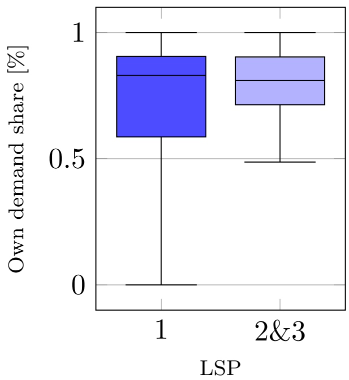

Figure B.4: Distribution of the daily share of own demand volume assigned to own services for $\alpha = 0.8$

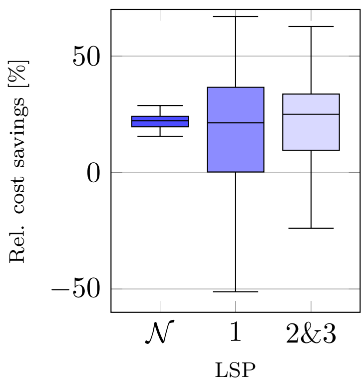

Figure B.5: Distribution of daily relative cost savings [%] for $\gamma = 1\%$

Appendix C

Appendix for Fast Shapley value approximation through machine learning with application in routing problems

C.1 Mathematical model formulations

C.1.1 TSP formulation

Given a complete graph $G = (\mathcal{N}, \mathcal{A})$ with the set of nodes $\mathcal{N}$ and the set of arcs $\mathcal{A}$, where each arc $(i, j) \in \mathcal{A}$ has an associated non-negative cost c_{ij}, and where x_{ij} is a binary decision variable that takes the value 1 if arc (i, j) is used in the tour and 0 otherwise, the TSP can be formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{i,j \in \mathcal{A}} c_{ij} \cdot x_{ij} & \\
& \sum_{i \in \mathcal{N}: i \neq j} x_{ij} = 1 & \forall j \in \mathcal{N} \\
& \sum_{j \in \mathcal{N}: j \neq i} x_{ij} = 1 & \forall i \in \mathcal{N} \\
& \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}: j \neq i} x_{ij} \leq |\mathcal{S}| - 1 & \forall \mathcal{S} \subset \mathcal{N}, |\mathcal{S}| \geq 2 \\
& x_{ij} \in \{0, 1\} & \forall i, j \in \mathcal{A}
\end{aligned}
$$

C.1.2 CVRP formulation

Given a complete graph $G = (\mathcal{N}, \mathcal{A})$ with the set of nodes $\mathcal{N}$ and the set of arcs $\mathcal{A}$, where each arc $(i, j) \in \mathcal{A}$ has an associated non-negative cost c_{ij}, and each node a certain demand volume q_i, and where x_{ij} is a binary decision variable that takes the value 1 if arc (i, j) is used in the tour and 0 otherwise, and u_i represents a decision variable for the cumulative demand up to node i, and Q is the vehicle capacity, the CVRP can be formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{i,j \in \mathcal{A}} c_{ij} \cdot x_{ij} & \\
& \sum_{j \in N: j \neq i} x_{ij} = 1 & \forall i \in \mathcal{N} \\
& \sum_{i \in N: i \neq j} x_{ij} = 1 & \forall j \in \mathcal{N} \\
& x_{ij} = 1 \Rightarrow u_i + q_j = u_j & \forall i, j \in \mathcal{A} : j \neq 0, i \neq 0 \\
& q_i \leq u_i \leq Q & \forall i \in \mathcal{N} \\
& x_{ij} \in \{0, 1\} & \forall i, j \in \mathcal{A}
\end{aligned}
$$

C.1.3 Bin packing with size and weight formulation

Given a set of bins $\mathcal{B}$ and a set of items $\mathcal{I}$. Let w_i and s_i be the weight and size of item $i \in \mathcal{I}$ and W and S be the uniform weight and size capacity for each bin $b \in \mathcal{B}$. With the binary decision variables y_b indicating whether a bin $b \in \mathcal{B}$ is selected and x_{ib} indicating whether item $i \in \mathcal{I}$ is packed into bin $b \in \mathcal{B}$, where M is a sufficiently large constant, the problem can be modeled as follows:

$$\begin{aligned}
\min \quad & \sum_{b \in \mathcal{B}} y_b \\
& \sum_{b \in \mathcal{B}} x_{ib} = 1 && \forall i \in \mathcal{I} \\
& \sum_{i \in \mathcal{I}} x_{ib} \leq M \cdot y_b && \forall b \in \mathcal{B} \\
& \sum_{i \in \mathcal{I}} x_{ib} \cdot w_i \leq W && \forall b \in \mathcal{B} \\
& \sum_{i \in \mathcal{I}} x_{ib} \cdot s_i \leq S && \forall b \in \mathcal{B} \\
& y_b \in \{0, 1\} && \forall b \in \mathcal{B} \\
& x_{ib} \in \{0, 1\} && \forall i \in \mathcal{I}, b \in \mathcal{B}
\end{aligned}$$

C.2 Hyperparameters

Table C.1: Configuration of neural network

Hyperparameter	TSP	CVRP	Bin Packing
Number of hidden layers	4	3	4
Units per Layer	[208, 504, 32, 40]	[408, 184, 160]	[208, 136, 296, 8]
Activation function	[tanh, relu, relu, relu]	[relu, relu, relu]	[relu, tanh, relu, relu]
Learning rate	0.0017	0.0032	0.0016
Epochs*	200	200	200
Optimizer	adam	adam	adam

**Tuned individually for the different MLSVA modifications.*

Table C.2: Configuration of random forest

Hyperparamter	TSP	CVRP	Bin Packing
nEstimators	350	350	400
maxFeatures	None	None	None
maxDepth	25	28	25
minSamplesLeaf	2	2	2
bootstrap	True	True	True

Table C.3: Configuration of xgboost

Hyperparameter	TSP	CVRP	Bin Packing
nEstimators	250	300	273
maxDepth	11	10	16
minChildWeight	10	12	10
subsample	0.58	1.0	1.0
learningRate	0.06	0.07	0.05
lambda	1.7	5.0	4.99
alpha	0.0	0.0	0.25
booster	dart	gbtree	gbtree
colsampleBytree	1	1	0.77

Table C.4: Configuration of polynomial regression

Hyperparameter	TSP	CVRP	Bin Packing
Polynomial Features Degree	3	3	3
Ridge Alpha	1	0.1	0.1

Table C.5: Configuration of support vector regression

Hyperparameter	TSP	CVRP	Bin Packing
C	100	10	0.1
Epsilon	0.1	0.01	0.01
maxiter	200 000	200 000	200 000

Table C.6: Configuration of KNN

Hyperparamter	TSP	CVRP	Bin Packing
K	10	14	20

C.3 Robustness of approximations

In this section, we report the 95th Percentile Absolute Percentage Error (PAPE95) for different approximation methods. This metric provides insight into the robustness of each method by indicating the absolute percentage error that is not exceeded in 95% of the cases.

The PAPE95 is formally defined with Expression (C.1):

$$\text{PAPE95} = \text{Percentile}_{95}\left(\left|\frac{\hat{\phi}_n - \phi_n^{SV}}{\phi_n^{SV}}\right| \times 100, \text{ for } n \in \mathcal{N}\right) \tag{C.1}$$

Tables C.7 to C.9 show the results for the TSP, CVRP and Bin Packing.

Table C.7: PAPE95 [%] for MLSVA differentiated by ML Models and Benchmarks (TSP)

	MLSVA						Benchmarks				
No. of nodes	**NN**	**RF**	**XGB**	**PR**	**SVR**	**KNN**	**SHAPO**	**Depot**	**Reroute**	**ACAM**	**MCS**
7	**4.56**	18.16	11.18	8.48	11.58	76.02	5.59	63.94	114.48	37.52	17.94
8	**4.67**	19.27	11.32	8.10	9.62	69.45	7.96	66.78	120.39	42.18	19.44
9	**5.34**	20.90	11.16	8.52	11.31	63.86	10.03	66.99	132.49	46.50	20.25
10	**5.78**	20.69	11.45	8.88	10.57	55.48	11.82	67.32	143.72	48.52	21.39
11	**6.57**	20.99	11.88	10.03	11.10	52.68	13.72	67.83	148.94	51.19	21.92
12	**7.63**	21.04	12.59	11.45	12.24	48.39	15.36	67.07	153.54	53.78	22.16
13	**8.27**	21.47	12.71	12.92	13.00	49.77	17.27	67.56	161.40	54.18	22.92
14	**9.14**	22.24	13.56	13.17	14.04	45.82	18.95	68.86	155.97	53.35	22.72
15	**9.90**	23.46	14.72	14.24	14.77	44.90	21.01	68.02	163.06	55.36	23.97
avg.	**6.87**	20.92	12.29	10.64	12.03	56.26	13.52	67.15	143.78	49.18	21.41

Table C.8: PAPE95 [%] for MLSVA differentiated by ML Models and Benchmarks (CVRP)

	MLSVA						Benchmarks			
No. of nodes	**NN**	**RF**	**XGB**	**PR**	**SVR**	**KNN**	**Depot**	**Reroute**	**ACAM**	**MCS**
7	**9.48**	18.45	13.03	13.84	16.16	76.87	59.08	102.94	82.10	14.40
8	**9.42**	17.88	13.54	12.39	13.34	67.58	57.32	99.99	82.02	14.78
9	**10.01**	18.48	13.42	12.50	13.03	59.48	61.35	99.93	72.61	14.35
10	**9.58**	18.11	13.70	11.63	12.79	53.87	61.36	99.84	71.70	14.36
11	**10.39**	19.34	13.88	12.19	12.89	57.16	63.56	99.85	74.61	14.49
12	**10.85**	19.01	13.88	12.03	13.01	52.31	66.80	99.79	74.50	15.12
13	**11.06**	18.91	14.06	12.72	13.91	54.18	68.20	99.87	76.72	14.78
avg.	**10.11**	18.60	13.64	12.47	13.59	60.21	62.52	100.32	76.32	14.61

Table C.9: PAPE95 [%] for MLSVA differentiated by ML Models (Bin Packing)

No. of nodes	NN	RF	XGB	PR	SVR	KNN	WS
7	**9.64**	11.93	11.19	13.37	12.62	20.32	61.28
8	**11.21**	13.34	11.88	13.31	12.74	17.90	58.49
9	**11.54**	13.48	11.98	12.94	12.53	17.68	57.76
10	**10.97**	13.11	11.30	12.61	12.09	15.94	54.55
11	11.26	12.14	**10.53**	12.08	11.41	15.14	56.91
12	**10.06**	11.93	10.32	11.42	10.86	14.10	59.96
13	11.21	12.74	**11.13**	12.60	12.05	14.87	59.83
14	10.18	11.22	**9.86**	11.45	10.20	12.73	58.69
15	10.37	11.87	**9.96**	11.43	10.45	13.40	60.69
avg.	**10.72**	12.42	10.91	12.36	11.66	15.79	58.68

C.4 Detailed Illustration

C.4.1 TSP Illustration

In the following, we provide a detailed breakdown of a 7-node TSP instance, consisting of one depot and six customers. The optimally solved instance is visualized in Figure C.1.

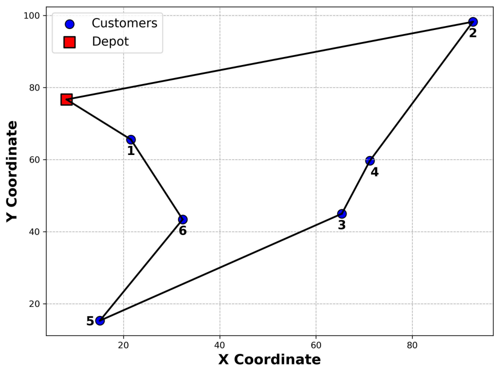

Figure C.1: Optimally solved 7-node TSP instance (one depot and six customers)

The corresponding distance matrix is presented in Table C.10, while a subset of the characteristic function, showing optimal cost values for selected coalitions, is displayed in Table C.11.

Table C.10: Distance matrix for the 7-node TSP instance

ID	0	1	2	3	4	5	6
0	0.00	17.39	87.14	65.40	65.29	61.76	41.10
1	17.39	0.00	78.21	48.42	50.01	50.68	24.63
2	87.14	78.21	0.00	59.82	44.10	113.51	81.52
3	65.40	48.42	59.82	0.00	15.82	58.37	33.11
4	65.29	50.01	44.10	15.82	0.00	71.54	42.18
5	61.76	50.68	113.51	58.37	71.54	0.00	32.94
6	41.10	24.63	81.52	33.11	42.18	32.94	0.00

Finally, Table C.12 presents the true Shapley values alongside their approximations obtained from different methods.

Table C.11: Characteristic function: Selected coalition costs

Coalition (Subset)	Total Cost
$\{1\}$	34.77
$\{2\}$	174.29
$\{3\}$	130.79
$\{1,2\}$	182.74
$\{1,3\}$	131.20
$\{2,3\}$	212.36
$\{1,2,3\}$	212.78
$\{1,2,3,4\}$	212.87
$\{1,2,3,4,5\}$	273.50
$\{1,2,3,4,5,6\}$	280.40

Table C.12: True Shapley values and approximations by different methods for TSP

Method	1	2	3	4	5	6
Shapley value	8.72	99.41	39.25	38.48	71.36	23.18
MLSVA (NN)	8.77	99.94	39.68	38.62	70.59	22.81
SHAPO	7.73	98.51	38.31	38.59	72.98	24.28
MCS	8.99	99.09	51.54	27.74	69.26	23.78
Depot	14.42	72.28	54.24	54.15	51.23	34.09
Reroute	10.60	133.17	11.47	0.19	111.74	13.25
ACAM	12.93	95.96	37.61	33.16	74.76	25.98

C.4.2 CVRP Illustration

In the following, we provide a detailed breakdown of a 7-node CVRP instance, consisting of one depot and six customers. The vehicle capacity in this instance is set to 13. The optimally solved instance is visualized in Figure C.2.

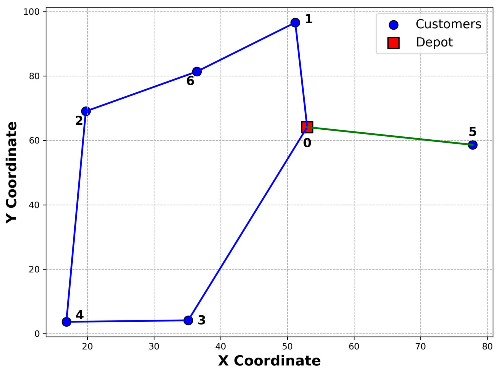

Figure C.2: Optimally solved 7-node CVRP instance (one depot and six customers)

The corresponding distance matrix and the demands q of the customers are presented in Table C.13, while a subset of the characteristic function, showing optimal cost values for selected coalitions, is displayed in Table C.14.

Finally, Table C.15 presents the true Shapley values alongside their approximations obtained from different methods.

C.5 Visualisation of modified instances

The heatmap of the modified instances is shown in Figure C.3.

Table C.13: Distance matrix for the 7-node CVRP instance with demands

ID	**Demand**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0.00	32.52	33.59	62.58	70.42	25.43	23.92
1	4	32.52	0.00	41.80	93.84	99.06	46.36	21.20
2	1	33.59	41.80	0.00	66.69	65.43	58.98	20.74
3	1	62.58	93.84	66.69	0.00	18.29	69.22	77.25
4	3	70.42	99.06	65.43	18.29	0.00	82.08	80.14
5	3	25.43	46.36	58.98	69.22	82.08	0.00	47.22
6	3	23.92	21.20	20.74	77.25	80.14	47.22	0.00

Table C.14: Characteristic function: Selected coalition costs for CVRP

Coalition (Subset)	**Total Cost**
$\{1\}$	65.04
$\{2\}$	67.17
$\{3\}$	125.16
$\{1, 2\}$	107.91
$\{1, 3\}$	188.94
$\{2, 3\}$	162.86
$\{1, 2, 3\}$	203.59
$\{1, 2, 3, 4\}$	220.62
$\{1, 2, 3, 4, 5\}$	252.69
$\{1, 2, 3, 4, 5, 6\}$	271.61

Table C.15: True Shapley values and approximations by different methods for CVRP

Method	**1**	**2**	**3**	**4**	**5**	**6**
Shapley value	45.68	28.27	59.39	74.94	41.69	21.64
MLSVA (NN)	46.81	28.14	60.02	73.86	41.02	21.76
MCS	44.50	33.00	60.28	71.94	41.15	20.75
Depot	35.55	36.72	68.41	76.99	27.80	26.15
Reroute	75.30	16.46	16.19	55.50	78.82	29.34
ACAM	53.52	27.56	44.81	67.28	50.86	27.59

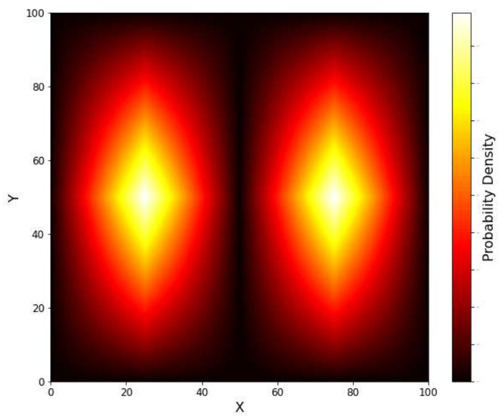

Figure C.3: Heatmap of probability distribution of modified instances

Bibliography

Agarwal, R., Ergun, Ö., 2010. Network design and allocation mechanisms for carrier alliances in liner shipping. Operations Research 58 (6), 1726–1742.

Altan, B., Özener, O. Ö., 2019. Cost allocation mechanisms in a peer-to-peer network. Networks 73 (1), 104–118.

Asín-Achá, R., Espinoza, A., Goldschmidt, O., Hochbaum, D. S., Huerta, I. I., 2024. Selecting fast algorithms for the capacitated vehicle routing problem with machine learning techniques. Networks 84 (2), 200–219.

Aziz, H., Cahan, C., Gretton, C., Kilby, P., Mattei, N., Walsh, T., 2016. A study of proxies for Shapley allocations of transport costs. Journal of Artificial Intelligence Research 56, 573–611.

Bai, R., Chen, X., Chen, Z.-L., Cui, T., Gong, S., He, W., Jiang, X., Jin, H., Jin, J., Kendall, G., et al., 2023. Analytics and machine learning in vehicle routing research. International Journal of Production Research 61 (1), 4–30.

Baldacci, R., Mingozzi, A., Roberti, R., Calvo, R. W., 2013. An exact algorithm for the two-echelon capacitated vehicle routing problem. Operations Research 61 (2), 298–314.

Basso, R., Kulcsár, B., Sanchez-Diaz, I., 2021. Electric vehicle routing problem with machine learning for energy prediction. Transportation Research Part B: Methodological 145, 24–55.

Bendel, D., Haviv, M., 2018. Cooperation and sharing costs in a tandem queueing network. European Journal of Operational Research 271 (3), 926–933.

Bengio, Y., Lodi, A., Prouvost, A., 2021. Machine learning for combinatorial optimization: A methodological tour d'horizon. European Journal of Operational Research 290 (2), 405–421.

Bertsimas, D., Kim, C. W., 2023. A prescriptive machine learning approach to mixed-integer convex optimization. INFORMS Journal on Computing 35 (6), 1225–1241.

Bonku, R., Alkaabneh, F., Davis, L. B., 2025. Collaborative vehicle routing for equitable and effective food allocation in nonprofit settings. Journal of Humanitarian Logistics and Supply Chain Management 15 (2), 135–153.

Borgonovo, E., Plischke, E., Rabitti, G., 2024. The many shapley values for explainable artificial intelligence: A sensitivity analysis perspective. European Journal of Operational Research 318 (3), 911–926.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Breunig, U., Schmid, V., Hartl, R. F., Vidal, T., 2016. A large neighbourhood based heuristic for two-echelon routing problems. Computers & Operations Research 76, 208–225.

Browne, M., Allen, J., Nemoto, T., Patier, D., Visser, J., 2012. Reducing social and environmental impacts of urban freight transport: A review of some major cities. Procedia-Social and Behavioral Sciences 39, 19–33.

Bruni, M. E., Fadda, E., Fedorov, S., Perboli, G., 2023. A machine learning optimization approach for last-mile delivery and third-party logistics. Computers & Operations Research 157, 106262.

Bruni, M. E., Khodaparasti, S., Perboli, G., 2024. A bi-level approach for last-mile delivery with multiple satellites. Transportation Research Part C: Emerging Technologies 160, 104495.

Castro, J., Gómez, D., Molina, E., Tejada, J., 2017. Improving polynomial estimation of the shapley value by stratified random sampling with optimum allocation. Computers & Operations Research 82, 180–188.

Castro, J., Gómez, D., Tejada, J., 2009. Polynomial calculation of the Shapley value based on sampling. Computers & Operations Research 36 (5), 1726–1730.

Cattaruzza, D., Absi, N., Feillet, D., González-Feliu, J., 2017. Vehicle routing problems for city logistics. EURO Journal on Transportation and Logistics 6 (1), 51–79.

Cepolina, E. M., Farina, A., 2015. A new urban freight distribution scheme and an optimization methodology for reducing its overall cost. European Transport Research Review 7 (1), 1–14.

Chamurally, S., Rieck, J., 2024. A practical and robust approach for solving the multi-compartment vehicle routing problem under demand uncertainty using machine learning. Networks 84 (3), 300–325.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

Choudhury, D., Borkotokey, S., Kumar, R., Sarangi, S., 2021. The egalitarian Shapley value: A generalization based on coalition sizes. Annals of Operations Research 301 (1), 55–63.

Ciardiello, F., Genovese, A., Simpson, A., 2020. A unified cooperative model for environmental costs in supply chains: The Shapley value for the linear case. Annals of Operations Research 290, 421–437.

Cohen, S., Dror, G., Ruppin, E., 2007. Feature selection via coalitional game theory. Neural Computation 19 (7), 1939–1961.

Costa, L., Contardo, C., Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transportation Science 53 (4), 946–985.

Crainic, T. G., 2000. Service network design in freight transportation. European Journal of Operational Research 122 (2), 272–288.

Crainic, T. G., Errico, F., Rei, W., Ricciardi, N., 2016. Modeling demand uncertainty in two-tier city logistics tactical planning. Transportation Science 50 (2), 559–578.

Crainic, T. G., Feliu, J. G., Ricciardi, N., Semet, F., Van Woensel, T., 2023. Operations research for planning and managing city logistics systems. In: Handbook on City Logistics and Urban Freight. Edward Elgar Publishing, pp. 190–223.

Crainic, T. G., Gendreau, M., Jemai, L., 2020. Planning hyperconnected, urban logistics systems. Transportation Research Procedia 47, 35–42.

Crainic, T. G., Kim, K. H., 2007. Intermodal transportation. Handbooks in Operations Research and Management Science 14, 467–537.

Crainic, T. G., Laporte, G., 1997. Planning models for freight transportation. European Journal of Operational Research 97 (3), 409–438.

Crainic, T. G., Perboli, G., Ricciardi, N., 2021. City logistics. In: Network Design with Applications to Transportation and Logistics. Springer, pp. 505–534.

Crainic, T. G., Ricciardi, N., Storchi, G., 2004. Advanced freight transportation systems for congested urban areas. Transportation Research Part C: Emerging Technologies 12 (2), 119–137.

Crainic, T. G., Ricciardi, N., Storchi, G., 2009. Models for evaluating and planning city logistics systems. Transportation Science 43 (4), 432–454.

Crainic, T. G., Sgalambro, A., 2014. Service network design models for two-tier city logistics. Optimization Letters 8 (4), 1375–1387.

Cruijssen, F., Bräysy, O., Dullaert, W., Fleuren, H., Salomon, M., 2007a. Joint route planning under varying market conditions. International Journal of Physical Distribution & Logistics Management 37 (4), 287–304.

Cruijssen, F., Cools, M., Dullaert, W., 2007b. Horizontal cooperation in logistics: Opportunities and impediments. Transportation Research Part E: Logistics and Transportation Review 43 (2), 129–142.

Cruijssen, F., Dullaert, W., Fleuren, H., 2007c. Horizontal cooperation in transport and logistics: A literature review. Transportation Journal 46 (3), 22–39.

De-Arteaga, M., Feuerriegel, S., Saar-Tsechansky, M., 2022. Algorithmic fairness in business analytics: Directions for research and practice. Production and Operations Management 31 (10), 3749–3770.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., 1996. Support vector regression machines. Advances in neural information processing systems 9.

Elshaer, R., Awad, H., 2020. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. Computers & Industrial Engineering 140, 106242.

Elting, S., Ehmke, J. F., Gansterer, M., 2025. Preference learning for efficient bundle selection in horizontal transport collaborations. European Journal of Operational Research 324 (3), 953–968.

Engevall, S., Göthe-Lundgren, M., Värbrand, P., 1998. The traveling salesman game: An application of cost allocation in a gas and oil company. Annals of Operations Research 82 (0), 203–218.

Engevall, S., Göthe-Lundgren, M., Värbrand, P., 2004. The heterogeneous vehicle-routing game. Transportation Science 38 (1), 71–85.

Ermağan, U., Yıldız, B., Salman, F. S., 2022. A learning based algorithm for drone routing. Computers & Operations Research 137, 105524.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Vol. 96. pp. 226–231.

EU Commission, 2001. Guidelines on the applicability of Article 81 of the EC treaty to horizontal cooperation agreements. Official Journal of the European Communities 100, 2–30.

European Commission, 2021. The New EU Urban Mobility Framework. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions, COM(2021) 811 final.

Fernández, E., Roca-Riu, M., Speranza, M. G., 2018. The shared customer collaboration vehicle routing problem. European Journal of Operational Research 265 (3), 1078–1093.

Fontaine, P., Crainic, T. G., Jabali, O., Rei, W., 2021. Scheduled service network design with resource management for two-tier multimodal city logistics. European Journal of Operational Research 294 (2), 558–570.

Fontaine, P., Minner, S., Schiffer, M., 2023. Smart and sustainable city logistics: Design, consolidation, and regulation. European Journal of Operational Research 307 (3), 1071–1084.

Frisk, M., Göthe-Lundgren, M., Jörnsten, K., Rönnqvist, M., 2010. Cost allocation in collaborative forest transportation. European Journal of Operational Research 205 (2), 448–458.

Galambos, K. J., Palomino-Hernández, A. B., Hemmelmayr, V. C., Turan, B., 2024. Sustainability initiatives in urban freight transportation in Europe. Transportation Research Interdisciplinary Perspectives 23, 101013.

Gansterer, M., Hartl, R. F., 2016. Request evaluation strategies for carriers in auction-based collaborations. OR Spectrum 38, 3–23.

Gansterer, M., Hartl, R. F., 2018. Collaborative vehicle routing: A survey. European Journal of Operational Research 268 (1), 1–12.

Gansterer, M., Hartl, R. F., Salzmann, P. E., 2018. Exact solutions for the collaborative pickup and delivery problem. Central European Journal of Operations Research 26, 357–371.

Gansterer, M., Hartl, R. F., Savelsbergh, M., 2020a. The value of information in auction-based carrier collaborations. International Journal of Production Economics 221, 107485.

Gansterer, M., Hartl, R. F., Sörensen, K., 2020b. Pushing frontiers in auction-based transport collaborations. Omega 94, 102042.

Gansterer, M., Hartl, R. F., Vetschera, R., 2019. The cost of incentive compatibility in auction-based mechanisms for carrier collaboration. Networks 73 (4), 490–514.

Ghiani, G., Manni, A., Manni, E., 2022. A scalable anticipatory policy for the dynamic pickup and delivery problem. Computers & Operations Research 147, 105943.

Gianessi, P., Alfandari, L., Létocart, L., Wolfler Calvo, R., 2016. The multicommodity-ring location routing problem. Transportation Science 50 (2), 541–558.

Gillies, D. B., 1959. Solutions to general non-zero-sum games. In: Contributions to the Theory of Games IV. Vol. 40. Princeton University Press, pp. 47–85.

Gmira, M., Gendreau, M., Lodi, A., Potvin, J.-Y., 2020. Travel speed prediction based on learning methods for home delivery. EURO Journal on Transportation and Logistics 9 (4), 100006.

Gnecco, G., Hadas, Y., Sanguineti, M., 2019. Some properties of transportation network cooperative games. Networks 74 (2), 161–173.

Gómez-Rodríguez, M., Davila-Pena, L., Casas-Méndez, B., 2024. Cost allocation problems on highways with grouped users. European Journal of Operational Research 316 (2), 667–679.

Göthe-Lundgren, M., Jörnsten, K., Värbrand, P., 1996. On the nucleolus of the basic vehicle routing game. Mathematical Programming 72, 83–100.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.-M., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European Journal of Operational Research 254 (1), 80–91.

Groß, P.-O., Ehmke, J. F., Mattfeld, D. C., 2020. Interval travel times for robust synchronization in city logistics vehicle routing. Transportation Research Part E: Logistics and Transportation Review 143, 102058.

Guajardo, M., Rönnqvist, M., 2016. A review on cost allocation methods in collaborative transportation. International Transactions in Operational Research 23 (3), 371–392.

Gückel, J., Crainic, T. G., Fontaine, P., 2024. Resource Planning and Cost Allocation for Tactical Planning in Cooperative Two-Tier City Logistics Systems. Publication Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Montréal, QC, Canada.

Hastie, T., Tibshirani, R., Friedman, J. H., 2009. The elements of statistical learning: Data mining, inference, and prediction. Vol. 2. Springer.

Hemmelmayr, V. C., Cordeau, J.-F., Crainic, T. G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Computers & Operations Research 39 (12), 3215–3228.

Holguín-Veras, J., Leal, J. A., Sánchez-Diaz, I., Browne, M., Wojtowicz, J., 2020. State of the art and practice of urban freight management: Part I: Infrastructure, vehicle-related, and traffic operations. Transportation Research Part A: Policy and Practice 137, 360–382.

Irannezhad, E., Prato, C. G., Hickman, M., 2018. The effect of cooperation among shipping lines on transport costs and pollutant emissions. Transportation Research Part D: Transport and Environment 65, 312–323.

Jepsen, M., Spoorendonk, S., Ropke, S., 2013. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. Transportation Science 47 (1), 23–37.

Karels, V. C., Veelenturf, L. P., Van Woensel, T., 2020. An auction for collaborative vehicle routing: Models and algorithms. EURO Journal on Transportation and Logistics 9 (2), 100009.

Kellner, F., Schneiderbauer, M., 2019. Further insights into the allocation of greenhouse gas emissions to shipments in road freight transportation: The pollution routing game. European Journal of Operational Research 278 (1), 296–313.

Kimms, A., Kozeletskyi, I., 2016a. Core-based cost allocation in the cooperative traveling salesman problem. European Journal of Operational Research 248 (3), 910–916.

Kimms, A., Kozeletskyi, I., 2016b. Shapley value-based cost allocation in the cooperative traveling salesman problem under rolling horizon planning. EURO Journal on Transportation and Logistics 5 (4), 371–392.

Kinga, D., Adam, J. B., et al., 2015. A method for stochastic optimization. In: International conference on learning representations (ICLR). Vol. 5. San Diego, California;, p. 6.

Krajewska, M. A., Kopfer, H., Laporte, G., Ropke, S., Zaccour, G., 2008. Horizontal cooperation among freight carriers: Request allocation and profit sharing. Journal of the Operational Research Society 59 (11), 1483–1491.

Kuipers, J., Mosquera, M. A., Zarzuelo, J. M., 2013. Sharing costs in highways: A game theoretic approach. European Journal of Operational Research 228 (1), 158–168.

Lange, J., Crainic, T. G., Gschwind, T., Rei, W., 2025. An Integrated Modeling Approach for Day-Before Planning in Two-Tier Multi-Modal City Logistics with Satellite Synchronization. Publication Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Montréal, QC, Canada.

Larsen, E., Lachapelle, S., Bengio, Y., Frejinger, E., Lacoste-Julien, S., Lodi, A., 2022. Predicting tactical solutions to operational planning problems under imperfect information. INFORMS Journal on Computing 34 (1), 227–242.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Levinger, C., Hazon, N., Azaria, A., 2021. Efficient computation and estimation of the Shapley value for traveling salesman games. IEEE Access 9, 129119–129129.

Li, J., Rong, G., Feng, Y., 2015. Request selection and exchange approach for carrier collaboration based on auction of a single request. Transportation Research Part E: Logistics and Transportation Review 84, 23–39.

Li, X., Bai, R., Siebers, P.-O., Wagner, C., 2019. Travel time prediction in transport and logistics: Towards more efficient vehicle GPS data management using tree ensemble methods. VINE Journal of Information and Knowledge Management Systems 49 (3), 277–306.

Lone, S., Harboul, N., Weltevreden, J., 2021. European E-commerce Report.

Lundberg, S. M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Vol. 30. Curran Associates, Inc.

Machado, B., Pimentel, C., de Sousa, A., 2023. Integration planning of freight deliveries into passenger bus networks: Exact and heuristic algorithms. Transportation Research Part A: Policy and Practice 171, 103645.

Mancini, S., Gansterer, M., Hartl, R. F., 2021. The collaborative consistent vehicle routing problem with workload balance. European Journal of Operational Research 293 (3), 955–965.

Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., Tlahig, H., 2017. Optimization of a city logistics transportation system with mixed passengers and goods. EURO Journal on Transportation and Logistics 6 (1), 81–109.

Matl, P., Hartl, R. F., Vidal, T., 2019. Workload equity in vehicle routing: The impact of alternative workload resources. Computers & Operations Research 110, 116–129.

Metulini, R., Gnecco, G., 2023. Measuring players' importance in basketball using the generalized Shapley value. Annals of Operations Research 325 (1), 441–465.

Mitchell, R., Cooper, J., Frank, E., Holmes, G., 2022. Sampling permutations for Shapley value estimation. The Journal of Machine Learning Research 23 (1), 2082–2127.

Montoya-Torres, J. R., Muñoz-Villamizar, A., Vega-Mejía, C. A., 2016. On the impact of collaborative strategies for goods delivery in city logistics. Production Planning & Control 27 (6), 443–455.

Morabit, M., Desaulniers, G., Lodi, A., 2024. Learning to repeatedly solve routing problems. Networks 83 (3), 503–526.

Myerson, R. B., 1980. Conference structures and fair allocation rules. International Journal of Game Theory 9, 169–182.

Nataraj, S., Ferone, D., Quintero-Araujo, C., Juan, A., Festa, P., 2019. Consolidation centers in city logistics: A cooperative approach based on the location routing problem. International Journal of Industrial Engineering Computations 10 (3), 393–404.

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al., 2019. Kerastuner. `https://github.com/keras-team/keras-tuner`.

Özener, O. Ö., Ergun, Ö., Savelsbergh, M., 2013. Allocating cost of service to customers in inventory routing. Operations Research 61 (1), 112–125.

Padmanabhan, B., Huynh, N., Ferrell, W., Badyal, V., 2022. Potential benefits of carrier collaboration in vehicle routing problem with pickup and delivery. Transportation Letters 14 (3), 258–273.

Pan, S., Trentesaux, D., Ballot, E., Huang, G. Q., 2019. Horizontal collaborative transport: Survey of solutions and practical implementation issues. International Journal of Production Research 57 (15-16), 5340–5361.

Parisa, D. N., Samuel, K. E., Espinouse, M.-L., 2019. Systematic literature review on city logistics: Overview, classification and analysis. International Journal of Production Research 57 (3), 865–887.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Perboli, G., Brotcorne, L., Bruni, M. E., Rosano, M., 2021. A new model for last-mile delivery and satellite depots management: The impact of the on-demand economy. Transportation Research Part E: Logistics and Transportation Review 145, 102184.

Pérez-Bernabeu, E., Juan, A. A., Faulin, J., Barrios, B. B., 2015. Horizontal cooperation in road transportation: A case illustrating savings in distances and greenhouse gas emissions. International Transactions in Operational Research 22 (3), 585–606.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Computers & Operations Research 34 (8), 2403–2435.

Popescu, D. C., Kilby, P., 2020. Approximation of the shapley value for the euclidean traveling salesman game. Annals of Operations Research 289 (2), 341–362.

Raihan, A., Tuspekova, A., 2022. The nexus between economic growth, energy use, urbanization, tourism, and carbon dioxide emissions: New insights from singapore. Sustainability Analytics and Modeling 2, 100009.

Rao, C., Goh, M., Zhao, Y., Zheng, J., 2015. Location selection of city logistics centers under sustainability. Transportation Research Part D: Transport and Environment 36, 29–44.

Roshanaei, V., Luong, C., Aleman, D. M., Urbach, D. R., 2017. Collaborative operating room planning and scheduling. INFORMS Journal on Computing 29 (3), 558–580.

Rusich, A., et al., 2017. Collaborative logistics networks. Ph.D. thesis, Università degli Studi di Trieste.

Sánchez, A. L., Lujak, M., Semet, F., Billhardt, H., 2022. On balancing fairness and efficiency in routing of cooperative vehicle fleets. In: Twelfth International Workshop on Agents in Traffic and Transportation co-located with the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI 2022). Vol. 3173.

Savelsbergh, M., Van Woensel, T., 2016. 50th anniversary invited article—city logistics: Challenges and opportunities. Transportation Science 50 (2), 579–590.

Scherr, Y. O., Saavedra, B. A. N., Hewitt, M., Mattfeld, D. C., 2019. Service network design with mixed autonomous fleets. Transportation Research Part E: Logistics and Transportation Review 124, 40–55.

Schmeidler, D., 1969. The nucleolus of a characteristic function game. SIAM Journal on Applied Mathematics 17 (6), 1163–1170.

Schmidt, J., Tilk, C., Irnich, S., 2022. Using public transport in a 2-echelon last-mile delivery network. European Journal of Operational Research 317 (3), 827–840.

Schopka, K., Kopfer, H., 2017. Cost allocation for horizontal carrier coalitions based on approximated Shapley values. In: Operations Research Proceedings 2015: Selected Papers of the International Conference of the German, Austrian and Swiss Operations Research Societies (GOR, ÖGOR, SVOR/ASRO), University of Vienna, Austria, September 1-4, 2015. Springer, pp. 133–140.

Shapley, L. S., 1953. A value for n-person games. In: Kuhn, H. W., Tucker, A. W. (Eds.), Contributions to the Theory of Games II. Princeton University Press, Princeton, pp. 307–317.

Shapley, L. S., 1971. Cores of convex games. International Journal of Game Theory 1, 11–26.

Shapley, L. S., Mann, I., 1960. Values of large games, iv: Evaluating the electoral college by monte carlo techniques. Tech. rep., RAND Corporation.

Shelbourne, B. C., Battarra, M., Potts, C. N., 2017. The vehicle routing problem with release and due dates. INFORMS Journal on Computing 29 (4), 705–723.

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., Van Woensel, T., 2023. Two-echelon vehicle routing problems: A literature review. European Journal of Operational Research 304 (3), 865–886.

Sobhanan, A., Park, J., Park, J., Kwon, C., 2024. Genetic algorithms with neural cost predictor for solving hierarchical vehicle routing problems. Transportation Science 59 (2), 322–339.

Soriano, A., Gansterer, M., Hartl, R. F., 2023. The multi-depot vehicle routing problem with profit fairness. International Journal of Production Economics 255, 108669.

Stellingwerf, H. M., Laporte, G., Cruijssen, F. C., Kanellopoulos, A., Bloemhof, J. M., 2018. Quantifying the environmental and economic benefits of cooperation: A case study in temperature-controlled food logistics. Transportation Research Part D: Transport and Environment 65, 178–193.

Sun, L., Rangarajan, A., Karwan, M. H., Pinto, J. M., 2015. Transportation cost allocation on a fixed route. Computers & Industrial Engineering 83, 61–73.

Sun, Y., Ernst, A., Li, X., Weiner, J., 2021. Generalization of machine learning for problem reduction: A case study on travelling salesman problems. OR Spectrum 43 (3), 607–633.

Tijs, S. H., Driessen, T. S., 1986. Game theory and cost allocation problems. Management Science 32 (8), 1015–1028.

Timmer, J., Chessa, M., Boucherie, R. J., 2013. Cooperation and game-theoretic cost allocation in stochastic inventory models with continuous review. European Journal of Operational Research 231 (3), 567–576.

Touati, S., Radjef, M. S., Lakhdar, S., 2021. A Bayesian Monte Carlo method for computing the Shapley value: Application to weighted voting and bin packing games. Computers & Operations Research 125, 105094.

Umweltbundesamt Österreich, 2024. Emissionsfaktoren der verkehrsträger. Technical report, Umweltbundesamt Österreich, Vienna, Austria, accessed: 2025-02-14.
URL `https://www.umweltbundesamt.at/umweltthemen/mobilitaet/mobilitaetsdaten/emissionsfaktoren-verkehrsmittel`

United Nations, 2015. Sustainable Development Goals. Department of Economic and Social Affairs.
URL `https://sdgs.un.org/goals`

United Nations, 2018. World Urbanization Prospects. United Nations, Dept. of Economic and Social Affairs, Population Division.
URL `https://population.un.org/wup/publications/`

Van der Hagen, L., Agatz, N., Spliet, R., Visser, T. R., Kok, L., 2024. Machine learning–based feasibility checks for dynamic time slot management. Transportation Science 58 (1), 94–109.

Vanovermeire, C., Vercruysse, D., Sörensen, K., 2014. Analysis of different cost allocation methods in a collaborative transport setting. International Journal of Engineering Management and Economics 4 (2), 132–150.

Verdonck, L., Beullens, P., Caris, A., Ramaekers, K., Janssens, G. K., 2016. Analysis of collaborative savings and cost allocation techniques for the cooperative carrier facility location problem. Journal of the Operational Research Society 67 (6), 853–871.

Voigt, S., Frank, M., Fontaine, P., Kuhn, H., 2022. Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. Computers & Operations Research 146, 105856.

Weisberg, S., 2005. Applied linear regression. Vol. 528. John Wiley & Sons.

Winkenbach, M., Kleindorfer, P. R., Spinler, S., 2016. Enabling urban logistics services at la poste through multi-echelon location-routing. Transportation Science 50 (2), 520–540.

Young, H. P., 1985. Monotonic solutions of cooperative games. International Journal of Game Theory 14 (2), 65–72.

Zhang, Q., Huang, M., Wang, H., 2024. Collaborative service network design for multiple logistics carriers considering demand uncertainty. Symmetry 16 (8), 1083.

Zhang, T., Niu, C., Nair, D. J., Robson, E. N., Dixit, V., 2023. Transportation resilience optimization from an economic perspective at the pre-event stage. Sustainability Analytics and Modeling 3, 100027.

Zhang, X., Chen, L., Gendreau, M., Langevin, A., 2022. Learning-based branch-and-price algorithms for the vehicle routing problem with time windows and two-dimensional loading constraints. INFORMS Journal on Computing 34 (3), 1419–1436.

Zhu, S., Bell, M. G., Schulz, V., Stokoe, M., 2023. Co-modality in city logistics: Sounds good, but how? Transportation Research Part A: Policy and Practice 168, 103578.

Zolezzi, J. M., Rudnick, H., 2002. Transmission cost allocation by cooperative games and coalition formation. IEEE Transactions on power systems 17 (4), 1008–1015.

AI Support Statement

During the preparation of this thesis, I used ChatGPT for text revision, meaning grammar and wording. The results have been reviewed and I take full responsibility for the content.

Ingolstadt, January 2026

(Johannes Gückel)

Declaration of Honor

I hereby certify that I have prepared this written dissertation independently and without unauthorized outside assistance. I have not used any other writings and aids than those indicated in the thesis and have marked the passages taken verbatim or in content from the works used. In particular, I affirm that I have not made use of the help of mediation or consulting services.
Furthermore, I affirm that I have not made any previous attempts at a doctorate, completed any doctorates, or submitted the dissertation in the same or a different form in another attempt or in another examination procedure.

Ingolstadt, January 2026

(Johannes Gückel)