TUM School of Management
Technische Universität München

Uncertainty-aware optimization of public transportation systems

Benedikt Linus Lienkamp

Vollständiger Abdruck der von der TUM School of Management der Technischen Universität München zur Erlangung eines Doktors der Wirtschafts- und Sozialwissenschaften (Dr. rer. pol.) genehmigten Dissertation.

Vorsitz : Prof. Dr. Rainer Kolisch

Prüfende der Dissertation:

1. Prof. Dr. Maximilian Schiffer
2. Prof. Dr. Pirmin Fontaine

Die Dissertation wurde am 26.05.2025 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.07.2025 angenommen.

Contents

Acknowledgments

I would like to express my heartfelt gratitude to several individuals who have played a pivotal role in my academic journey. First and foremost, I extend my deepest thanks to Prof. Dr. Maximilian Schiffer for his unwavering support and guidance. His trust in my abilities and the freedom he provided in my work have been instrumental in my personal and professional growth. I am also grateful to Prof. Dr. Mike Hewitt for his extensive expertise and for facilitating the opportunity and funding for my research stay in Chicago, which greatly enriched my research experience. Prof. Dr. Axel Parmentier deserves special mention for his invaluable professional expertise. Furthermore, I would like to thank Prof. Dr. Pirmin Fontaine for being my second examiner and Prof. Dr. Rainer Kolisch for chairing the examination board.

I am thankful to all my teachers who have inspired my passion for mathematics, from learning about the basics of compound interest at Edith Stein Primary School to cryptography at Werner Heisenberg High School and finally higher mathematics at the Technical University of Munich. I especially want to thank Felix Schüller and Prof. Dr. Jürgen Richter-Gebert, whose enthusiasm for the subject has left a lasting impression on me.

My research team has made this journey feel like a breeze, and I am thankful for their camaraderie and support over the years. In this context, I would like to acknowledge Patrick for continually igniting a spark of motivation within me through our numerous research discussions. Additionally, I appreciate Paul for sharing his management perspective, which has been essential for incorporating managerial insights into my work.

On a personal note, my parents, siblings, in-laws, and friends have been a constant source of encouragement throughout this journey. I am particularly thankful to my father, whose years of mentoring have been invaluable for my development, and my mother, who continually reminds me not to overlook the emotional aspects of my work.

Lastly, I want to express my deepest appreciation to my wife, Viola, who has been my rock for many years, providing me with daily motivation to continue growing. This journey would not have been possible without you!

Abstract

Public transportation systems form the backbone of urban mobility, providing a reliable and efficient means of travel for millions of people per day. Designing and operating today's public transportation systems to allow for sustainable transport requires advanced planning approaches from strategic, tactical, and operational perspectives. This thesis aims to provide tools and algorithms to address complex planning problems arising from these different perspectives. Specifically, we analyze traditional fixed-line public transportation systems from a strategic perspective, as well as tactical and operational planning problems with stochastic passenger demand that arise in so-called demand adaptive systems. These demand adaptive systems offer a transportation service that combines the regularity and predictability of traditional public transportation with the flexibility of on-demand systems.

This thesis includes an introduction, a literature review, three methodological chapters, and a conclusion. The introduction outlines the different planning problems arising in designing public transportation system with a focus on the integration of demand adaptive systems. Following this, the literature review connects these planning problems to relevant streams of literature, offering a comprehensive overview of the current state-of-the-art, highlighting potential research gaps, and positioning the contributions of this thesis within a broader methodological context. The three methodological chapters are the main contributions of this thesis.

The first methodological chapter addresses determining optimal passenger flows in large scale intermodal transportation systems. The chapter designs a partially time-expanded network that allows us to formulate the underlying optimization problem as a minimum cost multi commodity network flow problem. To solve this problem, we develop a column generation approach to find continuous solutions, which we combine with a Price-and-Branch procedure to obtain integer solutions. Specifically, we solve instances with up to 56,295 passengers to optimality and show that the computation time of our algorithm can be reduced by up to 60% through the use of our pricing filter and by up to an additional 90% through the use of our A^*-based pricing algorithm.

The second methodological chapter deals with determining the general visit order of transit stops in demand adaptive systems under uncertain passenger demand. We formulate this problem as a stochastic program and propose a Branch-and-Price algorithm for solving its deterministic equivalent. Additionally, this chapter develops a local search approach with which we improve the performance of the Branch-and-Price approach. We assess the efficiency of the proposed methods on a set of instances from the literature and demonstrate that the proposed methods outperform a known benchmark, improving upper bounds by up to 85% and lower bounds by up to 55%. Furthermore, the solutions of the stochastic model are both more cost-effective and robust than those of the deterministic model.

The third methodological chapter proposes an algorithmic framework to solve the operational planning problem under uncertainty for demand adaptive systems in which passenger requests must be accepted or rejected as soon as they are issued. This chapter

models the proposed problem as a Markov decision process and utilizes a rolling horizon approach to approximate the Markov decision process via a two-stage stochastic program. Furthermore, we decompose the two-stage stochastic program into several full information planning problems in each timestep to decide on the next action and propose a consensus-based heuristic and a myopic approach. We assess the efficiency of the proposed methods on real-world data and show that our exact decomposition yields the best results in under five seconds and that our heuristic approach can reduce the serial computation time by half compared to our exact decomposition, with a solution quality decline of less than one percent. From a managerial perspective, we show that by switching a fixed-line bus route to a demand adaptive system, an operator can increase profit by up to 49% and the number of served passengers by up to 35% while only increasing the travel distance of the bus by 14%. Furthermore, we show that an operator can reduce their cost per passenger by 43 - 51% by increasing route flexibility and that incentivizing passengers to walk slightly longer distances reduces the cost per passenger by 83-85%.

The thesis concludes with a detailed summary and critical discussion of the managerial and methodological contributions. It further outlines opportunities for future research made available through the contributions of this thesis, highlighting how to leverage the developed methodology for this purpose. Note that parts of the introduction, literature review, and conclusion originate from the three methodological chapters.

1 Introduction

1.1 Background

According to the United Nations, the global urban population will increase by 900 million from 2020 to 2030, reaching 5.2 billion (UN, 2018). This population increase, in combination with already high congestion and traffic emissions in cities all around the world, calls for better utilization of existing infrastructure and the development of novel transportation concepts to keep up with future urban travel demand. In this context, public transportation systems (PTS) play a crucial role in modern cities as they provide an efficient and cost-effective way to meet people's travel demands. They naturally allow for a high utilization, which leads to less congestion, traffic emissions, and transportation costs. In most modern cities, a PTS consists of multiple transportation modes such as subway, tram, and bus lines. By offering a diverse range of transit options, PTS can cater to the unique needs of different passengers for both daily work commutes and leisure trips. Additionally, the different modes of transportation can complement one another, creating a comprehensive transit network that offers multiple routes and options to reach various destinations.

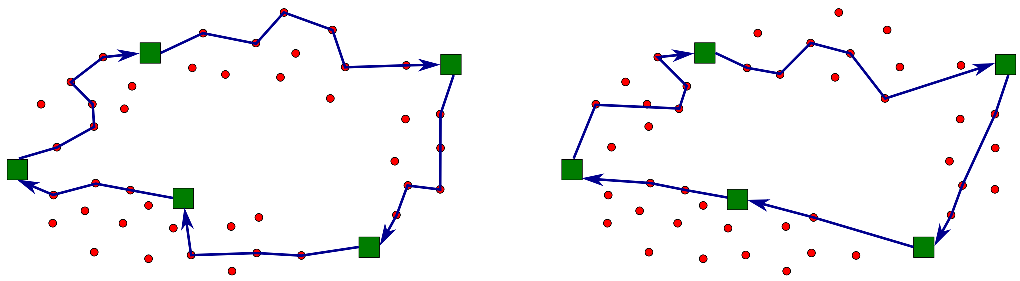
Designing and operating today's PTS to allow for sustainable transport requires advanced planning approaches from strategic, tactical, and operational perspectives. The decision support tools for these planning tasks span simulation-based and optimization-based approaches. To this end, simulation-based approaches are used to model daily operations, particularly when aiming to understand passenger mode choice behavior (see, e.g., Hintermann et al., 2023; Huang et al., 2023). Optimization-based approaches often simplify user behavior and are used to make network design decisions from a strategic perspective (cf. Amberg et al., 2019; Farahani et al., 2013). Between these two purely operational or strategic perspectives, optimization-based approaches that assume a central planner to decide on system optimal passenger flows allow for tactical planning and system analyses (see, e.g. Salazar et al., 2019; Schiffer et al., 2021). Specifically, such approaches are useful for the following purposes: first, planners can use system optimal passenger flows to obtain an upper bound on a PTS's throughput, which allows quantifying the maximum improvement potential within the PTS compared to status quo operations. Second, such system optimal passenger flow information can be used to inform or benchmark operational control algorithms, e.g., pricing mechanisms or online algorithms, that aim to improve the system's performance by influencing user behavior. Third, obtaining information on a system's maximum throughput is helpful for short-term network design decisions, e.g., which line or mode of transport to activate or deactivate during system disruptions such as driver or vehicle shortage, construction work, and medical emergencies.

Determining system optimal passenger flows requires solving Multi-Commodity Network Flow (MCNF) problems that significantly exceed problem sizes that have so far been solved to optimality. Additionally, the intermodal structure of PTS needs to be considered. So far, a tool that allows a central planner to determine system optimal passenger flows in large-scale intermodal PTS does not exist.

Focusing on bus lines in PTS, we notice that **line-based bus lines** serve as a cost-effective and reliable backbone of PTS networks, offering widespread coverage that connects diverse communities. Their high passenger capacity and established routes promote social inclusivity and accessibility to essential services. Moreover, they enable efficient utilization of existing infrastructure, allowing the transport of a larger number of people using fewer vehicles, which can contribute to reduced traffic congestion and emissions. However, they can be susceptible to traffic congestion and limited flexibility in adapting to changing travel demands, leading to potential delays and less personalized service experiences for passengers.

Demand Responsive Systems (DRS) offer more flexibility and can adapt to fluctuating individual transportation demands, thus enabling more personalized services while maintaining a degree of resource sharing. DRS were first introduced under the name of Dial-a-Ride (DAR) as a door-to-door service for users with reduced mobility (Wilson et al., 1971; Ioachim et al., 1995; Toth and Vigo, 1996). One drawback of DAR are the higher operational costs associated with providing individualized service and accommodating specific travel requests. Additionally, the real-time nature of pickups and drop-offs, i.e., significant route changes during operation, can lead to unpredictable travel times for passengers, which might not be suitable for those with time-sensitive commitments.

Demand Adaptive Systems (DAS) (Malucelli et al., 1999; Quadrifoglio et al., 2007) incorporate DRS into scheduled bus transportation: a DAS bus line provides a line-based transit-line service for a set of compulsory stops, which are bound to a schedule with fixed time windows Passengers can issue requests at non-compulsory stops that are optional from an operator perspective and can be served via detours in between compulsory stops (see Figure 1.1). Accordingly, a DAS combines the regularity and predictability of line-based PTS, which enables seamless integration with other modes of transportation and services that do not require advanced booking, with the flexibility of a DRS. Ultimately,



(a) Route variant 1 (b) Route variant 2

Figure 1.1: A DAS line with compulsory stops (green) and optional stops (red).

a DAS combines the advantages of line-based PTS and DRS which can increase efficiency, utilization, and service level of the respective PT network.

Planning the operations of DAS is a highly complex task that requires balancing the characteristics of line-based and on-demand systems and involves both a service design phase, where routes and schedules are established and an operational phase, with adjustments to vehicle routes and schedules in response to user requests. The complexity in planning DAS operations has been extensively studied in the literature, with a comprehensive review of methodological aspects presented in Errico et al. (2013).

Focusing on the aforementioned service design phase, the tactical planning of a DAS line encompasses three main interrelated tasks: (1) identifying and selecting compulsory stops; (2) determining a general visit order of transit stops within the service area; and (3) establishing schedules for compulsory stops. Errico (2008) and later Errico et al. (2011a) introduce the Single-line DAS Design Problem (SDDP), which assumes the set of stops to be served by the DAS line, the transportation demand, and the travel time between all stop pairs to be known. Based on this information, the SDDP determines the compulsory stops, their sequence, the DAS segments, and the master schedule of the compulsory stops. The authors propose two decompositions of the SDDP consisting of the selection of the compulsory stops, the General Minimum Latency Problem (GMLP), which is a sequencing problem of the compulsory and optional stops, and the master schedule problem. Errico et al. (2011a) address the selection of the compulsory stops and compare the two decompositions. Errico et al. (2017) propose the traveling salesman problem with generalized latency (TSP-GL), a variant of the GMLP where the objective function combines the vehicle routing cost and the passenger's travel times, and develop a Benders' decomposition approach to solve it efficiently. This TSP-GL efficiently determines a general visit order of transit stops within the service area. Crainic et al. (2012) proposes a mathematical description and a solution method based on probabilistic approximations to solve the master schedule problem. Focusing especially on the general order of transit stops within the service area, we see that so far, only average passenger demands are considered, and the distribution of uncertain passenger demands is ignored.

Looking at the aforementioned operational phase, efficiently determining whether to accept or reject passenger requests for a DAS service to maximize the operators' profit is pivotal for operating a DAS line economically. Crainic et al. (2005) analyze this problem in an offline setting and develop and compare different meta-heuristics to solve it. However, these solution approaches assume full information about all future passenger requests, which is not the case in practice, where an operator wants to inform passengers about their requests immediately after they are issued.

Against this background, we identify three levers to improve the state of the art for the optimization of intermodal transportation systems and the integration of stochasticity into the planning of DAS.

i) *Providing an in-depth potential analysis of intermodal public transportation systems* by optimizing passenger flows at scale from an operator's perspective.

ii) *Considering the stochastic distribution of passenger demands* when determining the general visit order of transit stops in the tactical planning of DAS.

iii) *Planning operations of a DAS line under partial information* by deciding on accepting or rejecting passenger requests in a dynamic setting, considering stochastic future passenger demand.

1.2 Aims and Scope

Against this background, this thesis aims to address challenges of high-practical relevance related to intermodal public transportation systems, with a focus on DAS at the tactical and operational levels. Specifically, the scope of this thesis is twofold. First, the thesis provides optimization-based frameworks and contributes novel models and applicable solution methods that allow to

i) *Optimize intermodal passenger flows:* This setting aims to optimize passenger flows in an intermodal public transportation system by minimizing the sum of travel times for all passengers from a system perspective.

ii) *Determine the general visit order of transit stops in DAS under uncertainty:* In this tactical problem setting, an operator selects a suitable subset of stops to serve in a route, i.e., a subset of stops that fulfills a percentage of passenger demand with a given probability. Additionally, the Hamiltonian tour on the subset of nodes is determined, minimizing a linear combination of route design and passenger routing costs.

iii) *Operate a DAS under uncertainty:* This operational problem setting assumes a DAS route with compulsory stops, optional stops, route segments, and the master schedule being determined. Based on this route information and historical passenger demand, an operator receives passenger requests, which have to be accepted or rejected immediately.

Second, extensive numerical experiments conducted on real-world data generate managerial and technical insights that assess and authenticate the advantages offered by the proposed solutions.

1.3 Contribution

The remainder of this thesis is structured as follows.

Chapter 2 reviews the literature related to the three optimization problems studied in this thesis. Accordingly, existing work on MCNF, Fixed-Charge Network Design Problems (FNDP), semi-flexible transit systems, and integrating uncertain passenger demand into optimization models for PTS are reviewed.

Chapter 3 proposes an algorithmic framework to determine the system optimum of a large-scale intermodal transportation system. To solve this problem, this chapter's contribution is four-fold: First, the intermodal transportation system is modeled by combining two core principles of network optimization, layered-graph structures and (partially) time-expanded networks, which allows us to implicitly encode problem-specific constraints related to intermodality. This enables us to solve a standard integer minimum-cost MCNF problem to determine the system optimum for an intermodal transportation system. Second, to solve the integer minimum-cost MCNF problem efficiently, this chapter presents a column generation (CG) approach to find continuous minimum-cost MCNF solutions, which we combine with a Price-and-Branch (P&B) procedure to obtain integer solutions. To speed up our CG approach, this chapter develops a pricing filter and an admissible distance approximation to utilize the A^* algorithm for solving the pricing problems. Third, this chapter shows the efficiency of the proposed algorithmic framework by applying it to a real-world case study for the city of Munich, where we solve instances with up to 56,295 passengers to optimality, and show that the computation time of our algorithm can be reduced by up to 60% through the use of our pricing filter and by up to additional 90% through the use of the A^*-based pricing algorithm. Fourth, this chapter opens this algorithmic framework as open-source code for further research.

Chapter 4 proposes an algorithmic framework to determine the general visit order of transit stops in DAS under uncertainty. To do so, it introduces an extension of the symmetric traveling salesman problem with generalized latency (TSP-GL) that explicitly models uncertainty. The Stochastic TSP-GL (STSP-GL) aims to choose a subset of nodes of an undirected graph and determines a Hamiltonian tour amongst those nodes, minimizing an objective function that is a weighted combination of route design and passenger routing costs. These nodes are selected to ensure that a predefined percentage of uncertain passenger demand is served with a given probability. This chapter formulates the STSP-GL as a stochastic program and proposes a Branch-and-Price (B&P) algorithm for solving its deterministic equivalent. Additionally, this chapter develops a local search approach with which we improve the performance of the B&P approach. This chapter assesses the efficiency of the proposed methods on a set of instances from the literature and demonstrates that the proposed methods outperform a known benchmark, improving upper bounds by up to 85% and lower bounds by up to 55%. Finally, this chapter shows that solutions of the stochastic model are both more cost-effective and more robust than those of the deterministic model.

Chapter 5 proposes an algorithmic framework to solve the operational route planning problem under uncertainty for a DAS in which passenger requests have to be accepted or rejected as soon as they are issued. To solve this problem, this chapter models the proposed problem as a Markov decision process and utilizes a rolling horizon approach to approximate the Markov decision process via a two-stage stochastic program. Furthermore, this chapter decomposes the two-stage stochastic program into several full information planning problems in each timestep to decide on the next action and proposes a consensus-based heuristic and a myopic approach. This chapter then assesses the efficiency of the proposed methods on real-world data and shows that our exact decomposition yields the best results in under five seconds and that our heuristic approach can reduce the serial computation time by half compared to our exact decomposition, with a solution quality decline of less than one percent. From a managerial perspective, this chapter shows that by switching a fixed-line bus route to a demand adaptive system, an operator can increase profit by up to 49% and the number of served passengers by up to 35% while only increasing the travel distance of the bus by 14%. Furthermore, this chapter shows that an operator can reduce their cost per passenger by 43 - 51% by increasing route flexibility and that incentivizing passengers to walk slightly longer distances reduces the cost per passenger by 83-85%.

Chapter 6 concludes the thesis through a comprehensive summary of its contributions and results, discusses limitations, and highlights future research perspectives.

2 State of the art

2.1 Introduction

Public transportation systems (PTS) play a crucial role in modern cities as they provide an efficient and cost-effective way to meet people's travel demands. They naturally allow for a high utilization, which leads to less congestion, traffic emissions, and reduced transportation costs. Designing and operating today's PTS to allow for sustainable transport requires advanced planning approaches from strategic, tactical, and operational perspectives.

This thesis introduces innovative tools and algorithmic approaches to help PTS operators tackle several planning problems. Specifically, it i) develops a methodology to determine optimal passenger flows in intermodal PTS, ii) proposes a mathematical decomposition to determine the general visit order of transit stops under uncertainty in the tactical planning of Demand Adaptive Systems (DAS), and iii) designs an algorithmic approach to decide on whether to accept or reject passenger requests under partial information in the operational planning of DAS.

Accordingly, the individual contributions of this work touch various research streams, ranging from Multi-Commodity Network Flow (MCNF) problems, over Fixed-Charge Network Design Problems (FNDP), to semi-flexible transit systems, and integrating uncertain passenger demand into optimization models for PTS. This chapter aims to provide a concise review of these research streams. To this end, it is divided into three distinct sections, corresponding to the three main contributions of this thesis. The first section reviews literature related to MCNF problems which are the basis for optimizing passenger flows in intermodal PTS. The second section reviews the literature on FNDPs and the integration of uncertain passenger demand into optimization models for PTS. The third section reviews the literature on semi-flexible transit systems, which include DASs.

2.2 Multi-Commodity Network Flow Problems

MCNF models have been used to model a variety of problems, amongst others in communication (Ouorou et al., 2000; Lin, 2001; Wagner et al., 2007), logistics (Erera et al., 2005; Al-Khayyal and Hwang, 2007; Psaraftis, 2011), and transportation systems (Ayar and Yaman, 2012; Paraskevopoulos et al., 2016; Zhang et al., 2019). Besides these research fields, MCNF models also find application in financial problem settings (Yang and Kim, 2015), evacuation planning (Pyakurel and Dhamala, 2017), production services (Atamtürk and Zhang, 2007), and traffic control (Bertsimas and Patterson, 2000), which shows the great versatility of MCNF models. Various heuristic, approximation, and exact solution methods have been used to solve different variants of MCNF problems.

The most prominent heuristic methods to solve MCNF problems are genetic algorithms (Khouja et al., 1998; Alavidoost et al., 2018), simulated annealing (Yaghini et al., 2012; Moshref-Javadi and Lee, 2016), and tabu search (Ghamlouche et al., 2003; Crainic et al., 2006). Various approximation methods exist for different types of network flow formulations. Lagrangian-based methods proved to be very efficient at solving MCNF problems and find their application in solving multi-commodity capacitated fixed charge network design problems (Crainic et al., 2001) and minimum-cost MCNF problems (Retvdri et al., 2004).

The two most prominent exact approaches to solve MCNF problems are branching and decomposition methods. Branching methods build on the idea of branch-and-bound (B&B) (Lawler and Wood, 1966) and have been used to solve integer MCNF problems (see, e.g., Brunetta et al., 2000). Decomposition methods were developed by Dantzig and Wolfe (1960) and used, for example, by Barnhart et al. (1994) to solve message routing problems, formulated as minimum-cost MCNF problems and solved via column generation (CG). Barnhart et al. (2000) utilized the branch-and-price (B&P) algorithm (Barnhart et al., 1998), a combination of CG and B&B, to solve integer MCNF problems. Here, the authors applied bounds provided by solving linear programs, using column-and-cut generation at nodes of the branch-and-bound tree in combination with cuts, generated at each node in the B&B tree, to mitigate symmetry effects.

In general, MCNF models have recently been used to model transportation systems, e.g., Autonomous Mobility on Demand (AMoD) systems in congested road networks (Rossi et al., 2018) and to study the interaction of AMoD with the public transportation system (Salazar et al., 2019) on a mesoscopic level. We refer to Salimifard and Bigharaz (2022) for an extensive overview of applications and solution methods for MCNF problems.

Dynamic flows add a time dimension to static network flow problems and allow flow values on arcs to change over time. In dynamic network flow models for traffic optimization, the size of the underlying graph can be a limiting factor for the model's granularity or time horizon. To address the problem of exponential growth in dynamic networks, Boland et al. (2017) introduced the concept of partially time-expanded networks to service network design problems. In partially time-expanded networks, not all timesteps are included in the network graph, which allows for iterative refinement to obtain a reduced graph size. Besides a reduced size of the underlying graph, multiple methods exist to speed up computation time in large network flow models. For an extensive overview of dynamic network flow problems, we refer to Skutella (2009).

2.3 Fixed-Charge Network Design Problems and uncertain passenger demand

Determining the general visit order of transit stops in DAS under uncertainty is closely related to the FNDP, which involves finding a set of arcs in a graph that enables the flow of commodities from their origins to their destinations in order to satisfy some demand characteristics. As we also consider uncertain passenger demand in the general visit order of transit stops, this section is divided into FNDP and integrating uncertain passenger demand into optimization models for PTS.

2.3.1 Fixed-Charge Network Design Problems

The FNDP minimizes routing costs, i.e., variable expenses incurred when sending flows through the network, or design costs, i.e., fixed expenses associated with establishing network infrastructure, or both (Costa, 2005).

Several exact and heuristic methods have been proposed to solve the FNDP. The most prominent exact methods to solve FNDPs are Lagrangian-based techniques, Benders decomposition, and polyhedral approaches. For an extensive overview of exact methods to solve FNDPs, we refer to Gendron et al. (1999), Gendron (2011), and Crainic et al. (2021).

Lagrangian-based techniques utilize either subgradient algorithms or bundle methods to solve the Lagrangian dual. Crainic et al. (2001) show that for Lagrangian-based techniques, bundle methods appear superior to subgradient approaches as they converge faster and are more robust relative to different relaxations and problem characteristics. Frangioni and Gorgone (2014) exploit the structure of the Dantzig-Wolfe reformulation of the Lagrangian dual to construct an efficient bundle method. Frangioni et al. (2017) analyzed a large class of subgradient methods to show that the theory developed in d'Antonio and Frangioni (2009) impacts the performance of subgradient methods. Gendron (2019) compares Lagrangian relaxation bounds for a large class of network design problems, which they obtain either by relaxing the linking constraints or the flow conservation constraints.

Costa et al. (2009) give an extensive overview on the application of Benders decomposition to FNDP and clarify the relationships between Benders, metric, and cutset inequalities, which are often used in solution algorithms for multi-commodity capacitated FNDP. Costa et al. (2012) propose a general scheme for generating extra cuts in a Benders decomposition, which they apply to the FNDP.

There exists a multitude of polyhedral approaches to solve the FNDP. Wolsey (2003) gives an overview on strong valid inequalities and tight extended formulations for general mixed integer linear programs. Chouman et al. (2017) develop a cutting-plane algorithm for the multi-commodity capacitated FNDP. Atamtürk (2002) analyzes the cut-set polyhedra of capacitated FNDP and shows that utilizing cut-set inequalities significantly improves computations. Atamtürk (2001) applies flow pack inequalities, which improve computations for capacitated fixed-charge network flow problems. Padberg et al. (1985) study the single-node fixed-charge flow problem and introduce flow cover inequalities.

The most prominent heuristic approaches to solve FNDP include classical heuristics such as local search (Walker, 1976), neighborhood-based metaheuristics such as tabu search (Gendreau and Potvin, 2019), simulated annealing (Delahaye et al., 2019), and iterated local search (Lourenço et al., 2019), population-based metaheuristics such as genetic algorithms (Whitley, 2019), and parallel metaheuristics (Crainic, 2019). Furthermore, several matheuristics have been applied to solve FNDPs.

There exist several approaches to solve FNDP via tabu search. Sun et al. (1998) apply tabu search with two strategies for each of the intermediate and long-term memory processes to the FNDP. Crainic et al. (2000) utilize a tabu search framework that combines pivot moves with column generation to solve multicommodity capacitated FNDP. Ghamlouche et al. (2003) propose cycle-based neighborhood structures for multicommodity-capacitated FNDP and show their efficiency through a simple tabu search method.

For neighborhood-based metaheuristics, Lotfi and Tavakkoli-Moghaddam (2013) introduce a priority-based encoding genetic algorithm for linear and nonlinear FNDP and show that it outperforms a spanning tree-based genetic algorithm in terms of solution quality and computation time. Ghamlouche et al. (2004) combine path relinking with cycle-based tabu search in an algorithm that shows stable solution quality and computation times. Crainic and Gendreau (2007) propose a scatter search approach for the capacitated FNDP, which cannot match the path relinking algorithm consistently but can produce better solutions on some of the test instances.

Several matheuristics have been applied to solve the FNDP. Rodríguez-Martín and Salazar-González (2010) utilize local branching (Fischetti and Lodi, 2003) to solve the multicommodity capacitated FNDP. They show that their algorithm outperforms existing cycle-based tabu search (Ghamlouche et al., 2003) and path relinking (Ghamlouche et al., 2004) approaches.

Hewitt et al. (2010) combine mathematical programming algorithms and heuristic search techniques to find provably high-quality solutions for capacitated FNDP quickly. They show that their algorithm outperforms cycle-based tabu search (Ghamlouche et al., 2003) and path relinking (Ghamlouche et al., 2004) as well.

Katayama and Yurimoto (2011) combine capacity scaling using a column-row generation technique and local branching. They show that their approach outperforms cycle-based tabu search (Ghamlouche et al., 2003), path relinking (Ghamlouche et al., 2004), local branching (Rodríguez-Martín and Salazar-González, 2010), and IP Search (Hewitt et al., 2010).

Yaghini et al. (2015) propose a cutting-plane neighborhood structure, which they implement in a tabu search framework and combine with local branching. They show that their approach outperforms cycle-based tabu search (Ghamlouche et al., 2003), path relinking (Ghamlouche et al., 2004), local branching (Rodríguez-Martín and Salazar-González, 2010), IP Search (Hewitt et al., 2010) and capacity scaling (Katayama and Yurimoto, 2011). We refer to Crainic et al. (2021) for an extensive overview of all exact and heuristic approaches to solve the FNDP.

2.3.2 Integrating uncertain passenger demand into optimization models for PTS

Several tactical planning problems for PTS have been studied under the assumption that all information about passenger demand is available before the operator's decisions are made. In most cases, this does not translate to real-world applications (Kall and Wallace, 1994). Traditionally, stochastic passenger demand is often not considered explicitly during the tactical planning phase but dealt with in the operational planning phase (Ibarra-Rojas et al., 2015). Accordingly, passenger demand is usually determined through forecasting methods, e.g., average historical data. Usually, the expected quality of a solution to a stochastic model is better than a solution to the deterministic counterpart when evaluated in the stochastic setting. Wallace (2000) and Higle and Wallace (2003) explain that this occurs, as the deterministic model is optimal for one specific scenario but might be very bad for scenarios where it is not optimal. Additionally, the deterministic solution is often feasible but not optimal in the stochastic model. Birge (1982) measures this difference in solution quality by the value of stochastic solution (VSS). Here, the VSS represents the potential benefit of solving the stochastic model instead of the deterministic model in the stochastic environment. Wallace (2000) shows that stochastic models may find solutions significantly different from deterministic models. Nevertheless, Thapalia et al. (2011) and Maggioni and Wallace (2012) show that for multiple other problems, certain structural patterns from the deterministic solutions may re-emerge in the stochastic solutions. We refer to Haneveld and Van der Vlerk (2020) for an extensive overview on the research area of stochastic programming.

In recent years, researchers have made significant strides in integrating uncertain passenger demand into optimization models for public transportation systems. Tian et al. (2021) consider the integration of autonomous vehicles into bus transit systems. In this context, they determine the optimal bus fleet size and its assignment onto multiple bus lines in a bus service network considering uncertain demand. Ma et al. (2021) study the problem of bus regulation with uncertain passenger demand and disturbance for urban rapid transit lines on the basis of a robust model predictive control algorithm. Gkiotsalitis and Alesiani (2019) incorporate travel time and passenger demand uncertainty to determine robust schedules that minimize the possible loss in worst-case scenarios. Li et al. (2019) propose a robust dynamic control framework that considers delay disturbances and passenger demand uncertainty to reduce bus bunching, a phenomenon that arises in a scheduled service where the delay of a leading bus leads to the following bus catching up.

2.4 Semi-flexible transit systems

Semi-flexible transit systems, which include DAS, have been studied from several perspectives under multiple different assumptions. Errico et al. (2013) categorize the planning decisions necessary to operate a semi-flexible transit system into strategic, tactical, and operational.

Strategic planning focuses on the definition of the service area, the itineraries in terms of potential compulsory and optional stops, the level of service, i.e., frequencies and fleet size, and whether to use fixed bus routes, DAS, or another type of public transportation system. In this context, Daganzo (1984) compares the cost between fixed route transit and two DAR variants to determine which type of transportation system is better suited to serve a given service area. Quadrifoglio and Li (2009) analyze the critical demand density below which their demand-responsive policy outperforms a fixed-route policy. Furthermore, Li and Quadrifoglio (2009) develop an analytical model that determines the optimal number of independently operated zones the service area should be divided into, to balance customer service quality and vehicle operating costs. Pratelli and Schoen (2001) propose a mathematical formulation to define the service area of a semi-flexible line based on the total aggregated travel time of passengers spent on board during deviations, the walking times, and the increase of waiting times at compulsory stops. Qiu et al. (2015) study a so-called demi-flexible operating policy of a feeder system in Zhengzhou City, China. This feeder system can deviate from its fixed route to offer door-to-door service. The demi-flexible operating policy allows rejected passengers to utilize curb-to-curb stops of accepted passengers to carry out their trips. The authors show that under certain assumptions, this demi-flexible policy offers advantages in terms of affordability and efficiency compared to fixed-route and fully flexible route policies. Viergutz and Schmidt (2019) perform a simulation-based study in a rural area in Germany and show that stop-based and door-to-door DRS are not economical in their rural service area because of their high operational costs. Bischoff et al. (2019) perform a simulation-based study in the city of Cottbus (Germany) to determine the fleet size of a stop-based and a door-to-door DRS and show that for autonomous vehicles, the two DRS services are significantly cheaper during peak hours compared to fixed-line policies.

Tactical planning analyses which compulsory and optional stops to include in the design of a semi-flexible line, in which sequence the compulsory stops should be visited, the distinct tour segments, i.e., between which two compulsory stops can an optional stop be visited, and the design of a so-called master schedule, which determines the time-windows of the compulsory stops. Errico et al. (2013) categorize tactical planning according to their solution method into continuous approximations and combinatorial optimization. As our methodology belongs to the latter solution method, we only review combinatorial optimization approaches for DAS in this paragraph. Errico (2008) and later Errico et al. (2011a) introduce the Single-line DAS Design Problem (SDDP), which assumes the set of stops to be served by the DAS line, the transportation demand, and the travel time between all stop pairs to be known. Based on this information, the SDDP determines the compulsory stops, their sequence, the DAS segments, and the master schedule of the compulsory stops. The authors propose two decompositions of the SDDP consisting of the selection of the compulsory stops, the General Minimum Latency Problem (GMLP), which is a sequencing problem of the compulsory and optional stops, and the master schedule problem. Errico et al. (2011a) address the selection of the compulsory stops and compare the two decompositions. Errico et al. (2017) propose the traveling salesman problem with generalized latency (TSP-GL), a variant of the GMLP where the objective function combines the vehicle routing cost and the passenger's travel times, and develop a Benders' decomposition approach to solve it efficiently. Crainic et al. (2012) proposes a mathematical description and a solution method based on probabilistic approximations to solve the respective master schedule problem.

Lastly, operational planning determines vehicle and driver schedules as well as real-time adjustments to the DAS route. There are two main operational problem categories: dynamic online and dynamic offline problems. Vansteenwegen et al. (2022) give an extensive overview of this categorization. Note that we do not explicitly consider static problems as they can often be transformed into dynamic offline strategies. Dynamic online strategies assume that requests can occur during the planning horizon and after the bus starts its trip. Here, Pei et al. (2019a) utilize a tabu search approach to reduce the number of visited stops in a fixed-bus line, maximizing the total system income, which is the income minus operating costs minus passenger time costs. They show that during off-peak hours, their approach outperforms fixed-line operations. Quadrifoglio et al. (2007) propose an insertion heuristic for Mobility Allowance Shuttle Transit (MAST), a special case of semi-flexible transit systems, which is described in detail in Errico et al. (2013). They show that their approach can be used to automate a MAST in Los Angeles (USA) and that their heuristic results are close to optimal solutions.

Dynamic offline strategies assume that requests can only be issued before the bus starts its trip. In this context, Malucelli et al. (1999) introduce DAS with three operational policies differing in user requirements. They propose linear integer formulations for these policies, which they solve through Lagrangean approaches and column generation. Malucelli et al. (2001) and Crainic et al. (2005) introduce a GRASP-based memory-enhanced constructive heuristic and a tabu search meta-heuristic to solve the operational problem of DAS under full information. Additionally, they combine the two heuristic approaches, resulting in multiple hybrid approaches, and show that for simple instances, a tabu search-based approach performs best. In contrast, for more complex instances, a hybrid approach yields the best results. Qiu et al. (2014) study a MAST system in Los Angeles (USA) and show that their dynamic stop strategy can reduce user cost by up to 30%.

We refer to Errico et al. (2013) and Vansteenwegen et al. (2022) for an in-depth overview of semi-flexible and demand-responsive systems.

2.5 Conclusion

This thesis contributes to the research fields outlined above by closing the following research gaps.

First, various optimization approaches exist to solve MCNF problems for different problem settings. So far, using an MCNF model to optimize intermodal passenger transport has only been done on a mesoscopic level (Salazar et al., 2019). Analysis of passenger transport in large transportation networks on a microscopic level is often done via simulation (Ziemke et al., 2019), which is not amenable for optimization. To the best of our knowledge, there exists no optimization framework that is capable of determining the passenger flow system optimum of a large-scale intermodal transportation system on a microscopic level. This thesis closes this gap by developing such an algorithmic framework.

Second, so far, uncertainty in passenger demand that allows a DAS operator to determine a general order of transit stops that is robust against volatile passenger demand has not been considered explicitly in the tactical planning phase of DAS. Accordingly, DAS operators are unable to analyze the impact of different service levels and the frequency of meeting them on the route design and served passenger demand. This thesis closes this gap by contributing the stochastic traveling salesman problem with generalized latency that accounts for serving a predefined percentage of stochastic passenger demand with a given probability.

Third, the planning process of the operations of semi-flexible transit systems and DASs has been studied from tactical and operational perspectives under several system assumptions utilizing simulations, continuous approximations, and combinatorial optimization. So far, operational DAS problems have only been considered under full demand information in Crainic et al. (2005), which does not allow an operator to immediately give feedback to a passenger about the acceptance or rejection decision. To close this gap this thesis proposes an algorithmic framework that considers the dynamic nature of passenger demand and allows an operator to decide whether to accept or reject passenger requests in real time.

3 Column generation for solving large scale multi-commodity flow problems for passenger transportation

In light of the need for design and analysis of intermodal transportation systems, we propose an algorithmic framework to determine the system optimum of an intermodal transportation system. To this end, we model an intermodal transportation system by combining two core principles of network optimization – layered-graph structures and (partially) time-expanded networks – to formulate our problem on a graph that allows us to implicitly encode problem specific constraints related to intermodality. This enables us to solve a standard integer minimum-cost multi-commodity flow problem to obtain the system optimum for an intermodal transportation system. To solve this integer minimum-cost multi-commodity flow problem efficiently, we present a column generation approach to find continuous minimum-cost multi-commodity flow solutions, which we combine with a price-and-branch procedure to obtain integer solutions. To speed up our column generation, we further develop a pricing filter and an admissible distance approximation to utilize the A^* algorithm for solving the pricing problems. We show the efficiency of our framework by applying it to a real-world case study for the city of Munich, where we solve instances with up to 56,295 passengers to optimality, and show that the computation time of our algorithm can be reduced by up to 60% through the use of our pricing filter and by up to additional 90% through the use of the A^*-based pricing algorithm.

This chapter is based on an article published as: Lienkamp, B., & Schiffer, M. (2024). Column generation for solving large scale multi-commodity flow problems for passenger transportation. European Journal of Operational Research, 314(2), 703-717

3.1 Introduction

According to the United Nations, the global urban population will increase by 900 million from 2020 to 2030, reaching a total of 5.2 billion (UN 2018). This population increase, in combination with already high congestion and traffic emissions in cities all around the world, calls for a better utilization of existing infrastructure and the development of novel transportation concepts to keep up with future urban travel demand. In this context, public transportation services (PTS) play a crucial role in modern cities as they provide an efficient and cost-effective way to meet people's travel demands. They naturally allow for a high utilization, which leads to less congestion, traffic emissions, and transportation costs. In most modern cities, a PTS consists of multiple transportation modes such as subway, tram, and bus lines. By offering a diverse range of transit options, PTS can cater to the unique needs of different passengers for both daily work commutes and leisure trips. Additionally, the different modes of transportation can complement one another, creating a comprehensive transit network that offers multiple routes and options to reach various destinations.

Designing and operating today's PTS to allow for sustainable transport requires advanced planning approaches from both strategic and operational perspectives. The decision support tools for these planning tasks span simulation-based and optimization-based approaches. To this end, simulation-based approaches are used to model daily operations, particularly when aiming to understand passenger mode choice behavior (see, e.g., Hintermann et al. 2023; Huang et al. 2023). Optimization-based approaches often simplify the user behavior and are used to take network design decisions from a strategic perspective (cf. Amberg et al. 2019; Farahani et al. 2013). Between these two purely operational or strategic perspectives, optimization-based approaches that assume a central planner to decide on system optimal passenger flows allow for tactical planning and system analyses (see, e.g. Salazar et al. 2019; Schiffer et al. 2021). Specifically, such approaches are useful for the following purposes: first, planners can use system optimal passenger flows to obtain an upper bound on a PTS's throughput, which allows quantifying the maximum improvement potential within the PTS compared to status quo operations. Second, such system optimal passenger flow information can be used to inform or benchmark operational control algorithms, e.g., pricing mechanisms or online algorithms, that aim to improve the system's performance by influencing user behavior. Third, obtaining information on a system's maximum throughput is helpful for short-term network design decisions, e.g., which line or mode of transport to activate or deactivate during system disruptions such as driver or vehicle shortage, construction work, and medical emergencies. Determining system optimal passenger flows requires solving multi-commodity network flow (MCNF) problems that significantly exceed problem sizes that have so far been solved to optimality. Additionally, the intermodal structure of PTS needs to be considered.

Against this background, we revisit the state of the art on solving MCNF problems for transportation systems. We aim to develop a column generation-based algorithm to find system optima of intermodal transportation networks and improve upon the current state of the art in terms of computational tractability and scalability. In the remainder of this section, we first review related literature, before we define our contributions and the paper's organization.

3.1.1 Related Literature

MCNF models have been used to model a variety of problems, amongst others in communication (Ouorou et al. 2000; Lin 2001; Wagner et al. 2007), logistics (Erera et al. 2005; Al-Khayyal & Hwang 2007; Psaraftis 2011), and transportation systems (Ayar & Yaman 2012; Paraskevopoulos et al. 2016; Zhang et al. 2019). Besides these research fields, MCNF models also find application in financial flows (Yang & Kim 2015), evacuation planning (Pyakurel & Dhamala 2017), production services (Atamtürk & Zhang 2007), and traffic control (Bertsimas & Patterson 2000), which shows the great versatility of MCNF models. Various heuristics, approximations, and exact solution methods have been used to solve different variants of MCNF problems.

The most prominent heuristic methods to solve MCNF problems are genetic algorithms (Khouja et al. 1998; Alavidoost et al. 2018), simulated annealing (Yaghini et al. 2012; Moshref-Javadi & Lee 2016), and tabu search (Ghamlouche et al. 2003; Crainic et al. 2006). Various approximation methods exist for different types of network flow formulations. Lagrangian-based methods proved to be very efficient at solving MCNF problems and find their application in solving multi-commodity capacitated fixed charge network design problems (Crainic et al. 2001) and minimum-cost MCNF problems (Retvdri et al. 2004).

The two most prominent exact approaches to solve MCNF problems are branching and decomposition methods. Branching problems build on the idea of branch-and-bound (B&B) (Lawler & Wood 1966) and have been used to solve integer MCNF problems (see, e.g., Brunetta et al. 2000). Decomposition methods were developed by Dantzig & Wolfe (1960) and used, for example, by Barnhart et al. (1994) to solve message routing problems, formulating them as minimum-cost MCNF problems and solving them via column generation column generation (CG). Barnhart et al. (2000) utilize the branch-and-price (B&P) algorithm (Barnhart et al. 1998), a combination of CG and B&B, to solve integer MCNF problems. Here, the authors applied bounds provided by solving linear programs, using column-and-cut generation at nodes of the branch-and-bound tree in combination with cuts, generated at each node in the B&B tree, to mitigate symmetry effects.

In general, MCNF models have recently been used to model transportation systems, e.g., autonomous Mobility-on-Demand (AMoD) systems in congested road networks (Rossi et al. 2018) and to study the interaction of AMoD with the public transportation system (Salazar et al. 2019) on a mesoscopic level. We refer to Salimifard & Bigharaz (2022) for an extensive overview of applications and solution methods for MCNF problems.

Dynamic flows add a time dimension to static network flow problems and allow flow values on arcs to change over time. In dynamic network flow models for traffic optimization, the size of the underlying graph can be a limiting factor for the model's granularity or time horizon. To address the problem of exponential growth in dynamic networks, Boland et al. (2017) introduced the concept of partially time-expanded networks to service network design problems. In partially time-expanded networks, not all timesteps are included in the network graph, which allows for iterative refinement to obtain a reduced graph size. Besides a reduced size of the underlying graph, multiple methods exist to speed up computation time in large network flow models. For an extensive overview of dynamic network flow problems, we refer to Skutella (2009).

In summary, various optimization approaches exist to solve MCNF problems for different problem settings. So far, using an MCNF model to optimize intermodal passenger transport has only been done on a mesoscopic level (Salazar et al. 2019). Analysis of passenger transport in large transportation networks on a microscopic level is often done via simulation (Ziemke et al. 2019), which is not amenable for optimization. To the best of the authors' knowledge, there exists no optimization framework that is capable of determining the passenger flow system optimum of an intermodal transportation system on a microscopic level.

3.1.2 Contribution and Organization

To close the research gap outlined above, we propose an algorithmic framework to determine the passenger flow system optimum of an intermodal transportation system. Specifically, we combine layered-graph structures and (partially) time-expanded networks, which allows us to implicitly encode problem specific constraints. This enables us to solve an integer minimum-cost MCNF problem to determine travel time optimal passenger flows. To solve this integer minimum-cost MCNF problem efficiently, we present a CG approach to find continuous minimum-cost MCNF solutions, and a price-and-branch (P&B) procedure to obtain integer solutions. We enhance our CG approach by developing a pricing filter and an admissible distance approximation to utilize the A^* algorithm for solving the pricing problems. We show the efficiency of our framework by applying it to a real-world case study for the city of Munich, solving instances with up to 56,295 passengers to optimality, and show that the computation time of our algorithm can be reduced by up to 60% through the use of our pricing filter and by up to additional 90% through the use of the A^*-based pricing algorithm. Furthermore, we showcase how to apply our algorithmic framework from a service provider point of view to analyze passenger utilization of individual routes based on the calculated passenger flows.

The remainder of this paper is as follows. Section 3.2 introduces our problem setting and Section 3.3 focuses on our methodology. Section 3.4 describes our case study for the city of Munich, and Section 3.5 presents numerical results that show the efficiency of our algorithmic framework, and showcase its applicability in practice. Section 3.6 concludes this paper by summarizing its main findings.

3.2 Problem Setting

We study optimal passenger routing in a large intermodal capacitated transportation system with fixed vehicle routes, e.g., bus, subway, and tram lines. An instance of our problem consists of a schedule of fixed vehicle routes in a transportation system and a set of passengers. The vehicle schedules contain information about the stops of each route, arriving times at its stops, and the capacity of the operating vehicle. Each passenger is associated with a transportation request, such that information about a trip's origin and destination coordinates and the departure time is available. In this setting, we aim to find paths for all passengers, such that the sum over all passenger travel times is minimal and all vehicle capacity constraints are kept. We formalize the underlying planning problem as follows.

Notation: We consider a set of passengers $\mathcal{P} = \{1, 2, ..., P\}$ and a set of vehicle route schedules $\mathcal{R} = \{1, 2, ..., R\}$ in the transportation system. Each passenger $p \in \mathcal{P}$ is associated with a request tuple $\zeta_p = (o_p, d_p, \delta_p)$ comprising an origin coordinate o_p, a destination coordinate d_p and a departure time δ_p, which is the timestep in which a passenger wants to begin its trip. Every vehicle route schedule $r \in \mathcal{R}$ contains information about the stops $\mathcal{S}_r$ of the route, the arriving times $T_r^\alpha(s)$ at stops $s \in \mathcal{S}_r$, and the capacity γ_r of the vehicle operating the route. Implicitly, the arrival times indicate in which order a route's stops are visited. Furthermore, we define $\mathcal{S} = \bigcup_{r \in \mathcal{R}} S_r$ as the set of all stops of the transportation system and $T(s) = \bigcup_{r \in \mathcal{R}} T_r^\alpha(s)$ as the set of all timestep in which vehicles arrive at stop $s \in \mathcal{S}$.

Solution: A solution to a problem instance is a set of paths $\boldsymbol{\psi} = (\phi_1, \phi_2, ..., \phi_P)$ with one path for each passenger $p \in \mathcal{P}$. Every path ϕ_p is either a list of tuples $[(o_p, \delta_p), (s_p^1, t_p^1), (s_p^2, t_p^2) ..., (d_p, t_p^{n_p})]$, or an empty set $\emptyset$ if there does not exist a feasible path for passenger p. Here for $i \in [1, ..., n_p - 1]$, $s_p^i \in \mathcal{S}$ is a stop in the transportation system, $t_p^i \in T(s_p^i)$ is a timestep in which a vehicle arrives at stop s_p^i, and $t_p^{n_p} > 0$ is the timestep at which passenger p arrives at its destination coordinate.

Objective: Our objective is to minimize the sum of travel times for all passengers $p \in \mathcal{P}$

$$\min c(\boldsymbol{\psi}) = \min \sum_{p \in \mathcal{P}} c(\phi_p) \tag{3.2.1}$$

Here, $c(\phi_p) = t_p^{n_p} - \delta_p$ is the travel time of passenger $p \in \mathcal{P}$ when using path ϕ_p. If no feasible path was found for pasenger p, i.e., $\phi_p = \emptyset$, we assign $c(\phi_p) = \rho$, with ρ being a predefined penalty.

Constraints: A solution $\boldsymbol{\psi}$ is feasible if the following constraints hold:

(i) The capacity γ_r of a vehicle driving route $r \in \mathcal{R}$ is greater than the number of passengers using the route at each point in time.

(ii) For each tuple pair $(s_p^i, t_p^i), (s_p^{i+1}, t_p^{i+1})$, $i \in [1, ..., n_p - 1]$ in ϕ_p with $p \in \mathcal{P}$, there

  (a) either exists a route $r \in \mathcal{R}$ where the vehicle departs from s_p^i in timestep t_p^i to arrive at s_p^{i+1} in t_p^{i+1} without any stopovers,

  (b) or the distance between s_p^i and s_p^{i+1} is smaller than a given maximum walking distance (Δ_w) and $t_p^{i+1} - t_p^i$ is at minimum the time necessary to walk from s_p^i to s_p^{i+1} with given walking speed ξ,

  (c) or the passenger waits at the stop, such that $s_p^i = s_p^{i+1}$ and $t_p^i > t_p^{i+1}$.

(iii) The distance between o_p and s_p^1 is smaller than a maximum access distance Δ_a.

(iv) The distance between $s_p^{n_p-1}$ and d_p is smaller than a maximum egress distance Δ_e.

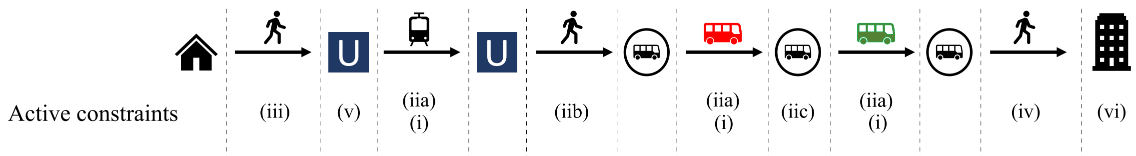
(v) The time $t_p^1 - \delta_p$ until a vehicle arrives at the first stop s_p^1 when a passenger first enters the transportation system is smaller than a given maximum waiting time Υ.

(vi) $t_p^{n_p} \leq \delta_p + t_{\max}$, where $t_{\max}$ is the maximum travel time a passenger is allowed to take for its trip.

Figure 1 illustrates these constraints. Assume a trip from a passenger's home to their office: they first walk to a subway station with at most a distance Δ_a (iii). The passenger must then wait at most Υ timesteps for the first subway to arrive (v). To go from the first subway stop to the second subway stop at timestep t_p^i, there has to exist a subway route that arrives at the first subway stop at timestep t_p^i and at the second subway stop at t_p^{i+1} (iia). In this context, we note that different routes can belong to the same line, and we differentiate them by their schedule. Each route r has an individual capacity γ_r (i), i.e., the subway route that belongs to the U5 line and leaves at 8:00am may have a different capacity than the route that runs on the same line but leaves at 8:05am. After the passenger arrives at the second subway stop, they walk to a nearby bus stop with a maximum distance of Δ_w to the subway stop (iib). The passenger takes a bus to another bus stop, where they leave the first bus and wait for another bus (iic). After they took the second bus to the last bus stop, they walk to their office, which has a maximum distance of Δ_e to the last bus stop (iv). The passenger has to arrive after at most $t_{\max}$ timesteps after they leave their home (vi).

Discussion: Four comments on this problem setting are in order. First, we assume that information about stops and arrival times at the stops is available for all vehicle routes. This assumption holds for scheduled transportation modes, e.g., bus, subway, and tram. Second, for each route, we assume fixed arc capacities with fixed travel times instead of flow-dependent travel times. This approximation is consistent with travel time observation on real-world data. Here, the number of passengers utilizing a public transportation vehicle does not influence the travel time of a vehicle. Furthermore, our model is able to incorporate time-dependent travel times straightforwardly. Third, we assume a fixed maximum travel time $t_{\max}$ for all passengers. In practice, our approach is not limited to a homogeneous $t_{\max}$ but allows to choose $t_{\max}$ passenger-dependent. From an algorithmic perspective, a passenger-dependent $t_{\max}$ reduces the size of the problem‘s multilayered digraph. Henceforth, we decided to choose a homogeneous and large $t_{\max}$ in order to challenge our algorithmic

Figure 1: Illustrative example of a passenger trip



framework. Finally, we note that our problem setting, as well as the following algorithmic framework, can be tailored to minimize cost instead of time without major modifications. However, since passengers usually pay a fixed fee to travel within certain zones of a PTS, we decided to focus our basic problem setting on minimizing time, which is the quantity sensitive to the respective route choice.

3.3 Methodology

In the following, we introduce an algorithmic framework (see Figure 2) to solve the planning problem introduced in Section 3.2. Intuitively, we can obtain a solution ψ, i.e., paths for all passengers such that the sum over all travel times is minimal, by solving an integer minimum-cost MCNF problem.

In a general minimum-cost MCNF problem let κ_{ij} be the capacity of an arc $a = (i,j)$ with $i, j \in \mathcal{V}$, d_i^k be the vertex demand of vertex $i \in \mathcal{V}$ and commodity $k \in \mathcal{K}$, and $x_{ij}^k \in \{0,1\}$ be the decision variable indicating whether commodity $k \in \mathcal{K}$ uses arc $(i,j) \in \mathcal{A}$ ($x_{ij}^k = 1$) or not ($x_{ij}^k = 0$). The vertex demand for commodity k at vertex $i \in \mathcal{V}$ is defined as

$$d_i^k = \begin{cases} 1 & \text{if } i = u_k, \\ -1 & \text{if } i = v_k, \\ 0 & \text{otherwise} \end{cases}$$

Here, $\mathcal{K}$ is a set of commodities, u_k the source, and v_k the sink of commodity $k \in \mathcal{K}$. In IP 1, the objective function minimizes the sum over the cost of all commodity flows, while Constraints (1b) ensure flow conservation with $\mathcal{N}^+(i)/\mathcal{N}^-(i)$ being the outgoing/ingoing neighbourhood of $i \in \mathcal{V}$. Note that an ingoing neighbor of i is a vertex $j \in \mathcal{V}$ such that $(j,i) \in \mathcal{A}$; an outgoing neighbor of i is a vertex j such that $(i,j) \in \mathcal{A}$. The remaining constraints enforce the capacity constraints of all arcs (1c) and ensure integer commodity flows (1d).

Figure 2: Flowchart of our algorithmic framework

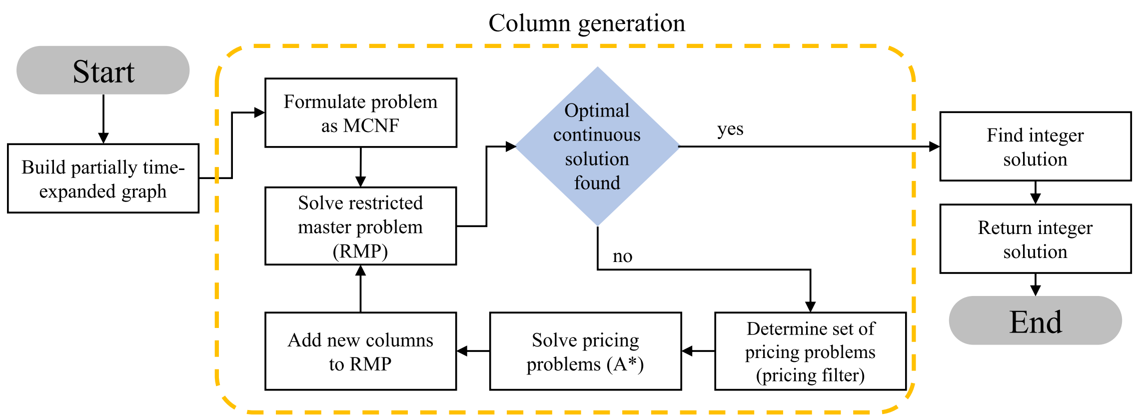
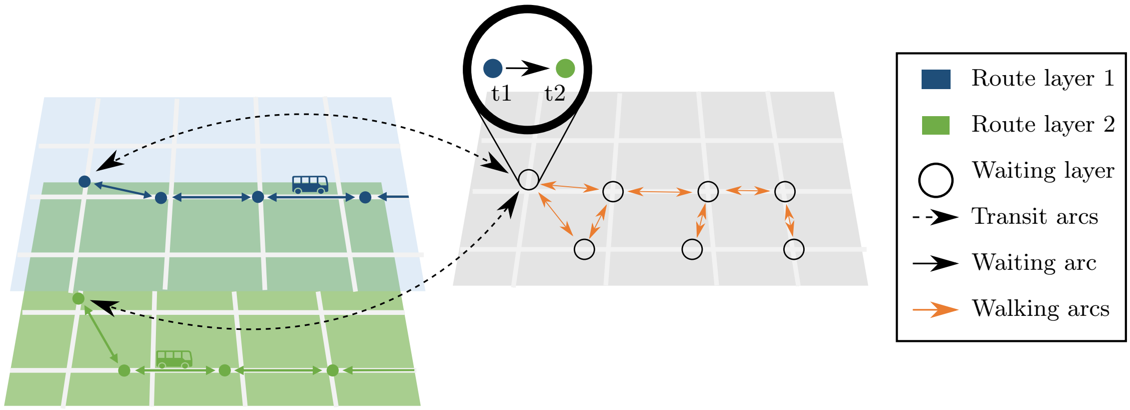


Figure 3: Scematic illustration of the multilayered network structure



$$
\begin{aligned}
\min_x \quad & \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij}\ x_{ij}^k && && (1a) \\
\text{s.t.} \quad & \sum_{j \in \mathcal{N}^+(i)} x_{ij}^k - \sum_{j \in \mathcal{N}^-(i)} x_{ji}^k = d_i^k, && i \in \mathcal{V}, k \in \mathcal{K} && (1b) \\
& \sum_{k \in \mathcal{K}} x_{ij}^k \leq \kappa_{ij}, && (i,j) \in \mathcal{A} && (1c) \\
& x_{ij}^k \in \{0,1\}, && (i,j) \in \mathcal{A}, k \in \mathcal{K} && (1d)
\end{aligned}
\qquad (\text{IP } 1)
$$

To utilize such a minimum-cost MCNF formulation for our problem setting, it is necessary to model the transportation system as a digraph, which includes origin and destination vertices for all passengers, considers the intermodality of our problem setting, and encodes the spatial and temporal route constraints.

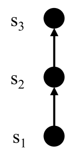
We introduce our methodology in three steps. First, we introduce the graph formulation that combines two core principles of network optimization: layered-graph structures and (partially) time-expanded networks. This allows us to implicitly encode problem specific constraints related to the intermodality of our planning problem and enables us to solve a standard integer MCNF problem to determine a solution $\boldsymbol{\psi}$. Second, we present a CG procedure to solve continuous MCNF problems. In this context, we introduce our pricing filter and our A^*-based pricing problem. Third, we elaborate on how to obtain integral solutions through the application of a P&B approach.

3.3.1 Graph formulation

To solve the planning problem as introduced in Section 3.2 with a standard minimum-cost MCNF formulation, we capture network specific constraints implicitly in the underlying graph structure, independently from the MCNF formulation. To do so, we model the transportation system as a multilayered digraph as schematically shown in Figure 3.

This digraph contains a route layer for each route in the transportation system, which allows passengers to use a vehicle operating a route. Additionally, waiting layers for each stop in the

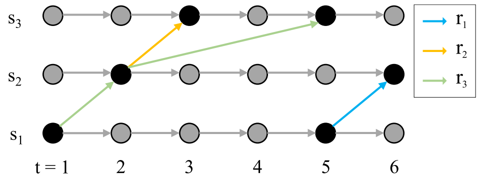
Figure 4: Illustration of the time-expansion of a static graph where black vertices and colored arcs in the time-expansion indicate the necessary elements to model given vehicle routes



(a) Static graph

	r_1	r_2	r_3
s_1	5	-	1
s_2	6	2	2
s_3	-	3	5

(b) Arrival times for vehicle routes

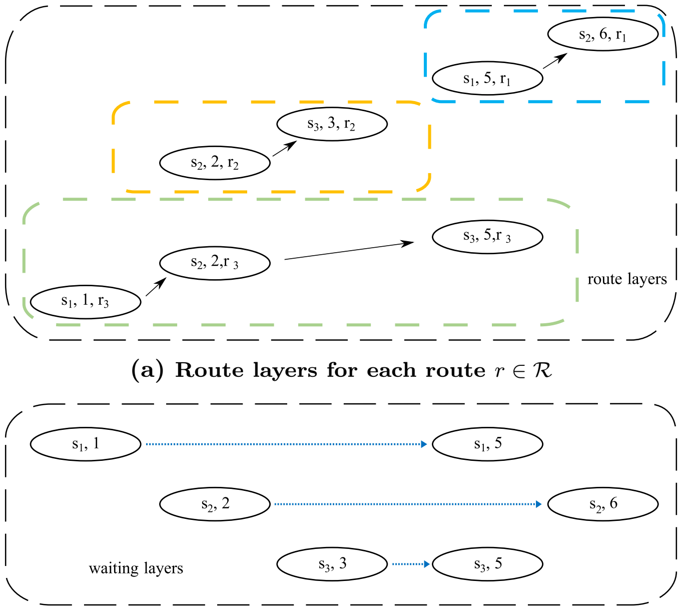


(c) Time expanded graph

transportation system allow passengers to wait at a stop for a new vehicle to arrive. Transit arcs enable passengers to switch between route and waiting layers and thus to enter or leave a vehicle at a stop. Walking arcs between different stop layers allow passengers to walk between stops in a certain proximity.

Such a multilayered transportation network can be modeled as a static digraph $\mathcal{G}_\sigma = (\mathcal{V}_\sigma, \mathcal{A}_\sigma)$ with a set of vertices $\mathcal{V}_\sigma$ and a set of arcs $\mathcal{A}_\sigma$. Here, the vertex set $\mathcal{V}_\sigma$ represents the stops $\mathcal{S}$ of the transportation system and an arc $a = (s_1, s_2) \in \mathcal{A}_\sigma$ indicates that a route $r \in \mathcal{R}$ exists, covering two consecutive stops s_1 and s_2. The drawback of such a static graph formulation is the missing time dimension such that the static graph formulation does not reflect the vehicle schedules. Time-expanded graphs as introduced by Ford Jr & Fulkerson (1958) allow for the integration of the time dimension (see Figure 4). A time-expanded graph $\mathcal{G}_\Theta = (\mathcal{V}_\Theta, \mathcal{A}_\Theta)$ contains one copy of the vertex set of an underlying static graph $\mathcal{V}_\sigma$ for each timestep. Additionally, arcs connect vertices in different time steps to account for the transit time.
While a time-expanded graph allows to implicitly model time, its size can grow rapidly even for relatively small underlying static graphs. To mitigate this drawback, we extend our layered graph with a partial time expansion (Boland et al. 2017). Here, the general idea is to copy only vertices at which a vehicle arrives at timestep t (black vertices in Figure 4c) and to only add a temporal arc $a = ((s_1, t_1)(s_2, t_2))$ to the time-expanded graph if a route covers stop s_1 in timestep t_1 and subsequently stop s_2 in timestep t_2 (colored arcs in Figure 4c). Such a graph formulation keeps all relevant vertices and arcs in its temporal expansion but the cardinality of the respective vertex and arc set is significantly smaller than for a fully time-expanded graph.
Accordingly, we represent an intermodal transportation system as a partially time-expanded multi-layered digraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with a set of temporal vertices $\mathcal{V}$ and a set of temporal arcs $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$. Let $\mathcal{S}$ be the set of all stops of the transportation network and let $\mathcal{T} = \{\bigcup_{p \in \mathcal{P}} \delta_p\} \cup \{\bigcup_{s \in \mathcal{S}} T(s)\} \cup \{\bigcup_{p \in \mathcal{P}} \delta_p + t_{\max}\}$ be the set of all relevant timesteps. This set consists of all passenger departure times δ_p, all timesteps during which any vehicle arrives at any station $T(s)$, and the latest timestep in which each passenger is allowed to arrive at its destination $\delta_p + t_{\max}$. Then, the graph $\mathcal{G}$ contains a route layer $G_r = (V_r, A_r)$ for each vehicle route $r \in \mathcal{R}$ and a waiting layer $G_s = (V_s, A_s)$ for each stop $s \in \mathcal{S}$. In the remainder of this subsection, we detail the construction of $\mathcal{G}$.

Figure 5: Route layers and waiting layers based on schedule of Figure 4b



(a) Route layers for each route $r \in \mathcal{R}$

(b) Waiting layers for each stop $s \in \mathcal{S}$

3.3.1.1 Construction of route and waiting layers

The route layers $G_{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} G_r$ contain temporal route vertices $V_{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} V_r$ and arcs $A_{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} A_r$. A temporal route vertex $i = (s, t, r) \in V_r$ of a route layer G_r with $r \in \mathcal{R}$ represents the vehicle of route r arriving at stop $s \in \mathcal{S}$ at timestep $t \in \mathcal{T}$. Consecutive stops of route r represented by temporal nodes $i, j \in V_r$ with $i = (s_i, t_i, r)$ and $j = (s_j, t_j, r)$ are connected via a temporal arc $a = (i, j) \in A_r$. Each arc in A_r has a cost $c_{ij} = t_j - t_i$ and a capacity γ_r which is equal to the capacity of the vehicle operating route r. This formulation allows us to consider different transportation modes and vehicle capacities between two stops.

The waiting layers $G_{\mathcal{S}} = \bigcup_{s \in \mathcal{S}} G_s$ contain temporal waiting vertices $V_{\mathcal{S}} = \bigcup_{s \in \mathcal{S}} V_s$ and arcs $A_{\mathcal{S}} = \bigcup_{s \in \mathcal{S}} A_s$ which allows passengers to wait at a stop. A temporal waiting vertex $i = (s, t) \in V_s$ is a tuple comprising a stop $s \in \mathcal{S}$ and a timestep $t \in \mathcal{T}$. For each timestep $t \in \mathcal{T}$ in which a vehicle of route $r \in \mathcal{R}$ arrives at stop $s \in \mathcal{S}$, there exists a temporal vertex $i = (s, t) \in V_s$ in the waiting layer of stop s. Consecutive temporal vertices are connected by waiting arcs $a \in A_s$. Waiting arcs have unlimited capacity, as we assume sufficient space at the stops for all passengers. The cost of waiting arcs results analogously to route arcs. Figure 5 shows the route and waiting layers based on the schedule of Figure 4.

3.3.1.2 Construction of transit and walking arcs

So far all route layers and waiting layers are disconnected. Transit arcs A_T connect these different route and waiting layers and can either point from a route layer G_r, $r \in \mathcal{R}$ to a waiting layer G_s, $s \in \mathcal{S}$ or vice versa. Such a transit arc $a = (i, j) \in A_T$ from route node $i = (s, t, r) \in V_r$ to waiting node $j = (s, t) \in V_s$ has infinite capacity and cost $c_{ij} = 0$. For every such transit arc $a = (i, j)$

pointing from a route layer to a waiting layer, there exists a corresponding transit arc $a' = (j, i)$ from the waiting layer to the route layer with identical capacity and cost. Walking arcs A_W connect waiting nodes of stops within a predefined walking distance Δ_w. Here, waiting nodes $i = (s, t_i)$ and $j = (s', t_j)$ of different stops $s, s' \in \mathcal{S}$ with $dist(s, s') \leq \Delta_\text{w}$ are connected with a walking arc $a = (i, j)$ with cost $c_{ij} = t_j - t_i$ and unconstrained capacity, if $t_i + \frac{dist(s,s')}{\xi} \leq t_j$. Here, $dist(s, s')$ denotes a distance measure between the stop coordinates of two stops $s, s' \in \mathcal{S}$.
The resulting graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with $\mathcal{V} = V_\mathcal{R} \cup V_\mathcal{S}$ and $\mathcal{A} = A_\mathcal{R} \cup A_\mathcal{S} \cup A_T \cup A_W$ represents the temporal and spatial expansion of the transportation network.

3.3.1.3 Construction of access and egress arcs

To formulate the underlying optimization problem as an MCNF problem, we add the origin vertex $\omega_p^o = (o_p, \delta_p)$ and the destination vertex $\omega_p^d = (d_p, \delta_p + t_\text{max})$ of each passenger $p \in \mathcal{P}$ to $\mathcal{G}$. We can now connect ω_p^o to all stops $s \in \mathcal{S}$ with $dist(o_p, s') \leq \Delta_\text{a}$, for which a waiting node $v = (s, t)$ with $\delta_p + \frac{dist(o_p,s)}{\xi} \leq t \leq \delta_p + \Upsilon$ exists. These access arcs $a = (\omega_p^o, v)$ have cost $c_{\omega_p^o,v} = t - \delta_p$. Egress arcs connecting the transportation network with the destination vertex ω_p^d originate from all waiting nodes $v = (s, t) \in V_s$ of a stop $s \in \mathcal{S}$ with $dist(s, d_p) \leq \Delta_\text{e}$ for which it holds that $\delta_p \leq t + \frac{dist(s,d_p)}{\xi} \leq \delta_p + t_\text{max}$. These egress arcs have cost $c_{v,\omega_p^d} = \frac{dist(s,d_p)}{\xi}$.

Thus, we expand the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ by the set of origin vertices V_O and the set of destination vertices V_D as well as the access arcset A_O and the egress arcset A_D resulting in $\mathcal{V} = V_\mathcal{R} \cup V_\mathcal{S} \cup V_O \cup V_D$ and $\mathcal{A} = A_\mathcal{R} \cup A_\mathcal{S} \cup A_T \cup A_W \cup A_O \cup A_D$. Each passenger travel demand is now represented as an individual flow going from an origin vertex to a destination vertex with flow demand one. This allows us to model our problem as a minimum-cost MCNF problem as introduced in IP 1, where $\mathcal{K} = \mathcal{P}$ and the origin and destination vertices correspond to sources and sinks for each commodity flow.

3.3.1.4 Illustrative example

The following example illustrates the resulting graph $\mathcal{G}$ for a transportation network. Table 1a depicts the timetables for three vehicle routes $\mathcal{R} = \{r_1, r_2, r_3\}$, which operate on a set of stops $\mathcal{S} = \{s_1, s_2, s_3\}$. In our example, we assume a capacity of $\gamma_r = 60$. We only consider one passenger with request tuple $\zeta_p = (o_p, d_p, 0)$. Table 1b depicts the access and egress distances, as well as the walking distances between the different stops.

Table 1: Network characteristics

(a) Arrival times for vehicle routes

	r_1	r_2	r_3
s_1	5	-	1
s_2	6	2	2
s_3	-	3	5

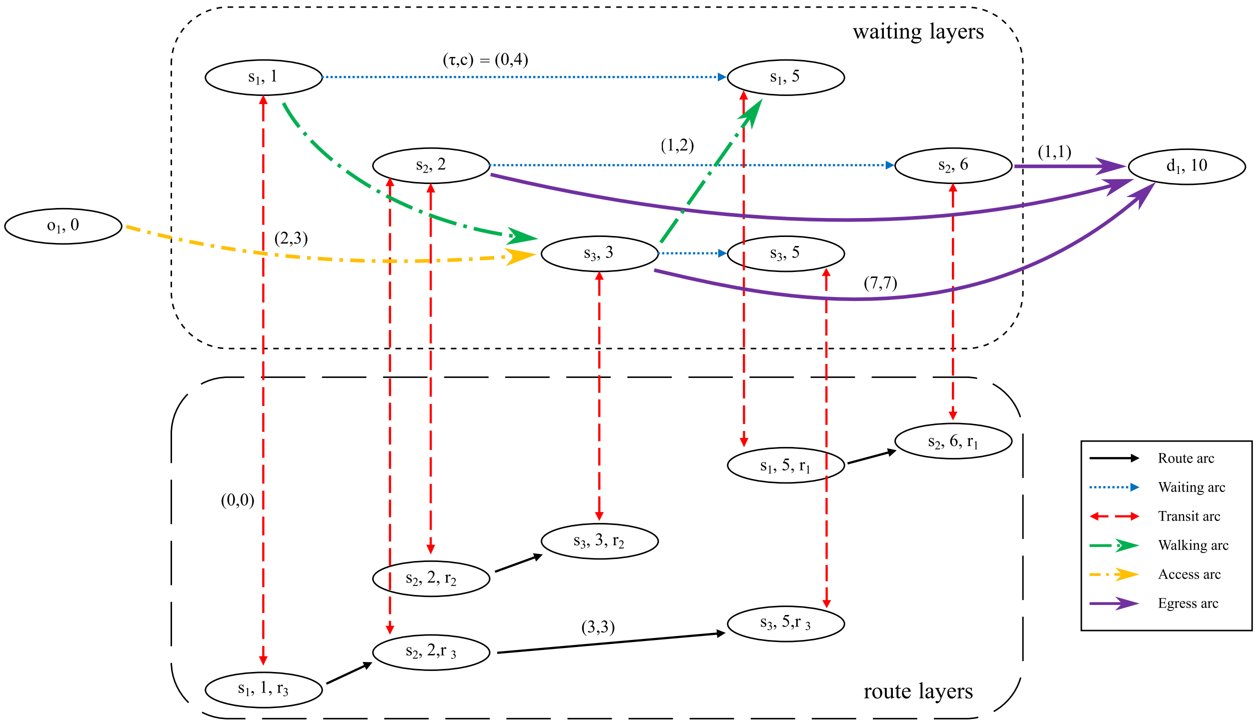
(b) Distances in the transportation network

	s_1	s_2	s_3	o_1	o_2
s_1	0	6	1	3	10
s_2	6	0	4	5	1
s_3	1	4	0	2	7

We assume a walking speed $\xi = 1$ for access, egress and walking distances. Furthermore, we set the maximum access distance $\Delta_{\mathrm{a}} = 3$, the maximum egress distance $\Delta_{\mathrm{e}} = 7$, and the maximum walking distance $\Delta_{\mathrm{w}} = 4$. Additionally, we set the maximum waiting time $\Upsilon = 4$ and the maximum travel time $t_{\max} = 10$. Figure 6 shows the multilayered partially time-expanded graph for this example.

For each route arc the travel time τ and the arc cost c coincide, see, e.g., arc $((s_2, 2, r_3),(s_3, 5, r_3))$. Nodes from the route layers are connected to nodes from the waiting layers via transit arcs. These transit arcs have a travel time and cost of zero, see, e.g., arc $((s_1, 1, r_3),(s_1, 1))$. Nodes in the same waiting layer are connected via waiting arcs, all having a travel time of zero, but an arc cost equivalent to the waiting time, see, e.g., arc $((s_1, 1),(s_1, 5))$. The origin vertex $(o_1, 0)$ is only connected to node $(s_3, 3)$ with travel time two and arc cost three. Note that even though s_1 has an access distance of three, there is no access arc to this waiting layer because the first possible departure after the passenger arrives at s_1 is at timestep five which exceeds $\Upsilon = 4$. There are two walking arcs between the stop layer of s_1 and s_3. Both walking arcs have a transit time of one and an arc cost of two. Note that even though the walking distance between s_2 and s_3 is smaller than Δ_{w}, there exist no walking arcs between these two waiting layers because the travel time is too long, e.g., a passenger which starts at $(s_2, 2)$ can earliest arrive at s_3 at timestep six but the latest possible timestep at which to arrive at the waiting layer of s_3 is five. The destination vertex $(d_p, 10)$ has three ingoing egress arcs. For each egress arc the travel time and arc cost coincide. There exists no egress arc from $(s_3, 5)$ because a passenger cannot arrive at the destination before timestep 12,

Figure 6: Partially time-expanded multilayered digraph of the example with travel time τ and arc cost c



which is greater than $\delta_p + t_{\max} = 10$. Thus, the two feasible paths for the passenger are

$$\eta_1 = ((o_1, 0), (s_3, 3), (d_1, 10))$$
$$\eta_2 = ((o_1, 0), (s_3, 3), (s_1, 5), (s_1, 5, r_1), (s_2, 6, r_1), (s_2, 6), (d_1, 10))$$

Here, the cost of η_1 is ten and the cost of η_2 is seven. Thus, η_2 can be used to construct path $\phi_1 = [(o_1, 0), (s_3, 3), (s_1, 5), (s_2, 6), (d_1, 7)]$ for our problem setting. Note that the elements of ϕ_1 do not correspond to nodes of Figure 6.

3.3.2 Column generation

We utilize the introduced minimum-cost MCNF formulation IP 1 to find a path from each origin vertex ω_p^o to its respective destination vertex ω_p^d in our partially time-expanded multilayered digraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. For large sets of passengers, Constraints (1b) in IP 1 grow exponentially, which makes this standard model formulation of minimum-cost MCNF problems non-tractable for large instances. To mitigate this issue, we reformulate IP 1 as a set-covering problem to solve it via CG, which can be applied to linear programs (LPs) with a large set of decision variables. Here, the idea is to solve the LP with only a subset of all variables and iteratively add new variables to the program which have the potential to improve the objective function. New promising variables can be found by solving so called pricing problems. If one can show that adding new variables cannot improve the objective value, the algorithm terminates.

IP 2 denotes the set covering reformulation of IP 1. Here, L_p denotes the set of different paths which exist for a passenger $p \in \mathcal{P}$. Parameter $(y_l^p)_{ij}$ shows whether path l of passenger p utilizes arc $(i,j) \in \mathcal{A}$ ($(y_l^p)_{ij} = 1$) or not ($(y_l^p)_{ij} = 0$) and we use decision variable λ_l^p to select exactly one path $l \in L_p$ for each passenger p. Furthermore, κ_{ij} represents the capacity of arc $(i,j) \in \mathcal{A}$.

$$\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & \sum_{p\in\mathcal{P}} \sum_{l\in L_p} \lambda_l^p \sum_{(i,j)\in\mathcal{A}} c_{ij} (y_l^p)_{ij} && && (2a) \\
\text{s.t.} \quad & \sum_{p\in\mathcal{P}} \sum_{l\in L_p} (y_l^p)_{ij} \lambda_l^p \leq \kappa_{ij} && \forall (i,j) \in \mathcal{A} && (2b) \\
& \sum_{l\in L_p} \lambda_l^p = 1 && \forall p \in \mathcal{P} && (2c) \\
& \lambda_l^p \in \{0,1\} && \forall p \in \mathcal{P}, \forall l \in L_p && (2d)
\end{aligned} \qquad \text{(IP 2)}$$

To apply CG, we relax the integer constraint (2d) to a non-negativity constraint and solve the continuous relaxation of IP 2. Note that the solution of the continuous relaxation is a lower bound of the integer solution and thus the system optimum. We add dummy variables with objective costs greater than $t_{\max}$ for each passenger. This allows us to initiate the CG algorithm with $L_p = \emptyset \; \forall p \in P$, i.e., without knowing any paths of the passengers. Then, the resulting restricted master problem (RMP) based on IP 2 is as follows

$$
\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & \sum_{p \in \mathcal{P}} \rho^p \lambda^p_{init} + \sum_{p \in P} \sum_{l \in L_p} \lambda^p_l \sum_{(i,j) \in \mathcal{A}} c_{ij} (y^p_l)_{ij} && && (3a) \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}} \sum_{l \in L_p} (y^p_l)_{ij} \lambda^p_l \leq \kappa_{ij} && \forall (i,j) \in \mathcal{A} && (3b) \\
& \lambda^p_{init} + \sum_{l \in L_p} \lambda^p_l = 1 && \forall p \in \mathcal{P} && (3c) \\
& \lambda^p_l \geq 0 && \forall p \in \mathcal{P}, \forall l \in L_p && (3d) \\
& \lambda^p_{init} \geq 0 && \forall p \in \mathcal{P} && (3e)
\end{aligned}
\qquad \text{(RMP)}
$$

Here, ρ^p is the penalty which occurs if passenger $p \in \mathcal{P}$ is not able to reach its destination within t_{max} timesteps. Accordingly, we add the expression $\sum_{p \in P} \rho^p \lambda^p_{init}$ to the objective function of our RMP to account for these infeasible trips. By solving LP 4, i.e., the pricing problem, for each passenger $p \in \mathcal{P}$, we then iteratively add new paths to L_p and thus new columns to our RMP.

$$
\begin{aligned}
\min_y \quad & \sum_{(i,j) \in A} (c_{ij} - w^*_{ij}) y^p_{ij} - \alpha^*_p && && (4a) \\
\text{s.t.} \quad & \sum_{p \in P} y^p_{ij} \leq \kappa_{ij}, && (i,j) \in A && (4b) \\
& \sum_{j \in N^+(i)} y^p_{ij} - \sum_{j \in N^-(i)} y^p_{ji} = d^p_i, && i \in V, p \in P && (4c) \\
& y^p_{ij} \in \{0,1\}, && (i,j) \in A, p \in P && (4d)
\end{aligned}
\qquad \text{(LP 4)}
$$

Here, w^*_{ij} is the the dual variable for the capacity constraints (3b) of arc $(i,j) \in \mathcal{A}$ and α^*_p the dual variable for the convexity constraint (3c) for passenger $p \in \mathcal{P}$. Decision variables y^p_{ij} determine whether the path we find for passenger p uses arc (i,j) ($y^p_{ij} = 1$) or not ($y^p_{ij} = 0$). Constraints (4c) ensure flow conservation for all vertices. We define vertex demand d^p_i as follows

$$
d^p_i = \begin{cases} 1 & \text{if } i = \omega^o_p, \\ -1 & \text{if } i = \omega^d_p, \\ 0 & \text{otherwise} \end{cases}
$$

If the objective value for the pricing problem of passenger p is negative, we add y^p as a new path to L_p in our RMP and expand the decision variables λ^p_l to account for the new found path. Notice that the constraint $\lambda^p_l \leq 1, \forall p \in \mathcal{P}, \ \forall l \in L_p$ is redundant in the RMP because of Constraints (3c).

Algorithm 1 shows the pseudocode of our CG approach. We start by initializing the RMP (l. 1) and setting the lower bound (LB) of the continuously relaxed master problem to zero (l. 2). In every iteration, we solve the current *RMP* (l. 4) and ensure its feasibility (l. 5). Afterwards two stopping criteria are checked. The first stopping criterion checks if the dual variables of the RMP did not change after it was solved (l. 8). If the dual variables of the RMP did not change, no new columns can be found. The second stopping criterion checks if the lower bound LB is equal to the current objective function $sol.objVal$ of the RMP. If one of the two stopping criteria is fulfilled, we found an optimal solution for the continuous master problem and return it (l. 9). If the algorithm did not terminate in the current iteration, we define a pricing pool (Π) which in its basic variant is equal to the set of all passengers $\mathcal{P}$ (l. 11). After that, we set a new variable β to the current objective value of the RMP which is used to try to find a new lower bound for the continuously relaxed master problem (l. 12). Afterwards, the pricing problems of all passengers in Π are solved (l. 14). If the objective value of the pricing problem is negative, a new column is added to the RMP (l. 16). Here, we add a new column to the RMP by:

(i) Adding y^p as a new path to L_p.

(ii) Adding a new decision variable λ_l^p to the formulation to account for the new found path.

After all new paths are added as new columns to the RMP, we update the RMP's lower bound (l. 20-22). When the algorithm solved all pricing problems and updated the lower bound, we start a new iteration of the CG (l. 4).

While this CG implementation is straightforward, its computing time can be improved by utilizing a pricing filter to reduce the size of Π (l. 11), which leads to fewer pricing problems that need to be solved (l. 13-19), and by using an A^*-based pricing algorithm instead of a straightforward Dijkstra-based pricing algorithm (Dijkstra 1959) to solve the pricing problems. In the remainder of this section, we elaborate on both of these improvement levers.

Algorithm 1 Column generation pseudocode

```
1: RMP ← build RMP based on IP 2
2: LB ← 0
3: while TRUE do
4:     sol ← solve(RMP)
5:     if RMP infeasible then
6:         RETURN Error: RMP infeasible
7:     end if
8:     if Dual variables of RMP did not change or LB = sol.objVal then
9:         RETURN sol
10:    end if
11:    Π ← P
12:    β = RMP.objVal
13:    for p ∈ Π do
14:        pp ← solvePricingProblem(p)
15:        if pp.objVal < 0 then
16:            addNewColumn(RMP)
17:            β += pp.ObjVal
18:        end if
19:    end for
20:    if β > LB then
21:        LB ← β
22:    end if
23: end while
```

3.3.2.1 Pricing Filter

Solving the pricing problem $|\Pi|$ times in every iteration of the CG is a key bottleneck of Algorithm 1. To mitigate this bottleneck, we reduce the pricing pool size $|\Pi|$ by adding a pricing filter. To derive such a pricing filter, we analyse the dual variables of the capacity constraints (3b). Intuitively, a negative dual capacity variable w^*_{ij} indicates that more passengers try to use arc $a = (i, j)$ than the arc capacity κ_{ij} allows. Accordingly, we add all passengers with a possible path which uses an arc with a negative dual capacity variable to the pricing pool. Note that in this case, the decision variable of the determining path does not have to be in the basis of the LP solution.

Algorithm 2 shows the pseudocode of our pricing filter. We iterate through all arcs $a \in \mathcal{A}$ of graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ (l. 3) and add arcs with a negative dual capacity constraint to the set Φ (l. 5). Next, we iterate through all found paths $l \in L = \bigcup_{p\in\mathcal{P}} L_p$ collect all arcs that the paths use (l. 9-12). If a path uses an arc with a negative dual capacity value, we add the corresponding passenger to Π (l. 13-15). Note that when applying our pricing filter, we have to run the pricing problems for all passengers again, as soon as the objective value of the RMP does not improve anymore, to ensure optimality.

3.3.2.2 Pricing Problem

Recall that we defined our MCNF problem such that each request has a flow value of one and each arc has a capacity of at least one. Accordingly, the capacity constraint (4b) is always fulfilled and we can solve the pricing problem LP 4 as a shortest path problem, in which we aim to find the shortest path from the origin to the destination node for every passenger $p \in \mathcal{P}$, with a new arc cost $c'_{ij} = c_{ij} - w^*_{ij}$ for all arcs $(i, j) \in \mathcal{A}$. In this setting, a negative objective value of the pricing problem LP 4 corresponds to the shortest path from (o_p, δ_p) to $(d_p, \delta_p + t_{\max})$ in graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with arc costs $\boldsymbol{c'}$ being shorter than a^*_p. Generally, constraint (3b) and the constraints (3d) ensure that this approach can also be used when not all flow values are fixed to one.

Algorithm 2 Pricing filter algorithm

1: $\Pi \leftarrow \emptyset$
2: $\Phi \leftarrow \emptyset$
3: **for** $(i, j) \in \mathcal{A}$ **do**
4:     **if** $w^*_{ij} < 0$ **then**
5:         Φ.add((i, j))
6:     **end if**
7: **end for**
8: **for** $Y_v \in Y$ **do**
9:     usedArcs $\leftarrow \emptyset$
10:     **if** $(y^p_l)_{ij}) > 0$ **then**
11:         usedArcs.add((i, j))
12:     **end if**
13:     **if** usedArcs $\cap \Phi \neq \emptyset$ **then**
14:         Π.add(corresponding passenger p of $(y^p_l)_{ij})$)
15:     **end if**
16: **end for**

A straightforward approach to solve the pricing problem (LP 4) as a shortest path problem is to use the Dijkstra algorithm. However, applying a straightforward Dijkstra finds its limits for large scale instances due to the size of $\mathcal{G}$. Here, we note that we cannot increase the Dijkstra's algorithm's efficiency by topological ordering as $\mathcal{G}$ contains cycles induced by the transit arcs. Accordingly, we utilize the A^* algorithm to increase the efficiency of the shortest path calculation.

To apply the A^* algorithm, we need to define an admissible distance approximation h, which the A^* algorithm uses to guide its search. This distance approximation h does not overestimate the cost of the shortest path from all nodes $i \in \mathcal{G}$ to all destination vertices $j \in V_D$ of all passengers in graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with cost function $c'_{ij} = c_{ij} - w^*_{ij}$ for all arcs $(i, j) \in \mathcal{A}$. The more accurate the distance approximation is, the fewer nodes A^* has to expand to find the shortest path.

In the following, we reduce our time-expanded multilayered digraph to a significantly smaller static digraph which allows us to introduce an admissible distance approximation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be the graph defined in Section 3.3.1 and let $\mathcal{H} = (\mathcal{V}_H, \mathcal{A}_H)$ be a new graph. Let $\mathcal{V}_H = \mathcal{S} \cup \mathcal{V}_D$ be the union of the set of all stops $\mathcal{S}$ of the transportation system and all destination vertices $\mathcal{V}_D$. Furthermore, for $i, j \in \mathcal{V}_H$ let A_{ij} be the set of all arcs $(a, b) \in \mathcal{A}$ of one of the following types

(i) $a, b \in \mathcal{V}_\mathcal{R}$, with $a = (i, t_a, r_a)$, $b = (j, t_b, s_b)$

(ii) $a \in \mathcal{V}_\mathcal{R}$, $b \in \mathcal{V}_\mathcal{S}$, with $a = (i, t_a, r_a)$, $b = (j, t_b)$

(iii) $a \in \mathcal{V}_\mathcal{S}$, $b \in \mathcal{V}_\mathcal{R}$, with $a = (i, t_a)$, $b = (j, t_b, s_b)$

(iv) $a \in \mathcal{V}_\mathcal{S}$, $b \in \mathcal{V}_D$, with $a = (i, t_a)$, $b = (j, t_b)$

If $|A_{ij}| > 0$, we add arc (i, j) to $\mathcal{A}_H$. The cost of arc (i, j) is defined as $c^\mathcal{H}_{ij} = \min\{c_{ab} | (a, b) \in \mathcal{A}_{ij}\}$.

The following first two definitions, introduce the stop of a temporal route or waiting vertex and the pricing cost of the arcs of our time-expanded digraph. This allows us to propose a distance approximation in the third definition.

Definition 3.3.1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be the graph defined in Section 3.3.1 and let $a \in \mathcal{V}_\mathcal{R} \cup \mathcal{V}_\mathcal{S}$ be a vertex. We define the stop of a as*

$$\eta(a) = \begin{cases} s_a, & \text{if } a = (s_a, t_a, r_a) \in \mathcal{V}_\mathcal{R} \\ s_a, & \text{if } a = (s_a, t_a) \in \mathcal{V}_\mathcal{S} \end{cases} \tag{3.3.1}$$

Definition 3.3.2. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be the graph defined in Section 3.3.1 and let $i, j \in \mathcal{V}$ be two vertices. We define the pricing cost c' of graph $\mathcal{G}$ as $c'_{ij} = c_{ij} - w^*_{ij}$, $(i, j) \in \mathcal{A}$ with w^*_{ij} as defined in LP 4.*

Definition 3.3.3. *Let $h_p(i)$ be the cost of the shortest path from i to d_p in $\mathcal{H}$ with $i \in \mathcal{V}_H$ and d_p the destination vertex of passenger $p \in \mathcal{P}$. We then define $h'_p(a)$ as follows*

$$h'_p(a) = \begin{cases} h_p(\eta(a)), & \text{if } a \in \mathcal{V}_\mathcal{R} \cup \mathcal{V}_\mathcal{S} \\ h_p(a), & \text{if } a \in \mathcal{V}_D \end{cases} \tag{3.3.2}$$

Theorem 3.3.1. *$h'_p(a)$ is an admissible distance approximation for the cost of the shortest path from $a \in \mathcal{V}\backslash\mathcal{V}_O$ to the destination vertex d_p of passenger $p \in \mathcal{P}$ in graph $\mathcal{G}$ with pricing cost c'.*

Proof. For $h'_p(a)$ to be an admissible distance approximation for the pricing cost c' of the shortest path from $a \in \mathcal{V}\backslash\mathcal{V}_O$ to the destination vertex d_p, $p \in \mathcal{P}$ in $\mathcal{G}$, we have to show that there cannot exist a path $\sigma = [x_1, x_2, ..., x_{n-1}, x_n]$ with $x_1 = a$ and $x_n = d_p$ in $\mathcal{G}$, for which it holds that

$$\sum_{i=1}^{n-1} c'_{x_i x_{i+1}} < h'_p(a) \tag{3.3.3}$$

Case 1: Let $a \in \mathcal{V}_D$. If $a = d_p$ then it holds that $h'_p(a) = h_p(a) = h_p(d_p) = 0$. If $a \neq d_p$ then there exist no outgoing arcs from a in $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ and thus no outgoing arcs from a in $\mathcal{H}$. Accordingly, there exists no path from a to d_p in $\mathcal{G}$, which implies that the pricing cost of the shortest path is ∞. By construction of $\mathcal{H}$ there also does not exist a path from a to d_p in $\mathcal{H}$ which implies $h'_p(a) = \infty$.

Case 2: Let $a \in \mathcal{V}_\mathcal{R} \cup \mathcal{V}_\mathcal{S}$. Furthermore, let $\sigma = [x_1, x_2, ..., x_{n-1}, x_n]$ with $x_1 = a$ and $x_n = d_p$ be the shortest path from a to d_p in $\mathcal{G}$ with pricing cost c'. By construction of $\mathcal{H}$ there (i) exists a path $\sigma = [\eta(x_1), \eta(x_2), ..., \eta(x_{n-1}), \eta(x_n)]$ in $\mathcal{H}$. Because (ii) $c^{\mathcal{H}}_{\eta(x_i)\eta(x_{i+1})} = \min\{c_{e,f} | (e, f) \in \mathcal{A}_{\eta(x_i)\eta(x_{i+1}}\}$ for all $i \in [1, ..., n]$ and (iii) $w^*_{ef} \leq 0$ for all $(e, f) \in \mathcal{A}$, it holds that $c'_{x_i x_{i+1}} \overset{(iii)}{\geq} c_{x_i x_{i+1}} \overset{(ii)}{\geq} c^{\mathcal{H}}_{\eta(x_i)\eta(x_{i+1})}$ for all $i \in [1, 2, ..., n]$. Thus,

$$\sum_{i=1}^{n-1} c'_{x_i x_{i+1}} \overset{(iii)}{\geq} \sum_{i=1}^{n-1} c_{x_i x_{i+1}} \overset{(ii)}{\geq} \sum_{i=1}^{n-1} c^{\mathcal{H}}_{\eta(x_i)\eta(x_{i+1})} \overset{(i)}{\geq} h'_p(a) \tag{3.3.4}$$

□

Remark: Vertices $a \in \mathcal{V}_O$ have no ingoing arcs, which implies that the A^* algorithm cannot expand such a node except as the origin of a shortest path. Accordingly, we do not need a distance approximation for $a \in \mathcal{V}_O$ to apply the A^* algorithm in the pricing problem of every passenger. Thus, we do not consider this case in the distance approximation.

Since distance approximation h'_p is admissible, optimality of the shortest path solution holds when we apply the A^* algorithm to the pricing problem of passenger $p \in \mathcal{P}$ with distance approximation h'_p. A straightforward approach to calculate the distance approximation is to initiate the Dijkstra algorithm from every vertex in $\mathcal{H}$, with all destination nodes in the target set. We can achieve a speedup in the computation time of the distance approximation if we preprocess a major part of the distance approximation calculations. To do so, let $A_Q = \{(q, j) | j \in V_D, q \in \mathcal{N}^-(j)\}$, where $\mathcal{N}^-(j)$ is the ingoing neighbourhood of $j \in V_D$ in graph $\mathcal{H}$ and $\mathcal{H}' = (V_H \setminus V_D, A_H \setminus A_Q)$ is a new graph, which does not contain the destination vertices. We can then calculate the shortest path $\sigma(i, q)$ between all vertex pairs in $\mathcal{H}'$ via the Dijkstra algorithm. This calculation of the all pairs shortest path problem in $\mathcal{H}'$ is equal for all instances on the same underlying transportation network and can thus be preprocessed. When we solve a new instance, we only have to find a path of minimum length $min\{\sigma(i, q) + c_{qj}) | i \in \mathcal{S}, j \in V_D, q \in \mathcal{N}^-(j)\}$ in $\mathcal{H}$ from all vertices $i \in \mathcal{S}$ to all destination vertices $j \in V_D$.

3.3.3 Integrality

Up to this point our CG algorithm does not yield integer solutions for IP 2. To analyse large transportation system's optima and thus the potential of different transportation systems, finding fractional solutions often suffices. Even though integer solutions are not the main focus of our work, we can obtain integer solutions of good quality through a CG-based P&B approach. In this approach, we apply the CG algorithm (Algorithm 1) once to find an optimal continuous solution and solve the resulting program with the new columns found via CG as an integer program (IP) with a standard IP solver. Note that we can now utilize our pricing filter (Algorithm 2) and the A^*-based pricing algorithm in Algorithm 1. This CG-based P&B approach does not guarantee optimal integer solutions because the feasible region of the resulting IP is a subset of the feasible region of the original master problem after Algorithm 1 terminates. Nevertheless, our CG-based P&B approach allows us to find integer solutions of good quality quickly. From such an integer solution we can then construct a path for every passenger $p \in \mathcal{P}$ - if one exists - (see Section 3.3.1.4) in which every path fulfills all the constraints defined in Section 3.2.

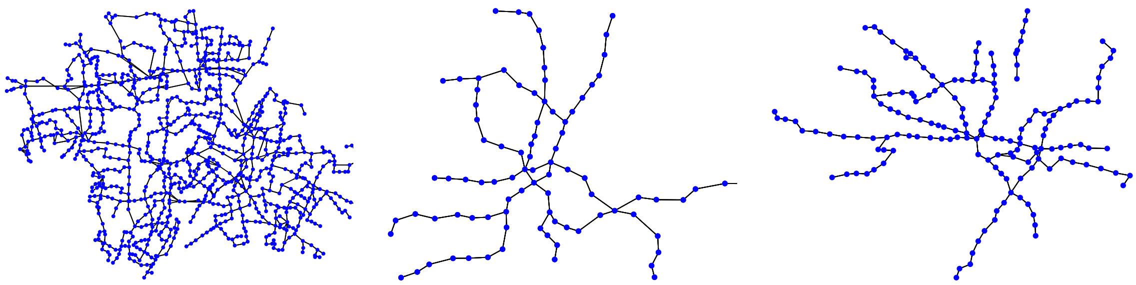
Note that generally, the main drawback of a P&B approach is, that the resulting IP can be infeasible (Sadykov et al. 2019). To mitigate this drawback, we ensure that our CG-based P&B approach always finds an integer solution because the variables used for initiating the CG algorithm are still contained in the IP and give a trivial integer solution.

3.4 Case Study

Our case study bases on a real-world setting for the city of Munich, Germany. We derive the public transport network consisting of bus, subway, and tram lines (see Figure 7), together with its schedules from GTFS data (GTFS 2022). The bus network consists of 986 vertices and 2,228 arcs, while the subway network consists of 89 vertices and 184 arcs, and the tram network consists of 163 vertices and 338 arcs. We received information about vehicle capacity by the public transportation provider MVG (MVG 2022). Here, busses have a capacity of 60, subways a capacity of 940 and trams a capacity of 215. We obtained travel demand from the modeling tool MITO (cf. Moeckel et al. 2019), which generates passenger data through a Monte-Carlo sampling followed by a nested mode choice model. We focus on three passenger datasets. The SUBWAY dataset consists of passengers only using the subway network with 2,206 trips between 7-9am, the BUS dataset consists of passengers only using the bus network with 26,320 trips between 7-9am and the BUS-SUBWAY-TRAM dataset consists of passengers using all possible intermodal combinations of the bus, subway, and tram network with 62,550 trips between 7-9am.

We use this case study to analyse the effectiveness of i) our CG approach, ii) our pricing filter, iii) our A^*-based pricing algorithm, and iv) the quality of our integer solution, obtained by our CG-based P&B approach. For analysis i) we generate instances with 132-662 passengers (6-30%) from the SUBWAY dataset. For each passenger set size, we run 10 random instances, in which we scale the capacity of the vehicles according to the subset size, i.e., in an instance, with 10% of

Figure 7: Transportation networks of our Munich case study



(a) Bus network **(b) Subway network** **(c) Tram network**

all passengers of the dataset, only 10% of the vehicles' capacity is available. For analysis ii) - iv), we generate instances with 2,632-23,688 passengers (10-90%) for the BUS dataset and 6,255-56,295 (10-90%) passengers for the BUS-SUBWAY-TRAM dataset. For each passenger set size, we again run 10 random instances, in which we scale the capacity of the vehicles according to the subset size. Note that we hold out 10% of the passenger data during the instance generation to be able to generate at minimum 10 different instance for each instance size.

3.5 Results

In Sections 3.5.1 - 3.5.4 we show the effectiveness of i) our CG approach, ii) our pricing filter, iii) our A^*-based pricing algorithm, and iv) the quality of our integer solutions based on the experiments described in Section 3.4. We set the maximum runtime of the algorithm to 60 minutes. All our experiments have been conducted on a standard desktop computer equipped with an `Intel(R) Core(TM) i9-9900, 3.1 GHz CPU` and `16 GB` of `RAM`, running `Ubuntu 20.04`. We have implemented the CG algorithm in `Python (3.8.11)` using `Gurobi 9.5` to solve the restricted master problem. Our source code as well as an overview of all results can be found on https://github.com/tumBAIS/intermodalTransportationNetworksCG. Additionally, Section 3.5.5 showcases how to apply our algorithmic framework to analyze passenger utilization of individual routes based on the calculated passenger flows.

3.5.1 Effectiveness of the column generation

Figure 8 shows the computation time of our CG-based P&B algorithm and IP 1 solved with a standard IP solver. Here, the CG-based P&B algorithm does not utilize our pricing filter and solves the pricing problems with Dijkstra's algorithm. We see that our CG-based P&B algorithm vastly outperforms a standard IP formulation even without any additional enhancements. For the IP formulation, we run out of memory for all instances with more than 662 passengers. Table 2 complements this observation by showing the number of solved instances, construction time of the IP/RMP, the solving time, and the total time for instances with 132 - 662 passengers of the

Figure 8: Total time comparison P&B vs. IP 1 - SUBWAY

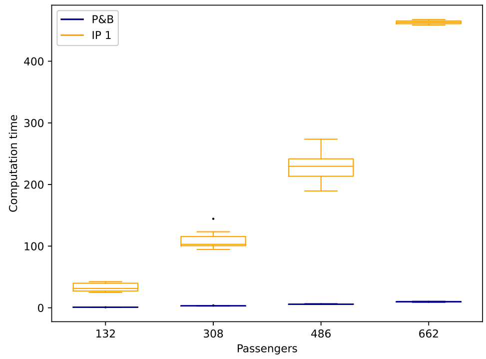


Table 2: Algorithm comparison IP 1 (IP) and Algorithm 1 without A^* and filter (CG) - SUBWAY

Number of passengers	**132**		**308**		**486**		**662**	
Algorithm	CG	IP	CG	IP	CG	IP	CG	IP
# Solved instances (out of 10)	10	10	10	10	10	10	10	2
Median IP/RMP construction time	<1s	48s	<1s	142s	<1s	315s	<1s	553s
Mean solving time	1s	33s	3s	109s	6s	228s	10s	463s
Mean total time	1s	82s	3s	252s	6s	542s	10s	1016s

SUBWAY dataset. Here, the IP/RMP construction time is the time needed to formulate IP 1 or the RMP from Algorithm 1 respectively. The CG-based P&B algorithm needs less than one second to find an integer solution while the IP formulation takes more than 9 minutes in the biggest subset size. The solving time in the CG-based P&B algorithm is also much lower than in the IP formulation. With only 662 passengers, the standard IP formulation is not solvable for eight out of ten instances, because we run into memory bounds.

3.5.2 Effectiveness of the pricing filter

Table 3 shows the number of passengers, solved instances, computation time, and number of solved pricing problems for the subset of all travel requests ranging from 2,632-23,688 passengers of the BUS dataset, computed with our CG-based P&B algorithm utilizing and not utilizing our pricing filter, using the A^*-based pricing algorithm to solve each pricing problem. All instances are solved to optimality with and without our pricing filter. However, when we apply our pricing filter the number of solved pricing problems decreases significantly by 60-65%, which leads to a 35-50% reduction in computation time. Figure 9 and Figure 10 show the number of solved pricing problems and the computation time for the different instance sizes respectively. We see that our pricing filter leads to stable numbers of pricing problems and computation times.

Table 4 extends our analysis to the BUS-SUBWAY-TRAM dataset. Here, 6,255 trips can be solved in one minute and 56,295 trips take 48 minutes to solve when using our pricing filter. For 43,785

Table 3: Algorithm comparison with and without the filter using A^* - BUS dataset

Number of passengers	**2,632**		**7,896**		**13,160**		**18,424**		**23,688**	
Pricing filter (On/Off)	On	Off	On	Off	On	Off	On	Off	On	Off
# Solved instances (out of 10)	10	10	10	10	10	10	10	10	10	10
Mean computation time	33s	51s	107s	192s	245s	420s	382s	737s	603s	1129s
Mean # pricing problems	9,606	24,491	23,450	62,405	34,956	97,418	46,143	132,699	59,372	165,887

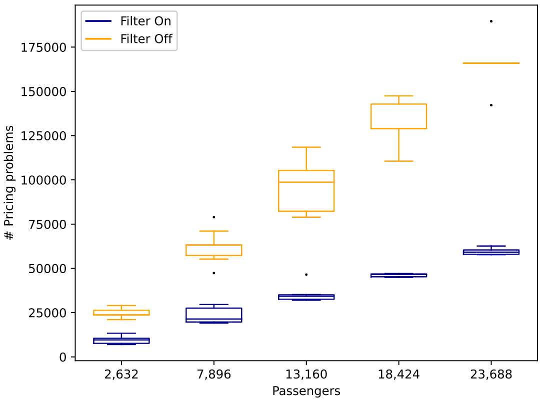


Figure 9: Number of pricing problems (BUS)

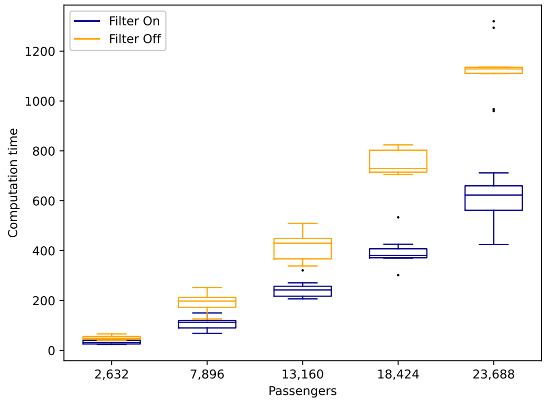


Figure 10: Computation time (BUS)

Table 4: Algorithm comparison with and without the filter using A^* - BUS-SUBWAY-TRAM dataset

Number of passengers	**6,255**		**18,765**		**31,275**		**43,785**		**56,295**	
Pricing filter (On/Off)	On	Off	On	Off	On	Off	On	Off	On	Off
# Solved instances (out of 10)	10	10	10	9	10	8	10	0	10	0
Mean computation time	66s	142s	370s	891s	900s	2243s	1766s	-	2856s	-
Mean # pricing problems	13,810	36,273	40,157	108,816	66,696	178,240	93,469	-	119,233	-

passengers and beyond, no instance can be solved without using the pricing filter. In the instances that can be solved without our pricing filter, a computational speedup of up to 60% can be achieved by applying our pricing filter. The reduction of 60% in the number of solved pricing problems is similar to the reduction in the instances of the BUS dataset. Figure 11 and Figure 12 again show the number of solved pricing problems and the computation time for the different instance sizes respectively. We see that similar to the BUS dataset our pricing filter leads to a stable number of pricing problems and to stable computation times. Concluding, our pricing filter leads to improved stable performances across different instance sizes as well as single and intermodal graph structures.

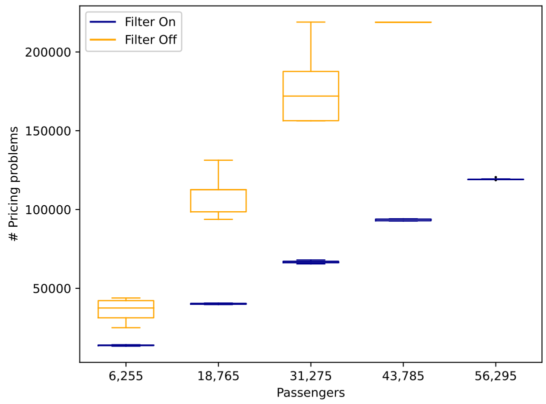


Figure 11: Number of pricing problems (BUS-SUBWAY-TRAM)

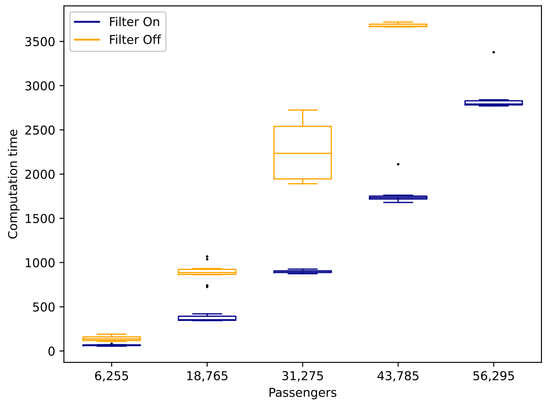


Figure 12: Computation time (BUS-SUBWAY-TRAM)

3.5.3 Effectiveness of the A^*based pricing algorithm

Table 5 shows the number of passengers, solved instances, and the computation time of our CG-based P&B algorithm with the A^*-based pricing algorithm and the Dijkstra-based pricing algorithm for the BUS dataset, utilizing our pricing filter in both algorithms. We see that the integration of the A^*-based pricing algorithm yields a computational speedup of up to 85% in the BUS dataset. No instance with 23,688 passengers and only eight instances with 18,424 passengers can be solved within 60 minutes with the Dijkstra-based pricing algorithm, where on the other hand, all instances can be solved with the A^*-based pricing algorithm.

Table 6 extends our analysis to the BUS-SUBWAY-TRAM dataset. For this dataset, we observe a similar speedup of the computation time as in the BUS dataset for smaller instances with 6,255 instances. Here, the A^*-based pricing algorithm leads to 90% faster computation times than the Dijkstra-based pricing algorithm. In contrast to the BUS dataset, bigger instances with 18,765 passengers or more cannot be solved with the Dijkstra-based pricing algorithm, while all instances can be solved with the A^*-based pricing algorithm. Here, the graph constructed for the intermodal BUS-SUBWAY-TRAM transportation network is much bigger than the graph constructed for the BUS transportation network. Without the additional information provided through the distance approximation, the Dijkstra algorithm has to expand much more nodes to find the shortest path for a pricing problem. The bigger the underlying graph, the greater the difference between the number of the explored nodes between the Dijkstra-based and the A^*-based pricing algorithm gets. Figure 13 and Figure 14 complement this analysis and show that our algorithm with the A^*-based pricing algorithm leads to more stable computation times in both datasets compared to the Dijkstra-based pricing algorithm.

We note that the distance approximation needed for the A^*-based pricing algorithm was calculated in under 17 seconds for all BUS instances and in under 45 seconds for all BUS-SUBWAY-TRAM instances. The significant improvement in computation time in the CG justifies this additional setup time.

Table 5: Algorithm comparison A^* vs. Dijkstra with the filter active - BUS

Number of passengers	**2,632**		**7,896**		**13,160**		**18,424**		**23,688**	
Pricing problem solver	A^*	Dijkstra	A^*	Dijkstra	A^*	Dijkstra	A^*	Dijkstra	A^*	Dijkstra
# Solved instances (out of 10)	10	10	10	10	10	10	10	8	10	0
Mean computation time	33s	239s	107s	914s	245s	1932s	382s	3022s	603s	-

Table 6: Algorithm comparison A^* vs. Dijkstra with the filter active - BUS-SUBWAY-TRAM

Number of passengers	**6,255**		**18,765**		**31,275**		**43,785**		**56,295**	
Pricing problem solver	A^*	Dijkstra	A^*	Dijkstra	A^*	Dijkstra	A^*	Dijkstra	A^*	Dijkstra
# Solved instances (out of 10)	10	10	10	0	10	0	10	0	10	0
Mean computation time	66s	793s	370s	-	900s	-	1765s	-	2856s	-

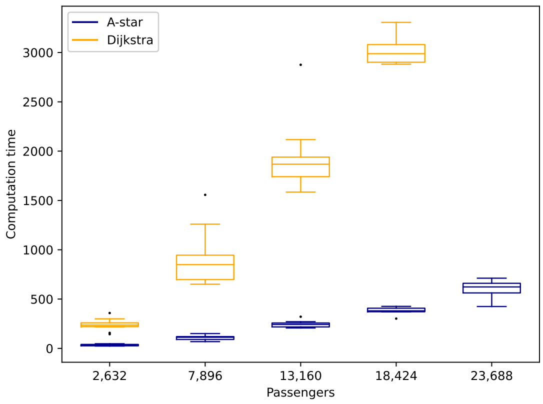


Figure 13: Computation time A^* vs. Dijkstra (BUS)

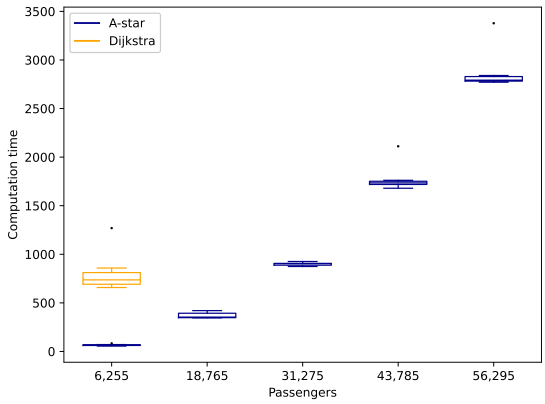


Figure 14: Computation time A^* vs. Dijkstra (BUS-SUBWAY-TRAM)

3.5.4 Integral solution quality

Since finding integer solutions is not the main focus of this work, we used a standard IP solver to solve the IP introduced in our CG-based P&B approach in Section 3.3.3. Table 7 shows the number of solved instances, the total computation time, the computation time needed to find our integer solutions, and the optimality gap for instances of the BUS and BUS-SUBWAY-TRAM datasets, utilizing our CG-based P&B algorithm including our pricing filter and the A^*-based pricing algorithm. Although our CG-based P&B approach does not guarantee optimality for integer solutions, we are able to find optimal integer solutions for all instances in under five seconds. Here, we use the continuous solution of the CG algorithm as a lower bound. Furthermore, we see that finding integer solutions accounts only for a small percentage of the total computation time. Concluding, our CG-based P&B approach leads to stable performances across different instance sizes as well as uni and intermodal transportation networks.

Table 7: Integral solutions

	BUS					BUS-SUBWAY-TRAM				
Number of passengers	**2,632**	**7,896**	**13,160**	**18,424**	**23,688**	**6,255**	**18,765**	**31,275**	**43,785**	**56,295**
# Solved instances (out of 10)	10	10	10	10	10	10	10	10	10	10
Mean total computation time	33s	124s	245s	392s	639s	66s	370s	900s	1766s	2856s
Mean integer time	1.3s	1.7s	2.0s	2.1s	2.7s	1.8s	4.0s	5.0s	3.9s	4.6s
Mean optimality gap	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

3.5.5 Practical insights

Our algorithmic framework allows us to analyze the passenger utilization of individual routes based on the calculated passenger flows. In this context, we can study multiple metrics such as walking-, waiting-, and in-vehicle-times. In this section, we showcase such a system analysis in which we briefly present differences in line utilization between three scenarios: a basecase, a removed-arcs, and a removed-routes scenario. In the basecase scenario, we analyze the passenger flow system optimum in the Munich transportation system between 6-8 am for 80% of passengers for the subway, bus, and tram networks. In the removed-arcs scenario we remove all subway route arcs between Giselastraße-Universität-Odeonsplatz-Marienplatz to see how e.g., construction work or medical emergencies influence passenger flows. In the removed-routes case, we analyze a scenario where we remove the subway line U5 from the transportation network. There are 37,897 trips in all three scenarios, with a maximum trip time of 90 minutes.

A PTS provider can utilize such an approach to evaluate different short-term network design decisions, e.g., which line or mode to activate or deactivate during system disruptions such as driver or vehicle shortage, construction work, and medical emergencies. Showcasing such an analysis to underline the practical relevance of our algorithmic framework remains the scope of this section.

Figure 15 shows a similar walking-, waiting-, and in-vehicle-time distribution for all three scenarios. This is because the Munich public transportation network is very dense and in such capable to compensate for small perturbations in the system.

Figure 16 shows how the perturbed passenger flows influence the utilization of different routes in the remove-arcs scenario. We see that for the analyzed subway route, the maximum utilization rises from 80 to 130 (see Figure 16a). Figure 16b shows an increase in utilization from under 25 to more than 100 on the analyzed tram route.

Figure 17 shows similar information for the remove-route scenario. We see that for this analyzed subway route, the maximum utilization at the same time again rises from 100 to more than 200 (see Figure 17a). It also occurs, that even though we remove a subway line from the network, the utilization of some routes drops significantly. This happens when we assign passengers flow to multiple routes where the removal of one route makes another route superfluous, e.g., taking a tram to a closed subway stop. Figure 17b shows a decrease of maximal utilization from more than 35 to under 15 on the analyzed tram route.

Figure 15: Distribution of walking-, waiting-, and in-vehicle-times

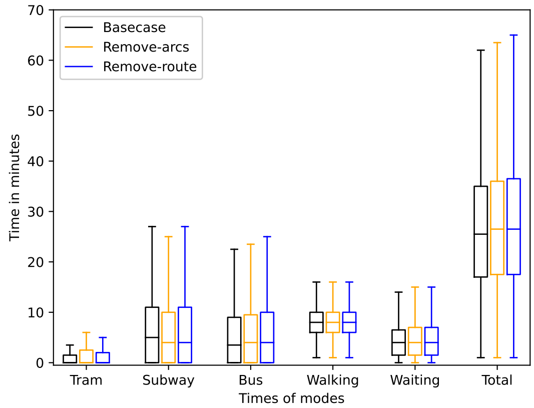
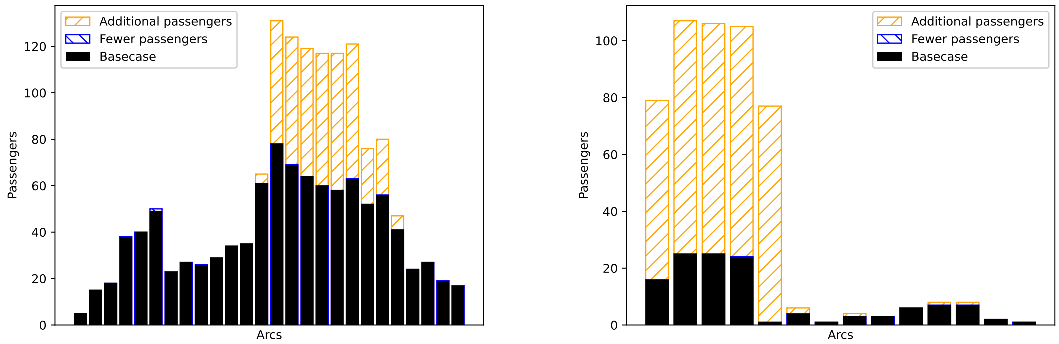


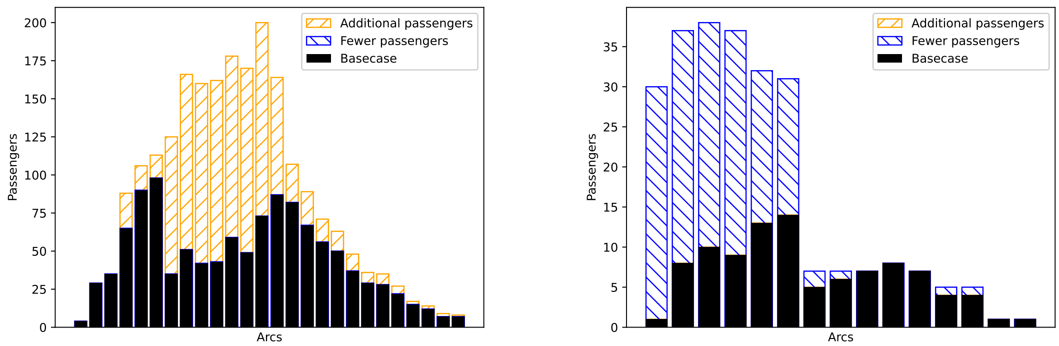
Figure 16: Differences in route utilization between basecase and remove-arc scenarios for selected routes



(a) Subway route

(b) Tram route

Figure 17: Differences in route utilization between basecase and remove-route scenarios for selected routes



(a) Subway route

(b) Tram route

As can be seen in Figures 16 and 17, route changes (modeled by removing a route) or route interruptions (modeled by removing specific arcs) can lead to significant changes in a PTS's throughput. Analyzing these changes is of interest for practitioners to take informed decisions on line closures as well as emergency or construction work related interruptions. In this context, the algorithmic framework presented in this paper provides a computational efficient tool with which a service provider can perform extensive analyses of the PTS, study different route candidates, and search for interdependencies between different routes in a short amount of time.

3.6 Conclusion

With this work, we introduced an algorithmic framework to determine the passenger flow system optimum of an intermodal transportation system. To do so, we modeled a transportation system as a partially time-expanded digraph, which implicitly encodes problem specific constraints related to the intermodality of the problem. This allowed us to find the intermodal transportation system's passenger flow optimum by solving a standard integer minimum-cost multi-commodity flow problem. We solved this problem with a column generation based price-and-branch approach, for which we introduced a pricing filter to reduce the number of solved pricing problems. Furthermore, we proposed a distance approximation to utilize the A^* algorithm to solve the pricing problems more efficiently. We applied our methodology to a real-world case study for the city of Munich. Our results show that our pricing filter reduces the computation time of our algorithm by up to 60% and the application of the A^*-based pricing algorithm additionally reduces it by up to 90%. Our algorithm is able to solve unimodal instances with 23,688 trips in 11 minutes and intermodal instances with 56,295 trips in 48 minutes, whereas straightforward MIP models fail to solve instances with more than a few hundred trips. Our algorithmic framework allows us to analyze the passenger utilization of individual routes based on the calculated passenger flows which is helpful for short-term network design decisions.

Even though our price-and-branch framework finds optimal solutions for all instances of our case study, it does not possess an optimality guarantee for integer solutions. To this end, extending our framework to a branch-and-price framework that guarantees optimality for integer solutions remains an interesting avenue for future research. Moreover, our current problem setting does not consider additional path constraints, e.g., limits on passenger transfers, which remains another interesting direction for future research.

4 Branch-and-Price for the Stochastic TSP with Generalized Latency

We consider an extension of the symmetric traveling salesman problem with generalized latency (TSP-GL) that explicitly models uncertainty. The Stochastic TSP-GL (STSP-GL) aims to choose a subset of nodes of an undirected graph and determines a Hamiltonian tour amongst those nodes, minimizing an objective function that is a weighted combination of route design and passenger routing costs. These nodes are selected to ensure that a predefined percentage of uncertain passenger demand is served with a given probability. We formulate the STSP-GL as a stochastic program and propose a branch-and-price algorithm for solving its deterministic equivalent. We also develop a local search approach with which we improve the performance of the B&P approach. We assess the efficiency of the proposed methods on a set of instances from the literature. We demonstrate that the proposed methods outperform a known benchmark, improving upper bounds by up to 85% and lower bounds by up to 55%. Finally, we show that solutions of the stochastic model are both more cost-effective and robust than those of the deterministic model.

This chapter is based on an article published as: Lienkamp, B., Hewitt, M., & Schiffer, M. (2023). Branch and Price for the Stochastic Traveling Salesman Problem with Generalized Latency. Transportation Science.

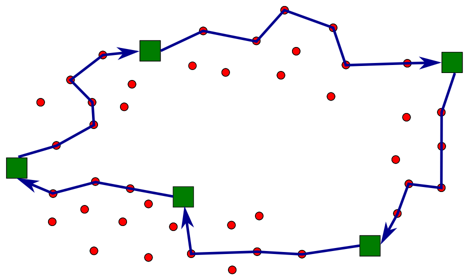
4.1. Introduction

As urban areas experience rapid population growth, the importance of public transportation (PT) becomes increasingly evident. In light of expanding cities, challenges such as traffic congestion, environmental pollution, and limited resources become more pressing. Within this context, PT takes on a crucial role as it offers a viable solution to tackle these challenges by presenting an alternative to private vehicle usage. This alternative helps to alleviate traffic congestion and to reduce pollution. Moreover, PT ensures accessibility for individuals across diverse socioeconomic backgrounds, connecting them to vital services, education, and employment opportunities. Additionally, its ability to optimize land use and curb urban sprawl promotes compact, well-connected communities while enhancing social cohesion. By establishing and strengthening comprehensive PT networks, cities can attain improved organizational efficiency, heightened environmental sustainability, and greater societal inclusivity. In this context, PT networks can contain different services.

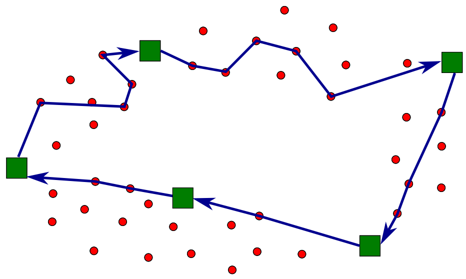
Line-based bus lines serve as a cost-effective and reliable backbone of PT networks, offering widespread coverage that connects diverse communities. Their high passenger capacity and established routes promote social inclusivity and accessibility to essential services. Moreover, they enable efficient utilization of existing infrastructure, allowing to transport a larger number of people using fewer vehicles, which can contribute to reduced traffic congestion and emissions. However, they can be susceptible to traffic congestion and limited flexibility in adapting to changing travel demands, leading to potential delays and less personalized service experiences for passengers.

Demand Responsive Systems (DRS) offer more flexibility and can adapt to fluctuating individual transportation demands, thus enabling more personalized services while maintaining a degree of resource sharing. DRS were first introduced under the name of Dial-a-Ride (DAR) as a door-to-door service for users with reduced mobility (Wilson et al. 1971; Ioachim et al. 1995; Toth and Vigo 1996). One drawback of DAR are the higher operational costs associated with providing individualized service and accommodating specific travel requests. Additionally, the real-time nature of pickups and drop-offs, i.e., significant route changes during operation, can lead to unpredictable travel times for passengers, which might not be suitable for those with time-sensitive commitments.

Demand Adaptive Systems (DAS) (Malucelli et al. 1999; Quadrifoglio et al. 2007) incorporate DRS into scheduled bus transportation: a DAS bus line provides a line-based transit-line service for a set of compulsory stops, which are bound to a schedule with fixed time windows Passengers can issue requests at non-compulsory stops that are optional from an operator perspective and can be served via detours in between compulsory stops (see Figure 1). Accordingly, a DAS combines the regularity and predictability of line-based PT, which enables seamless integration with other modes of transportation and services that do not require advanced booking, with the



(a) Route variant 1



(b) Route variant 2

Figure 1 **A DAS line with compulsory stops (green) and optional stops (red).**

flexibility of a DRS. Ultimately, a DAS combines the advantages of line-based PT and DRS which can increase efficiency, utilization, and service level of the respective PT network.

Planning the operations of DAS is a highly complex task that requires balancing the characteristics of line-based and on-demand systems and involves both a service design phase, where routes and schedules are established and an operational phase, with adjustments to vehicle routes and schedules in response to user requests. The complexity in planning DAS operations has been extensively studied in the literature, with a comprehensive review of methodological aspects presented in Errico et al. (2013). Focusing on the aforementioned service design phase, the tactical planning of a DAS line encompasses three main interrelated tasks: (1) identifying and selecting compulsory stops; (2) determining a general visit order of transit stops within the service area; and (3) establishing schedules for compulsory stops. In this paper, we focus on task (2), and, unlike existing literature, we explicitly model uncertainty in passenger demands, which allows us to determine a general order of transit stops that is robust against volatile passenger demand.

For given compulsory stops, Task (2) can be modeled as a symmetric traveling salesman problem with generalized latency. While we review related literature in detail in Section 2, we note that this problem has been studied in Errico et al. (2017), who propose a Benders-branch-and-cut (BBC) algorithm to find a route between all compulsory and optional transit stops, minimizing passenger travel times and operating costs in a deterministic setting. Our work builds on this methodological foundation and focuses on integrating stochastic passenger demand to account for uncertainty during the tactical planning of DAS. Additionally, we do not assume all demand must be met but instead seek to meet a service level. These extensions allow DAS operators to analyze the impact of different service levels and the frequency of meeting them on the route design and served passenger demand. Furthermore, accounting for uncertainty in passenger demand allows for the design of DAS routes that are robust against volatile passenger demand.

Contribution: We propose an algorithmic framework to determine a Hamiltonian tour on a subset of nodes of an undirected graph, which minimizes a linear combination of route design and passenger routing costs. The subset of nodes includes a set of compulsory nodes as well as other non-compulsory nodes, which have to be selected such that serving the passengers requiring transportation between nodes in this set meets a percentage of the uncertain passenger demand with a given probability. Accordingly, our problem consists of selecting a suitable subset of nodes and determining a Hamiltonian tour on it. Specifically, we formulate the so-called stochastic TSP with generalized latency (STSP-GL) as a chance-constrained program and determine its deterministic equivalent. To solve this deterministic equivalent efficiently, we present a branch-and-price (B&P) approach, which allows us to decompose the STSP-GL into multiple TSP-GLs. Additionally, we develop a local search-based heuristic and a hybrid solution approach. The hybrid solution approach utilizes our heuristic approach in our B&P approach. We show the efficiency of determining good bounds through our approaches by applying them to the TSPGL_LIB instance set and show that our three approaches, i.e., B&P, heuristic, and hybrid, meet the MIP benchmark's bounds on small instances and find much better bounds on larger instances. Furthermore, we show that our three approaches find good upper bounds much faster than our MIP benchmark. Additionally, we show that when we compare our stochastic approach to a deterministic approach, as expected, the deterministic approach's solutions have lower design costs but do not meet the necessary feasibility frequencies, i.e., they are not feasible stochastic solutions. Finally, we show that the service levels influence the design costs much more than the feasibility frequencies. In summary, our work allows the integration of stochastic demand into the TSP-GL, which grants us more robust routes compared to the deterministic demand setting.

Organization: The remainder of this paper is as follows. Section 4.2 reviews related literature. We specify our problem setting in Section 4.3 and develop our methodology in Section 4.4. In Section 4.5, we describe a case study based on the TSPGL_LIB dataset, which we use to present numerical results that show the efficiency of our algorithmic framework. Section 4.6 concludes this paper by summarizing its main findings.

4.2. Literature Review

We define the STSP-GL as a generalization of the TSP-GL in which only a subset of all stochastic passenger demand can be served as long as a service level that measures the expected sum of all transported demand is met with a certain probability. Like the TSP-GL, the STSP-GL is closely related to the Fixed-Charge Network Design Problem (FNDP), which involves finding a set of arcs in a graph that enables the flow of commodities from their origins to their destinations in order to satisfy some demand characteristics. The FNDP minimizes routing costs, i.e., variable expenses incurred when sending flows through the network, or design costs, i.e., fixed expenses associated with establishing network infrastructure, or both (Costa 2005). In the STSP-GL, all arcs of the graph have infinite capacity, and the set of chosen arcs has to form a tour that visits all compulsory stops. Additionally, this set of arcs has to allow passenger flow for a subset of passengers to meet a predefined service level, i.e., the expected sum of served passenger demand has to be met with a certain probability. Several exact and heuristic methods have been proposed to solve the FNDP. The most prominent exact methods to solve FNDP are Lagrangian-based techniques, Benders decomposition, and polyhedral approaches. For an extensive overview of exact methods for FNDP, we refer to Gendron et al. (1999), Gendron (2011), and Crainic et al. (2021).

Lagrangian-based techniques utilize either subgradient algorithms or bundle methods to solve the Lagrangian dual. Crainic et al. (2001) show that for Lagrangian-based techniques, bundle methods appear superior to subgradient approaches as they converge faster and are more robust relative to different relaxations and problem characteristics. Frangioni and Gorgone (2014) exploit the structure of the Dantzig-Wolfe reformulation of the Lagrangian dual to construct an efficient bundle method. Frangioni et al. (2017) analyzed a large class of subgradient methods to show that the theory developed in d'Antonio and Frangioni (2009) impacts the performance of subgradient methods. Gendron (2019) compares Lagrangian relaxation bounds for a large class of network design problems, which they obtain either by relaxing the linking constraints or the flow conservation constraints.

Costa et al. (2009) give an extensive overview on the application of Benders decomposition to FNDP and clarify the relationships between Benders, metric, and cutset inequalities, which are often used in solution algorithms for multi-commodity capacitated FNDP. Costa et al. (2012) propose a general scheme for generating extra cuts in a Benders decomposition, which they apply to the FNDP.

There exists a multitude of polyhedral approaches to solve the FNDP. Wolsey (2003) gives an overview on strong valid inequalities and tight extended formulations for general mixed integer linear programs. Chouman et al. (2017) develop a cutting-plane algorithm for the multi-commodity capacitated FNDP. Atamtürk (2002) analyzes the cut-set polyhedra of capacitated FNDP and shows that utilizing cut-set inequalities significantly improves computations. Atamtürk (2001) applies flow pack inequalities, which improve computations for capacitated fixed-charge network flow problems. Padberg et al. (1985) study the single-node fixed-charge flow problem and introduce flow cover inequalities.

The most prominent heuristic approaches to solve FNDP include classical heuristics such as local search (Walker 1976), neighborhood-based metaheuristics such as tabu search (Gendreau and Potvin 2019), simulated annealing (Delahaye et al. 2019), and iterated local search (Lourenço et al. 2019), population-based metaheuristics such as genetic algorithms (Whitley 2019), and parallel metaheuristics (Crainic 2019). Furthermore, several matheuristics have been applied to FNDP.

There exist several approaches to solve FNDP via tabu search. Sun et al. (1998) apply tabu search with two strategies for each of the intermediate and long-term memory processes to the FNDP. Crainic et al. (2000) utilize a tabu search framework that combines pivot moves with column generation to solve multicommodity capacitated FNDP. Ghamlouche et al. (2003) propose cycle-based neighborhood structures for multicommodity-capacitated FNDP and show their efficiency through a simple tabu search method.

For neighborhood-based metaheuristics, Lotfi and Tavakkoli-Moghaddam (2013) introduce a priority-based encoding genetic algorithm for linear and nonlinear FNDP and show that it outperforms a spanning tree-based genetic algorithm in terms of solution quality and computation time. Ghamlouche et al. (2004) combine path relinking with cycle-based tabu search in an algorithm that is robust regarding solution quality and computational effort. Crainic and Gendreau (2007) propose a scatter search approach for the capacitated FNDP, which cannot match the path relinking algorithm consistently but can produce better solutions on some of the test instances.

Several matheuristics have been applied to FNDP. Rodríguez-Martín and Salazar-González (2010) utilize local branching (Fischetti and Lodi 2003) to solve the multicommodity capacitated FNDP. They show that their algorithm outperforms the cycle-based tabu search (Ghamlouche et al. 2003) and path relinking (Ghamlouche et al. 2004).

Hewitt et al. (2010) combine mathematical programming algorithms and heuristic search techniques to find provably high-quality solutions for capacitated FNDP quickly. They show that their algorithm outperforms cycle-based tabu search (Ghamlouche et al. 2003) and path relinking (Ghamlouche et al. 2004) as well.

Katayama and Yurimoto (2011) combine capacity scaling using a column-row generation technique and local branching. They show that their approach outperforms cycle-based tabu search (Ghamlouche et al. 2003), path relinking (Ghamlouche et al. 2004), local branching (Rodríguez-Martín and Salazar-González 2010), and IP Search (Hewitt et al. 2010).

Yaghini et al. (2015) propose a cutting-plane neighborhood structure, which they implement in a tabu search framework and combine with local branching. They show that their approach outperforms cycle-based tabu search (Ghamlouche et al. 2003), path relinking (Ghamlouche et al. 2004), local branching (Rodríguez-Martín and Salazar-González 2010), IP Search (Hewitt et al. 2010) and CALB (Katayama and Yurimoto 2011). We refer to Crainic et al. (2021) for an extensive overview of all exact and heuristic approaches to solve the FNDP.

Besides FNDP, the STSP-GL relates to three additional problems, the generalized minimum latency problem (GMLP), the TSP-GL, and the selective general minimum latency problem (SGMLP). Errico (2008) introduces the GMLP and the SGMLP. The GMLP considers a complete directed graph $G = (N, A)$ with associated design and routing cost on its directed arcs as well as deterministic origin-destination demands d_{hk} between its node pairs. The goal is to identify a Hamiltonian directed cycle that minimizes a combination of design costs, i.e., the time needed to travel the tour, and routing costs, i.e., average passenger travel time, of the demand on this cycle. The SGMLP builds on this setting and relaxes the condition of the Hamiltonian tour. Accordingly, the goal of the SGMLP is to find a simple tour on the graph, which does not have to be Hamiltonian, i.e., to serve a subset of deterministic passenger demands such that a given service level is met. The TSP-GL also builds on the methodology of the GMLP with the difference in the cycle design, as the GMLP seeks a directed tour while the TSP-GL seeks an undirected tour (Errico 2008; Errico et al. 2017). This difference allows the GMLP to consider asymmetric design costs but also fixes the flow direction of the passengers. Errico et al. (2021) describe the emergence and interaction of the TSP-GL within the broader context of DAS tactical planning. Through experimental analysis, they emphasize that passenger perceptions of service quality depend greatly on travel times. The STSP-GL combines the TSP-GL and the SGMLP as it seeks an undirected tour that does not have to include all graph nodes. Furthermore, it also assumes stochastic passenger demand instead of deterministic passenger demand, which is the case for GMLP, SGMLP, and TSP-GL. Errico et al. (2017) give an overview on how the TSP-GL and, thus also, the STSP-GL additionally relate to the minimum latency problem.

The GMLP, SGMLP, and TSP-GL have all been studied under the assumption that all information about passenger demand is available before the design decisions, i.e., which edges form the tour, are made. It is a general understanding that, in most cases, this does not translate to real-world applications (Kall and Wallace 1994). Traditionally, this is often not considered explicitly during the tactical planning phase but dealt with in the operational planning phase (Ibarra-Rojas et al. 2015). Accordingly, passenger demand is usually determined through forecasting methods, e.g., average historical data. Usually, the expected quality of a solution to a stochastic model is better than a solution to the deterministic counterpart when evaluated in the stochastic setting. Wallace (2000) and Higle and Wallace (2003) explain that this occurs, as the deterministic model is optimal for one specific scenario but might be very bad for scenarios where it is not optimal. Additionally, the deterministic solution is often feasible but not optimal in the stochastic model. Birge (1982) measures this difference in solution quality by the value of stochastic solution (VSS). Here, the VSS represents the potential benefit of solving the stochastic model instead of the deterministic model in the stochastic environment. Wallace (2000) shows that stochastic models may find solutions significantly different from deterministic models. Nevertheless, Thapalia et al. (2011) and Maggioni and Wallace (2012) show that for multiple other problems, certain structural patterns from the deterministic solutions may re-emerge in the stochastic solutions. We refer to Birge and Louveaux (2011) for an extensive overview on the research area of stochastic programming.

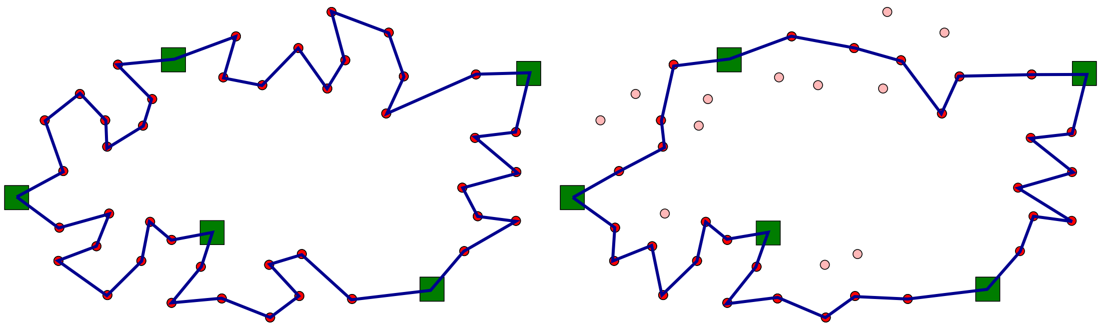
In summary, the STSP-GL combines the deterministic SGMLP and TSP-GL with stochastic demand as for an undirected graph with compulsory nodes, it aims to find a Hamiltonian tour on a subset of nodes, which includes all compulsory nodes, such that a given percentage of passenger demand is met with a given probability.

In recent years, researchers have made significant strides in integrating uncertain passenger demand into optimization models for public transportation systems. Tian et al. (2021) consider the integration of autonomous vehicles into bus transit systems and determine the optimal bus fleet size and its assignment onto multiple bus lines in a bus service network considering uncertain demand. Ma et al. (2021) study the problem of bus regulation with uncertain passenger demand and disturbance for urban rapid transit lines on the basis of a robust model predictive control algorithm. Gkiotsalitis and Alesiani (2019) incorporate travel time and passenger demand uncertainty to determine robust schedules that minimize the possible loss at worst-case scenarios. Li et al. (2019) propose a robust dynamic control framework that considers delay disturbances and passenger demand uncertainty to reduce bus bunching. So far, uncertainty in passenger demand that allows a DAS operator to determine a general order of transit stops that is robust against volatile passenger demand has not been considered explicitly in the tactical planning phase of DAS. Accordingly, DAS operators are unable to analyze the impact of different service levels and the frequency of meeting them on the route design and served passenger demand.

4.3. Problem description and mathematical formulations

In this work, we focus on Task (2) of a DAS line's aforementioned tactical planning tasks, i.e., determining a general order of transit stops. To do so, we build on the work of Errico et al. (2017) by incorporating stochastic passenger demand into our problem setting, i.e., we assume the volume of demand and the presence of demand between two stops to be uncertain. Accordingly, we aim to choose a set of arcs to form the tour of a DAS before the passenger demand is known. We define a *service level* as the percentage of passenger demand transported, which allows us to determine the order of compulsory stops and the service area, i.e., the sets of optional stops that must be covered between consecutive compulsory stops, such that they fulfill a given service level with a specified probability, e.g., 80% of the estimated passenger demand has to be fulfilled with a probability of 90% (see Figure 2).

As a consequence, the STSP-GL involves three levels of decisions. First, we select a subset of transit stops from all possible transit stops such that the expected sum of passengers transported satisfies a specified service level of passenger demand with a predefined probability. This subset of transit stops must include all compulsory stops. Second, we choose a set of edges that form a Hamiltonian circuit on the selected subset of transit stops. Third, we determine the direction of



(a) Solution for deterministic passenger demand (TSP-GL) with all optional stops

(b) Solution for stochastic passenger demand (STSP-GL) with reduced sets of optional stops

Figure 2 **Main components of the DAS lines for deterministic and stochastic passenger demand.**

passenger routes for the stochastic demand on the Hamiltonian circuit. Note that the Hamiltonian circuit is undirected while the passenger routes are directed. In this tactical planning problem, no decisions are taken after uncertainty is resolved, and perfect information is known. Our objective is to minimize a combination of the design cost of the route, i.e., the sum of selected edge costs, as well as the expected average passenger routing cost.

We note that our problem description differs from existing literature as it does not aim to find an ordering among all stops but among a subset of stops. Contrary to Errico et al. (2013) and Errico et al. (2021), where the selection of stops solely happens at a strategic level, our approach allows for more flexibility and validation of strategic decisions on a tactical level as based on the results of the STSP-GL an operator may want to disregard stops from the set of all stops chosen at a strategic level.

4.3.1. Chance-constrained STSP-GL

We present a similar mathematical formulation to Errico et al. (2017) to stay consistent with the literature. We consider a complete mixed graph $G = (N, E \cup A)$, where $N = \{1, \ldots, n\}$ describes a set of nodes, $E = \{[i, j] : i, j \in N, i < j\}$ describes a set of edges that models an ordered tuple of nodes, and $A = \{(i, j) : i, j \in N, i \neq j\}$ describes a set of directed arcs. Here, edges model vehicle movement while arcs model passenger movement. We associate a *design* cost $\bar{c}_{[i,j]} > 0$ to each edge $[i, j] \in E$. Moreover, let $d^{hk}(\xi) \geq 0$ be the uncertain amount of demand for each origin and destination node pair (h, k). Here, $d^{hk}(\xi) \in \Xi$ describes a discrete random variable. Let $D = \{(h, k) : h, k \in N, P(d^{hk}(\xi) > 0) > 0\}$ be the set of node pairs with a positive probability of non-zero demand. We associate a travel time c_{ij} to each arc $(i, j) \in A$ and a random variable for the *routing cost* $q^{hk}_{ij}(\xi) = c_{ij} \frac{d^{hk}(\xi)}{\theta \sum_{(s,t) \in D} d^{st}(\xi)}$ to every arc $(i, j) \in A$ and every node pair $(h, k) \in D$ that depends in turn on the random variable $d^{hk}(\xi)$. Furthermore, we set a *service level* $\theta \in [0, 1]$ and a *feasibility* *frequency* $1-\rho \in [0,1]$. Let α be a tradeoff parameter between design and routing costs. We seek to find a subset $T \subset E$ of edges, such that T describes a tour in G which visits all compulsory nodes $\mathcal{C}$ and enables θ percentage of all customers to travel from their origins to their destinations on the arcs whose corresponding edges are in the tour with a probability of $1-\rho$.

Accordingly, we introduce decision variables $x_{[i,j]}$ which indicate whether edge $[i,j] \in E$ lies in T ($x_{[i,j]} = 1$) or not ($x_{[i,j]} = 0$). Additionally, decision variables f_{ij}^{hk} indicate the percentage of demand $(h,k) \in D$ on arc $(i,j) \in A$. Furthermore, decision variables z^{hk} indicate whether tour T serves demand pair (h,k) ($z^{hk} = 1$) or not ($z^{hk} = 0$) and w_i indicates whether stop $i \in N$ is visited by tour T ($w_i = 1$) or not ($w_i = 0$). Consequently, $\sum_{(i,j)\in A} \sum_{(h,k)\in D} q_{ij}^{hk}(\xi) f_{ij}^{hk}$ measures the expected average passenger travel time for service level θ. Notice that, similar to the TSP-GL, we do not consider any capacity restrictions in our problem. We formulate the STSP-GL as follows:

$$\min_{x,f} \quad (1-\alpha) \sum_{[i,j]\in E} \bar{c}_{[i,j]} x_{[i,j]} + \alpha \mathbb{E}\left[\sum_{(i,j)\in A} \sum_{(h,k)\in D} q_{ij}^{hk}(\xi) f_{ij}^{hk} \right] \tag{1a}$$

$$\mathcal{P}\Big(\sum_{(h,k)\in D} z^{hk} d^{hk}(\xi) \geq \theta \sum_{(h,k)\in D} d^{hk}(\xi) \Big) \geq 1-\rho \tag{1b}$$

$$\sum_{(i,j)\in A} f_{ij}^{hk} - \sum_{(j',i)\in A} f_{j'i}^{hk} = \begin{cases} z^{hk} & \text{if } i = h, \\ -z^{hk} & \text{if } i = k, \\ 0 & i \neq h,k \end{cases} \qquad \forall i \in N,\ \forall (h,k) \in D \tag{1c}$$

$$w_i = 1 \qquad \forall i \in \mathcal{C} \tag{1d}$$

$$\sum_{[i,j]\in E} x_{[i,j]} = 2w_i \qquad \forall i \in N \tag{1e}$$

$$w_h \geq z^{hk} \qquad \forall (h,k) \in D \tag{1f}$$

$$w_k \geq z^{hk} \qquad \forall (h,k) \in D \tag{1g}$$

$$f_{ij}^{hk} \geq 0 \qquad \forall (i,j) \in A,\ \forall (h,k) \in D \tag{1h}$$

$$x_{[i,j]} - f_{ij}^{hk} \geq 0 \qquad \forall [i,j] \in E,\ \forall (h,k) \in D \tag{1i}$$

$$x_{[i,j]} - f_{ji}^{hk} \geq 0 \qquad \forall [i,j] \in E,\ \forall (h,k) \in D \tag{1j}$$

$$x_{[i,j]} \in \{0,1\} \qquad \forall [i,j] \in E \tag{1k}$$

$$z^{hk} \in \{0,1\} \qquad \forall (h,k) \in D \tag{1l}$$

$$w_i \in \{0,1\} \qquad \forall i \in N \tag{1m}$$

The objective function (1a) accounts for the sum of design costs and the sum of expected passenger routing costs. Constraint (1b) imposes that with a probability of $1-\rho$, a service level of θ is met. The multicommodity flow balance constraints (1c) ensure that if demand pair (h,k) is served

($z^{hk}=1$), one unit of flow is sent from h to k. Here, similar to the TSP-GL formulation, we model the routing of passenger demand by unit-flow variables and let q_{ij}^{hk} account for the application-specific interpretation of these routing costs (Errico et al. 2017). Constraints (1d) ensure that every compulsory stop is included in the tour and Constraints (1e) - (1g) impose that if demand pair (h,k) is served ($z^{hk}=1$), the selected edges incident to nodes h and k must be exactly two. Constraints (1i) and (1j) enforce that any commodity (h,k) can only travel on arcs (i,j) and (j,i) if the corresponding edge $[i,j]$ is in tour T. Note that our formulation does not include subtour elimination constraints as we focus on problem instances in which probability distributions induce a single connected component, i.e., there do not exist disjunct subsets of nodes such that passengers mostly travel only between nodes of the same subset. This is the case if D is dense enough and consequently does not induce more than one connected component. For a more thorough explanation on the absence of subtour elimination constraints, we refer to Errico et al. (2017).

4.3.2. Deterministic equivalent of the STSP-GL

The computational tractability of Problem 1 depends on characteristics of the uncertainty considered, i.e., whether it is possible to decouple decisions from random variables to transform probabilistic into deterministic constraints. Our problem is a chance-constrained optimization problem as an operator aims for a tour that fulfills a service level for a share instead of all passenger requests. In this setting, we study the problem's sample counterpart, which becomes computationally more tractable as it bears only a finite number of constraints. We now present a deterministic equivalent of our stochastic program, which ensures that the service level is above θ to a predefined feasibility frequency of $1-\rho$. We introduce binary variables Y_s, which indicate whether service level θ is fulfilled ($Y_s=1$) or not ($Y_s=0$) for a given scenario $s\in S$ in the scenario set of passengers with passenger demands d_s^{hk} and reformulate Problem 1 as Problem 2.

$$\min_{x,f} \quad (1-\alpha)\sum_{[i,j]\in E}\bar{c}_{[i,j]}x_{[i,j]}+ \alpha\sum_{(i,j)\in A}\sum_{(h,k)\in D}\tilde{q}_{ij}^{hk}f_{ij}^{hk} \tag{2a}$$

$$\sum_{(h,k)\in D} z^{hk}d_s^{hk} \geq \theta Y_s \sum_{(h,k)\in D} d_s^{hk} \qquad \forall s\in S \tag{2b}$$

$$\sum_{s\in S} Y_s \geq (1-\rho)|S| \tag{2c}$$

$$\sum_{(i,j)\in A} f_{ij}^{hk} - \sum_{(j',i)\in A} f_{j'i}^{hk} = \begin{cases} z^{hk} & \text{if } i=h,\\ -z^{hk} & \text{if } i=k,\\ 0 & i\neq h,k \end{cases} \qquad \forall i\in N,\ \forall(h,k)\in D \tag{2d}$$

$$w_i = 1 \qquad \forall i \in \mathcal{C} \tag{2e}$$

$$\sum_{[i,j] \in E} x_{[i,j]} = 2w_i \qquad \forall i \in N \tag{2f}$$

$$w_h \geq z^{hk} \qquad \forall (h,k) \in D \tag{2g}$$

$$w_k \geq z^{hk} \qquad \forall (h,k) \in D \tag{2h}$$

$$f_{ij}^{hk} \geq 0 \qquad \forall (i,j) \in A,\ \forall (h,k) \in D \tag{2i}$$

$$x_{[i,j]} - f_{ij}^{hk} \geq 0 \qquad \forall [i,j] \in E,\ \forall (h,k) \in D \tag{2j}$$

$$x_{[i,j]} - f_{ji}^{hk} \geq 0 \qquad \forall [i,j] \in E,\ \forall (h,k) \in D \tag{2k}$$

$$x_{[i,j]} \in \{0,1\} \qquad \forall [i,j] \in E \tag{2l}$$

$$z^{hk} \in \{0,1\} \qquad \forall (h,k) \in D \tag{2m}$$

$$w_i \in \{0,1\} \qquad \forall i \in N \tag{2n}$$

Here, we only reformulate Constraint (1b) as Constraints (2b) and Constaint (2c). All other constraints are unchanged. Additionally, we update the passenger routing cost $\tilde{q}_{ij}^{hk} = c_{ij}\left(\frac{1}{|S|}\sum_{s \in S}\frac{d_s^{hk}}{\theta \sum_{(u,v)\in D} s^{uv}}\right)$ to account for the deterministic counterpart of our stochastic program. In his context, we assume the scenarios to be equally likely.

4.3.3. Decomposition of STSP-GL using feasibility covers

This deterministic equivalent allows us to solve small instances but is still not tractable for larger ones since the number of combinations for z is exponential in the size of D. Accordingly, we utilize the notion of feasibility covers to further decompose our model to utilize Benders decomposition (Errico et al. 2017) for the resulting subproblems.

Definition 1 (Feasibility cover). A *feasibility cover* is a set $Q \subseteq D$ such that there exists a vector $\gamma \in \{0,1\}^{|S|}$ which is feasible for

$$\sum_{(h,k)\in Q} d_s^{hk} \geq \theta \gamma_s \sum_{(h,k)\in D} d_s^{hk} \qquad \forall s \in S, \tag{3a}$$

$$\sum_{s \in S} \gamma_s \geq (1-\rho)|S| \tag{3b}$$

In other words, a feasibility cover is a subset of requests for which the service level is met for $\lceil (1-\rho)|S| \rceil$ many scenarios.

Definition 2 (Minimal feasibility cover). A *minimal feasibility cover* is a feasibility cover $Q \subseteq D$ such that there is no feasibility cover $Q' \subseteq D$ such that $Q' \subset Q$.

Theorem 1. *For an optimal solution* (x^*, f^*, z^*, Y^*) *of Problem 2,* $K = \{(h,k) : (z^{hk})^* = 1, (h,k) \in D\}$ *is a minimal feasibility cover.*

Proof. Assume we have an optimal solution (x^*, f^*, z^*, Y^*) such that the corresponding set K is not a minimal feasibility cover. Since K is not a minimal feasibility cover, there has to exist a passenger request $(u,v) \in K$ for which there exists a $\gamma' \in \{0,1\}^{|S|}$ which fulfills

$$\sum_{(h,k)\in K-\{(u,v)\}} d_s^{hk} \geq \theta\gamma'_s \sum_{(h,k)\in D} d_s^{hk} \qquad \forall\, s \in S \tag{4a}$$

$$\sum_{s\in S} \gamma'_s \geq (1-\rho)|S| \tag{4b}$$

Accordingly, (x', f', z', Y') with

$$\begin{aligned}
x' &= x^* \\
(f_{ij}^{hk})' &= \begin{cases} (f_{ij}^{hk})^*, & \text{if } (h,k) \neq (u,v), \\ 0, & \text{if } (h,k) = (u,v), \end{cases} \\
(z^{hk})' &= \begin{cases} (z^{hk})^*, & \text{if } (h,k) \neq (u,v), \\ 0, & \text{if } (h,k) = (u,v), \end{cases} \\
Y' &= \gamma'
\end{aligned}$$

is feasible for Problem 2 and has an objective value of

$$\begin{aligned}
obj((x', f', z', Y')) = obj((x^*, f^*, z^*, Y^*)) - \\
\underbrace{\sum_{(i,j)\in A} c_{ij} \Big(\frac{1}{|S|} \sum_{s\in S} \frac{d_s^{hk}}{\theta \sum_{(r,t)\in D} s^{rt}} \Big) f_{ij}^{uv}}_{>0}
\end{aligned} \tag{5}$$

This contradicts our assumption as (x^*, f^*, z^*, Y^*) is an optimal solution. □

Definition 1 allows us to reformulate Problem 2 by using feasibility covers.

Let $\mathcal{X}_{\theta,\rho}$ be the set of all feasibility covers of D. We can then formulate our original problem as Problem 6.

$$\begin{aligned}
\min_{x,f} \quad (1-\alpha) \sum_{[i,j]\in E} \overline{c}_{[i,j]} x_{[i,j]} + \\
\alpha \sum_{(i,j)\in A} \sum_{(h,k)\in D} \tilde{q}_{ij}^{hk} f_{ij}^{hk}
\end{aligned} \tag{6a}$$

$$\sum_{Q\in\mathcal{X}_{\theta,\rho}} \chi_Q = 1 \tag{6b}$$

$$\sum_{[i,j]\in E} x_{[ij]} - 2 \sum_{Q\in\mathcal{X}_{\theta,\rho}} \chi_Q l_Q(i) = 0 \qquad \forall i \in N \tag{6c}$$

$$\begin{aligned}
\sum_{(i,j)\in A} f_{ij}^{hk} - \sum_{(j',i)\in A} f_{j'i}^{hk} = \\
\begin{cases} \sum_{Q\in\mathcal{X}_{\theta,\rho}} \chi_Q r_Q^{hk}, \text{ if } i = h, \\ -\sum_{Q\in\mathcal{X}_{\theta,\rho}} \chi_Q r_Q^{hk}, \text{ if } i = k, \\ 0, \text{else} \end{cases} \\
\forall i \in N,\ \forall\, (h,k) \in D
\end{aligned} \tag{6d}$$

$$f_{ij}^{hk} \geq 0 \qquad \forall (i,j) \in A,\ \forall (h,k) \in D \tag{6e}$$

$$x_{[i,j]} - f_{ij}^{hk} \geq 0 \qquad \forall [i,j] \in E,\ \forall (h,k) \in D \tag{6f}$$

$$x_{[i,j]} - f_{ji}^{hk} \geq 0 \qquad \forall [i,j] \in E,\ \forall (h,k) \in D \tag{6g}$$

$$x_{[ij]} \in \{0,1\} \qquad \forall [i,j] \in E \tag{6h}$$

$$\chi_Q \in \{0,1\} \qquad \forall Q \in \mathcal{X}_{\theta,\rho} \tag{6i}$$

Here, $l_Q(i) = 1$ if i is a compulsory stop ($i \in \mathcal{C}$) or there exists a demand pair $(h,k) \in Q$ with $i = h$ or $i = k$ and $l_Q(i) = 0$ otherwise. Furthermore, $r_Q^{hk} = 1$ if $(h,k) \in Q$ and $r_Q^{hk} = 0$ otherwise. Constraint (6b) ensures that exactly one feasibility cover is chosen. Constraints (6c) correspond to Constraints (2f) and ensure that a tour visits all stops for which $l_Q(i) = 1$ if Q is chosen in (6c). Constraints (6d) are flow conservation constraints for all requests in Q if Q is chosen in (6c). Constraints (6e)-(6h) did not change compared to Problem 1. Constraints (6i) define χ_Q's domain.

In the remainder of this section, we show that we can find an optimal solution of our STSP-GL by determining a feasibilty cover whose corresponding TSP-GL (see Problem 7) has the smallest objective function out of all feasibility covers' corresponding TSP-GLs. To do so, we first show the corresponding TSP-GL to a feasibility cover (see Problem 7) can be solved on a subgraph G' of our original graph G. Second, we show that once we chose a feasibility cover in our STSP-GL, the STSP-GL formulation decomposes into the aformentioned TSP-GL (see Theorem 3).

Let us revisit the TSP-GL given a feasibility cover Q on the complete mixed graph $G'(Q) \subset G$ where $G'(Q) = (N'(Q), E'(Q) \cup A'(Q))$ with $N'(Q) = \{i : (i,j) \in Q \textit{ or } (j',i) \in Q\}$, $E'(Q) = \{[i,j] : [i,j] \in E,\ i,j \in N'(Q)\}$ and $A'(Q) = \{(i,j) : (i,j) \in A,\ i,j \in N'(Q)\}$. We formulate the corresponding TSP-GL for Q as follows:

$$\min_{\bar{x},\bar{f}} (1-\alpha) \sum_{[i,j] \in E'(Q)} \bar{c}_{[i,j]} \bar{x}_{[i,j]} + \alpha \sum_{(i,j) \in A'(Q)} \sum_{(h,k) \in Q} \tilde{q}_{ij}^{hk} \bar{f}_{ij}^{hk} \tag{7a}$$

$$\sum_{(i,j) \in A'(Q)} \bar{f}_{ij}^{hk} - \sum_{(j',i) \in A'(Q)} \bar{f}_{j'i}^{hk} = \begin{cases} 1 & \text{if } i = h, \\ -1 & \text{if } i = k, \\ 0 & i \neq h,k \end{cases} \qquad \forall i \in N'(Q),\ \forall (h,k) \in Q \tag{7b}$$

$$\sum_{[i,j] \in E'(Q)} \bar{x}_{[i,j]} = 2 \qquad \forall i \in N'(Q) \tag{7c}$$

$$\bar{f}_{ij}^{hk} \geq 0 \qquad \forall (i,j) \in A'(Q),\ \forall (h,k) \in Q \tag{7d}$$

$$\bar{x}_{[i,j]} - \bar{f}_{ij}^{hk} \geq 0 \qquad \forall [i,j] \in E'(Q),\ \forall (h,k) \in Q \tag{7e}$$

$$\bar{x}_{[i,j]} - \bar{f}_{ji}^{hk} \geq 0 \qquad \forall [i,j] \in E'(Q),\ \forall (h,k) \in Q \tag{7f}$$

$$\bar{x}_{[i,j]} \in \{0,1\} \qquad \forall [i,j] \in E'(Q) \tag{7g}$$

THEOREM 2. *If Problem 6 is feasible and triangle inequality holds for $\bar{c}$, there exists an optimal solution, in which for each served request in an optimal feasibility cover Q^*, i.e., $r \in Q^*$ with $\chi_{Q^*} = 1$, it holds that $\bar{x}^*_{[i,j]} = 0\, \forall [i,j] \in E \setminus E'(Q^*)$ and $(\bar{f}^{hk}_{ij})^* = 0\, \forall (i,j) \in A$, $\forall (h,k) \in D$ if $(i,j) \notin A'(Q^*)$ or $(h,k) \notin Q^*$.*

Proof. All edges in $E \setminus E'(Q^*)$ are adjacent to a node $i \notin N'(Q^*)$, i.e., a node to which no request in Q^* has an origin or destination. If the tour formed by $\bar{x}$ includes such an edge, the design costs of an alternative tour which skips all nodes $i \notin N'(Q^*)$ would always be lower or equal to the original route because triangle inequality holds. Similarly, as no request in Q^* has an origin or destination in a node $i \notin N'(Q^*)$, alternative passenger routes could also skip these nodes, which, because of triangle inequality, leads to passenger routes with equal or smaller costs. As the original passenger routes only use arcs whose corresponding edges are in the tour formed by $\bar{x}$ and the alternative passenger routes would skip the same nodes as the alternative tour, all passenger routes would still be feasible. □

DEFINITION 3 (CHARACTERISTIC FUNCTION OF TSP-GL VARIABLES). For a solution $(\bar{x}, \bar{f})$ of Problem 7 with feasibility cover $Q \in \mathcal{X}_{\theta,\rho}$, define $\hat{x}(Q)$ and $\hat{f}(Q)$ as follows:

$$\hat{x}_{[i,j]}(Q) = \begin{cases} \bar{x}_{[i,j]} & \text{if } [i,j] \in E'(Q), \\ 0 & \text{else} \end{cases} \quad \forall [i,j] \in E \tag{8}$$

$$\hat{f}^{h,k}_{i,j}(Q) = \begin{cases} \bar{f}^{h,k}_{i,j} & \text{if } (i,j) \in A'(Q) \, and \, (h,k) \in Q, \\ 0 & \text{else} \end{cases} \quad \forall (i,j) \in E \, and \, (h,k) \in D \tag{9}$$

With the new STSP-GL formulation in Problem 6, the TSP-GL formulation for a feasibility cover Q in Problem 7, and the definition of $\hat{x}(Q)$ and $\hat{f}(Q)$, we can show the following theorem.

THEOREM 3. *If triangle inequality holds for $\bar{c}$ then for all $\chi = \mathbb{1}_{\{Q=Q'\}}$, Problem 6 decomposes into Problem 7. Furthermore, $(\mathbb{1}_{\{Q=Q^*\}}, \hat{x}(Q^*), \hat{f}(Q^*))$, with $Q^* = arg\,min\{obj_{TSP-GL}(Q) : Q \in \mathcal{X}_{\theta,\rho}\}$ is an optimal solution of Problem 6, with Q^* being a minimal feasibility cover.*

Proof. According to Theorem 2, we can assume that for an optimal solution $(\mathbb{1}_{\{Q=Q^{**}\}}, \bar{x}(Q^{**}), \bar{f}(Q^{**}))$ it holds that $\bar{x}^{**}_{[i,j]} = 0\; \forall [i,j] \in E \setminus E'(Q^{**})$ and $(\bar{f}^{hk}_{ij})^{**} = 0\; \forall (i,j) \in A$, $\forall (h,k) \in D$ if $(i,j) \notin A'(Q^{**})$ or $(h,k) \notin Q^{**}$. Thus, we can fix these decision variables for each $Q' \in \mathcal{X}_{\theta,\rho}$. In this case, the objective values of (6a) and (7a) are equal. Furthermore, since $\chi = \mathbb{1}_{\{Q=Q'\}}$, Constraints (6c) are equal to Constraints (7c) $\forall i \in N'(Q')$. By definition of $r^{hk}_{Q'}$ and since $\chi_{Q'} = 1$, Constraints (6d) are equal to Constraints (7b) $\forall i \in N'(Q')$, $\forall (h,k) \in Q'$. Additionally, Constraints (6e) - (6h) are equal to Constraints (7d) - (7g) for all variables which are not fixed through Theorem 2. Lastly, for each $Q' \in \mathcal{X}_{\theta,\rho}$, Constraint (6b) and Constraints (6i) hold for $\chi = \mathbb{1}_{\{Q=Q'\}}$. This shows, that if the triangle inequality holds for $\bar{c}$ then for all $\chi = \mathbb{1}_{\{Q=Q'\}}$, Problem 6 decomposes into Problem 7. The second part of the theorem follows directly. Here, Theorem 1 induces that Q^* is a minimal feasibility cover. □

Corollary 1. *Since the feasibility cover inducing an optimal solution for Problem 6 must be minimal (Theorem 3), utilizing minimal feasibility covers in set $\mathcal{X}_{\theta,\rho}$ of all feasibility covers of D in Problem 6 yields a tighter problem formulation than if using the set of all feasibility covers. Specifically, let $P(\mathcal{X}_{\theta,\rho})$ be the feasible region of Problem 6 utilizing feasibility covers and $P(\mathcal{X}^{min}_{\theta,\rho})$ be the feasible region utilizing minimal feasibility covers. Then $P(\mathcal{X}^{min}_{\theta,\rho}) \subseteq P(\mathcal{X}_{\theta,\rho})$, where $P(\mathcal{X}^{min}_{\theta,\rho})$ includes an optimal solution of Problem 6.*

4.4. Methodology

In this section, we first show a standard B&P approach (Section 4.4.1) on which we base our adapted B&P approach (Section 4.4.2). Then, we introduce a heuristic approach that allows us to solve the STSP-GL without finding solutions for its continuous relaxation (Section 4.4.3). Finally, we propose a hybrid approach in which we combine our adapted B&P and heuristic approaches (Section 4.4.4).

4.4.1. Branch-and-price approach

According to Theorem 3, we must find the feasibility cover Q^* to solve Problem 6. Even determining $\mathcal{X}_{\theta,\rho}$ is not a trivial task as the number of feasibility covers may grow exponentially in the size of all requests D. One straightforward approach to find Q^* is to embed Problem 6 in a B&P framework (Figure 3) in which the nodes in the branch-and-bound tree (green) with integer decision variables $x_{[i,j]}$ decompose into TSP-GLs.

In such a B&P framework, we start with a small subset $\mathcal{X}'_{\theta,\rho}$ of feasibility covers and iteratively add new feasibility covers to $\mathcal{X}'_{\theta,\rho}$ via column generation by solving the so-called pricing problem. In this context, let $\iota_i \in \mathbb{R}$ and $\epsilon^{hk}_h, \epsilon^{hk}_k \in \mathbb{R}$ be the dual variables of Constraints (6c) and (6d) respectively. The scalar β is the dual value of Constraint (6b). We aim to find a feasibility cover Q

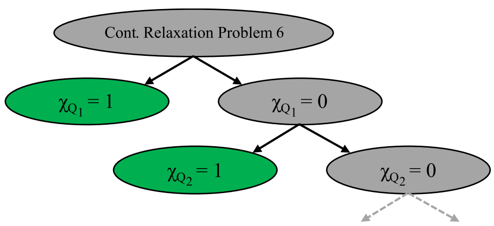


Figure 3 Branch-and-Price algorithm

in each iteration of our column generation for which the reduced cost condition (10), that indicates whether a variable should be brought into the basis, holds.

$$2\sum_{i\in N}\iota_i l_Q(i) + \sum_{(h,k)\in D} r_Q^{hk}(\epsilon_h^{hk} - \epsilon_k^{hk}) - \beta < 0 \tag{10}$$

Accordingly, we formulate our pricing problem as follows.

$$\min_{l,r} \quad 2\sum_{i\in N}\iota_i l_i + \sum_{(h,k)\in D} r^{hk}(\epsilon_h^{hk} - \epsilon_k^{hk}) - \beta \tag{11a}$$

$$l_i = 1 \qquad \forall\, i \in \mathcal{C} \tag{11b}$$

$$l_h \geq r^{hk} \qquad \forall\, (h,k) \in D \tag{11c}$$

$$l_k \geq r^{hk} \qquad \forall\, (h,k) \in D \tag{11d}$$

$$\sum_{(h,k)\in D} r^{hk} d_s^{hk} \geq \gamma_s \theta \sum_{(h,k)\in D} d_s^{hk} \qquad \forall\, s \in S \tag{11e}$$

$$\sum_{s\in S}\gamma_s \geq (1-\rho)|S| \tag{11f}$$

$$l_i \in \{0,1\} \qquad \forall\, i \in N \tag{11g}$$

$$r^{hk} \in \{0,1\} \qquad \forall\, (h,k) \in D \tag{11h}$$

$$\gamma_s \in \{0,1\} \qquad \forall\, s \in S \tag{11i}$$

In Problem 11, the decision variables r^{hk} determine whether request $(h,k) \in D$ is in the feasibility cover ($r^{hk} = 1$) or not ($r^{hk} = 0$). The decision variables l_i indicate whether at least one request (i,k) or (h,i) is in the feasibility cover or i is a compulsory node ($l_i = 1$) or not ($l_i = 0$). This pricing problem does not ensure that the feasibility covers determined by r^{hk} are minimal. We can enforce this by adding Constraints (12) as lazy constraints. Here, Ψ is the set of feasibility covers that are not minimal.

$$\sum_{(h,k)\in Q} r^{hk} \leq |Q| - 1 \qquad \forall Q \in \Psi \tag{12}$$

To consider the branching constraints $\chi_{Q_i} = 0$ of the B&P, we extend the pricing problem by Constraints (13).

$$\sum_{(h,k)\in Q_i} r^{hk} \leq |Q_i| - 1 \qquad \forall Q_i \in \Phi \tag{13}$$

Here, Φ is the set of feasibility covers Q_i on which we already branched, i.e., for which constraints $\chi_{Q_i} = 0$ exists.

4.4.2. Adapted branch-and-price approach

One drawback of such a straightforward approach is that the large number of feasibility covers makes solving even the root node of the B&P approach very hard. Accordingly, we propose an adapted B&P approach in which we iteratively find a set of feasibility covers through our Pricing Problem 11 without solving the continuous relaxation of our problem to optimality and solve their corresponding TSP-GLs to determine feasible solutions.

We initialize our algorithm by building our restricted master problem (RMP) with the empty set of minimal feasibility covers and variable χ_D, i.e., the feasibility cover, which includes all requests. Additionally, we build our Pricing Problem 11 with an empty set of branching Constraints (13). Afterward, we divide each iteration of our algorithm into three phases (see Figure 4).

In the first phase, we repeatedly solve Pricing Problem 11 as an IP to generate new minimal feasibility covers. Whenever we find a new minimal feasibility cover with negative reduced cost, we update the lower bound of our B&P and add it to the RMP.

In the second phase, we apply a heuristic mechanism to score these new found minimal feasibility covers and add them to a list of unevaluated covers. By doing so, we allow for an efficient evaluation of feasibility covers, as solving the corresponding TSP-GL of a minimal feasibility cover takes relatively long, and our heuristic mechanism allows us to discard bad minimal feasibility covers much faster than solving the corresponding TSP-GL. Our heuristic mechanism scores a minimal feasibility cover Q by evaluating its upper and lower bounds for the TSP-GL. If, for a minimal feasibility cover Q, the lower bound of its corresponding TSP-GL is higher than the upper bound of our B&P, we can discard Q as an optimal solution.

In the third phase, we evaluate a subset of unevaluated minimal feasibility covers chosen based on the heuristic scores by solving their corresponding TSP-GLs. Accordingly, we branch on the most promising minimal feasibility covers, i.e., the minimal feasibility covers in the set of unevaluated covers with the smallest upper bounds.

Determination of upper and lower bounds: In the second phase of our B&P approach, we score feasibility covers. Here, the objective of the TSP-GL contains two potentially conflicting terms, i.e.,

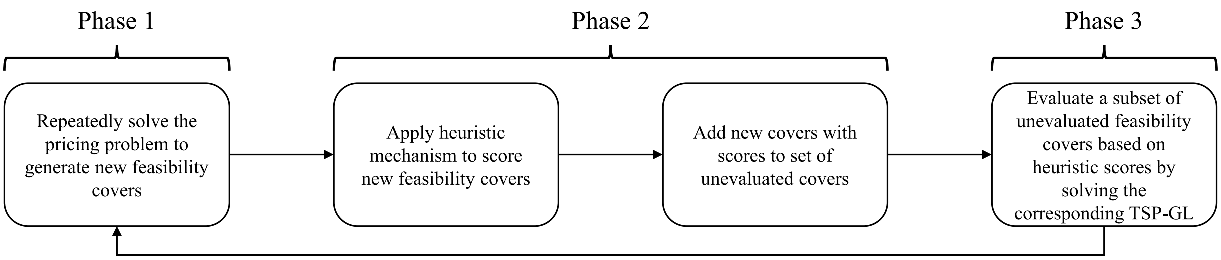


Figure 4 Flowchart B&P approach algorithm

design cost $\sum_{[i,j]\in E}\bar{c}_{[i,j]}x_{[i,j]}$ and routing cost $\sum_{(i,j)\in A}\sum_{(h,k)\in D}\tilde{q}_{ij}^{hk}f_{ij}^{hk}$. To derive a lower bound on the objective, we derive lower bounds on the individual terms. Accordingly, we solve the symmetric TSP on compulsory nodes and nodes with requests going from or to them in Q to obtain a lower bound on the design costs (lb_{design}). Furthermore, for each request $(h,k)\in Q$, we assume a lower bound of $\tilde{q}_{hk}^{hk}$ for the routing cost ($lb_{routing}^{hk}$), which corresponds to a direct route from the origin h to the destination k. Accordingly, $(1-\alpha)lb_{design}+\alpha\sum_{(h,k)\in Q}lb_{routing}^{hk}$ is a lower bound for the TSP-GL on Q. For the upper bound, we compute the shortest path of length $ub_{routing}^{hk}$ for each request $(h,k)\in Q$ on the solution of the symmetric TSP. Accordingly, $(1-\alpha)lb_{design}+\alpha\sum_{(h,k)\in Q}ub_{routing}^{hk}$ is an upper bound for the TSP-GL on Q.

Benders decomposition for the TSP-GL: To solve the corresponding TSP-GLs in the third phase of our B&P approach, similarly to Errico et al. (2017), we observe that for fixed design decision variables $\bar{x}$, the TSP-GL reduces to solving a one-commodity minimum cost flow model for each request in our minimal feasibility cover. Accordingly, we utilize a Benders decomposition approach to solve the TSP-GL. Denoting the routing cost of request $(h,k)\in Q$ with fixed design decision variables $\bar{x}$ as $z^{hk}(\bar{x})$, we define our PrimalSubproblem (PS) for each request (h,k) as

$$z^{hk}(\bar{x})=\min_{\bar{f}}\quad\sum_{(i,j)\in A'(Q)}\tilde{q}_{ij}^{hk}\bar{f}_{ij}^{hk}\tag{14a}$$

$$\sum_{(i,j)\in A'(Q)}\bar{f}_{ij}^{hk}-\sum_{(j',i)\in A'(Q)}\bar{f}_{j'i}^{hk}=\begin{cases}1 & \text{if } i=h,\\ -1 & \text{if } i=k,\\ 0 & i\neq h,k\end{cases}\qquad\forall i\in N'(Q)\tag{14b}$$

$$\bar{f}_{ij}^{hk}\geq 0\qquad\forall(i,j)\in A'(Q),\ \forall(h,k)\in Q\tag{14c}$$

$$\bar{x}_{[i,j]}-\bar{f}_{ij}^{hk}\geq 0\qquad\forall[i,j]\in E'(Q),\ \forall(h,k)\in Q\tag{14d}$$

$$\bar{x}_{[i,j]}-\bar{f}_{ji}^{hk}\geq 0\qquad\forall[i,j]\in E'(Q),\ \forall(h,k)\in Q\tag{14e}$$

As strong duality holds for each flow subproblem, we consider the DualSubproblems (DS) for every flow subproblem. Here, p_i^{hk} are the dual variables of Constraints (14b) $\forall i\in N'(Q)$ and λ_{ij}^{hk} the dual variables of Constraints (14d) and (14e) $\forall(i,j)\in A'(Q)$.

$$z^{hk}(\bar{x})=\max\quad p_h^{hk}-p_k^{hk}-\sum_{[i,j]\in E'(Q)}\bar{\lambda}_{[i,j]}^{hk}\bar{x}_{[i,j]}\tag{15a}$$

$$p_i^{hk}-p_j^{hk}-\lambda_{ij}^{hk}\leq q_{ij}^{hk}\qquad\forall(i,j)\in A'(Q)\tag{15b}$$

$$\lambda_{ij}^{hk}\geq 0\qquad\forall(i,j)\in A'(Q)\tag{15c}$$

$$p_i^{hk}\in\mathbb{R}\qquad\forall(i,j)\in A'(Q)\tag{15d}$$

Here, we define $\bar{\lambda}_{[i,j]}:=\lambda_{ij}+\lambda_{ji}$, $\forall[i,j]\in E'(Q)$. We now denote $extr(R)$ and $rays(R)$ as the set of extreme points and rays of $R=\{(\lambda,p)\in\mathbb{R}^{(|A'(Q)|+|N'(Q)|)|Q|}|(\lambda,p)$ that satisfy $(15b)-(15d)\}$

respectively. Consequently, as shown by Errico et al. (2017), we can express Problem 7 as Problem Problem 16 because we obtain $z(x)$ as a pointwise maximum of affine functions for a generic x and the number of extreme points and rays are finite.

$$\min \quad \alpha\eta + (1-\alpha) \sum_{[i,j]\in E'(Q)} \bar{c}_{[i,j]} x_{[i,j]} \tag{16a}$$

$$\eta + \sum_{\substack{[i,j]\in E'(Q)\\(h,k)\in Q}} \bar{\lambda}^{hk}_{[i,j]} x_{[i,j]} \geq \sum_{(h,k)\in Q} (p^{hk}_h - p^{hk}_k) \qquad \forall(\lambda,p)\in extr(R) \tag{16b}$$

$$\sum_{\substack{[i,j]\in E'(Q)\\(h,k)\in Q}} \bar{\lambda}^{hk}_{[i,j]} x_{[i,j]} \geq \sum_{(h,k)\in Q} (p^{hk}_h - p^{hk}_k) \qquad \forall(\lambda,p)\in rays(R) \tag{16c}$$

$$\sum_{[i,j]\in E'(Q)} x_{[i,j]} = 2 \qquad \forall i \in N'(Q) \tag{16d}$$

$$x_{[i,j]} \in \{0,1\} \qquad \forall [i,j]\in E'(Q) \tag{16e}$$

To solve Problem 16, we have to determine a set of feasibility cuts (16c) and optimality cuts (16b). We initialize our algorithm by using the solution of the symmetric TSP, which we solved for the bounds of Q, which is equivalent to initialization **h2** in Errico et al. (2017). We utilize a subtour elimination constraint strategy for our feasibility cut generation. As such, we adopt (17) as feasibility cuts. Here Θ is the set of all subtours in the current solution.

$$\sum_{[i,j]\in S} x_{[i,j]} \leq |S| - 1 \qquad \forall S \in \Theta \tag{17}$$

According to Errico et al. (2017), we can add cuts (18) as optimality cuts.

$$\eta^h + \sum_{\substack{k\in Nn[h]\\e\in E}} \bar{\lambda}^{hk}_e x_e \geq \sum_{k\in Nn[h]} p^{hk}_h - p^{hk}_k \qquad \forall h \in N'(Q) \tag{18}$$

Here, η^h is the contribution, of all commodities with origin h, with $\sum_{h\in N} \eta^h = \eta$, to η.

Accordingly, in our Benders approach, we initialize Problem 16 with a set of feasibility cuts, which are equal to the subtour elimination constraints we generated in the calculation of our bounds for the minimal feasibility covers and an empty set of optimality cuts. We additionally use the solution of the symmetric TSP obtained in the calculation of the bounds to generate optimality cuts. We then resolve Problem 16 and check if we can add new feasibility cuts. If not, we solve Problem 15 for each request $(h,k)\in Q$ and add new optimality cuts. We begin a new iteration by resolving Problem 16 and terminate when we can not find new feasibility or optimality cuts. Note that we can add a set of feasibility cuts based on all found subtours in the calculation of the symmetric TSP when we initialize our Benders approach.

Note that an alternative branching strategy in which we branch on the served requests yields a more balanced search tree but does not allow us to utilize our problem decomposition based on

Theorem 3 efficiently. This decomposition is essential to derive primal feasible solutions efficiently as it allows us to determine primal solutions without explicitly modeling the flow variables of each request. Accordingly, we focus only on branching on the feasibility set variables χ. For an in-depth discussion on an alternative branching strategy, we refer to Appendix C.

4.4.3. Local search-based heuristic approach

One primary concern with this B&P approach is the time required to solve the RMP in our column generation as the number of flow variables f_{ij}^{hk} and the number of Constraints (6d), (6f), and (6g) grow by a factor of $\mathcal{O}(n^4)$ in the size of the underlying graph. Accordingly, we develop a local search-based heuristic that determines minimal feasibility covers without having to solve the RMP by leveraging the notion of a minimal node cover.

Definition 4 (Minimal node cover). We define a *minimal node cover* V' to be a set for which it holds that $\mathcal{C} \subseteq V' \subseteq V$ and there exists a feasibility cover in $\{(h,k) : h,k \in V'\}$ and no feasibility cover in $\{(h,k) : h,k \in V' \setminus \{v\}\}$ for all $v \in V'$.

Our heuristic consists of four main parts. An exploration step, a local search, a heuristic mechanism to score minimal feasibility covers, and the solving step. In the exploration step, the heuristic aims to find new minimal feasibility covers based on minimal node covers. To do so, we start with a random minimal node cover $V' \subseteq V$ with a predetermined size $|V'| = n$, which has not been found so far. We then determine a minimal feasibility cover $Q \subseteq Q' := \{(h,k) : h,k \in V'\}$. Algorithm 1 outlines how to find a minimal feasibility cover from a feasibility cover Q'. Note that if no feasibility cover of size $|Q|-1$ exists for a feasibility cover Q, Q is minimal. Utilizing this, we initialize Algorithm 1 with a feasibility cover Q' with the aim of determining a minimal feasibility cover $Q \subseteq Q'$. To do so, we assign $Q \leftarrow Q'$ (l. 1) and check whether there exists a feasibility cover $\bar{Q} \subset Q$ with $|\bar{Q}| = |Q| - 1$ (l.2 - 6). If that is the case, we recursively call Algorithm 1 with $\bar{Q}$ as an input (l.5). Otherwise, we return the minimal feasibility cover Q.

In the local search step, the algorithm performs a simple swap procedure on a given feasibility cover. We always perform the local search on two minimal feasibility covers with the smallest upper bounds. Algorithm 2 outlines the procedure for one minimal feasibility cover Q'. Here, we aim to find a new minimal feasibility cover Q similar to Q'. To do so, we assign $Q \leftarrow Q'$ (l. 1) and remove random elements from Q until Q is no longer a feasibility cover (l.3-4). Simultaneously, we add all removed requests to R (l.4). Afterwards, we randomly add new requests that have not been removed before to Q (l.6-11) until either Q is a feasibility cover again (l.8-9) or there are no more requests to add. If Q is a feasibility cover, we determine a minimal feasibility cover through Algorithm 1 and return it (l.12-13). Otherwise, we randomly add previously removed requests to Q until it is a feasibility cover again (l.15-22). We then determine a minimal feasibility cover through Algorithm 1

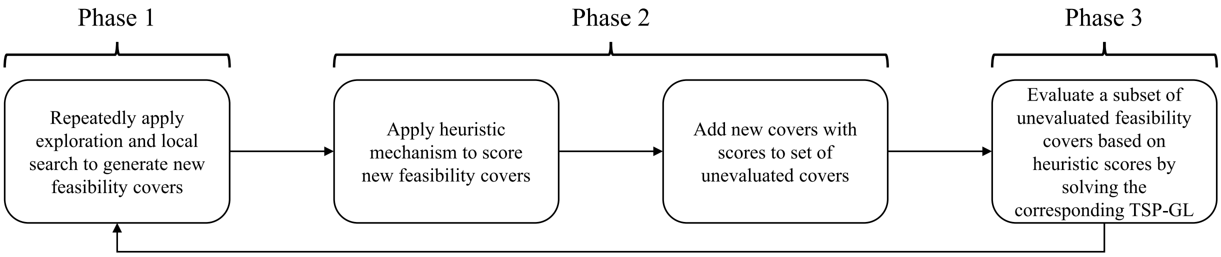


Figure 5 Heuristic flowchart

and return it. For all new-found minimal feasibility covers in the exploration and local search step, we use the same heuristic mechanism as in our B&P approach to score the covers and calculate upper and lower bounds. In the solving step, we choose the cover with the lowest upper bound from our queue and solve Problem 16 on it if its lower bound is smaller than the upper bound of the incumbent. We then update the upper bound of the node by the objective value of the TSP-GL and update the incumbent if necessary. Figure 5 shows the illustrative flowchart for this local search-based heuristic approach. Compared to our B&P approach, only the initialization through an exploration step and first phase, in which we generate minimal feasibility covers, change. The termination criterion for this approach is a time limit.

4.4.4. Hybrid approach

Based on our B&P and heuristic approaches, we can now combine the two of them to develop a hybrid approach. Accordingly, we integrate our heuristic approach's exploration and local search into our B&P approach. To do so, we initialize our B&P with a set of minimal feasibility covers $\mathcal{X}'_{\theta,\rho}$ which we determine through our exploration operator. Additionally, after each pricing problem in our B&P approach, we apply our local search and exploration operator to the new found minimal feasibility cover. This allows us to find minimal feasibility covers similar to the ones determined in

Algorithm 1: Determination of minimal feasibility cover

Data: feasibility cover Q'

Result: minimal feasibility cover Q

$Q \leftarrow Q'$
for $r \in Q$ **do**
  if $Q \setminus \{r\}$ *feasibility cover* **then**
    $\bar{Q} \leftarrow Q \setminus \{r\}$
    return minimal_feasibility_cover($\bar{Q}$)
  end
end
return Q

Algorithm 2: Local search

Data: minimal feasibility cover Q'; all requests D

Result: minimal feasibility cover Q

```
Q ← Q'
R ← ∅
while Q is feasibility cover do
    R.add(Q.pop())
end
for r ∈ D \ R do
    Q.add(r)
    if Q is feasibility cover then
        break
    end
end
if Q is feasibility cover then
    return minimal_feasibility_cover(Q)
end
else
    for r ∈ R do
        Q.add(r)
        if Q is feasibility cover then
            return minimal_feasibility_cover(Q)
        end
    end
end
```

our pricing problem through the local search and minimal feasibility covers, which visit a similar number of nodes as the newfound cover through the exploration. Afterwards, we continue with the second and third phases of our B&P approach.

4.4.5. Speed-up techniques

We use the following speed-up techniques to enhance the performance of our algorithms. First, closing the optimality gap of the continuous solution of the root node of our B&P and hybrid approaches is already challenging. Consequently, we limit the number of solved pricing problems that determine new feasibility covers of our algorithm (Figure 4) to $n = 5$ in each iteration of the column generation. Here, we also determine an overall lower bound of the STSP-GL in each iteration by calculating the sum of the current objective value of our RMP and the respective objective value of our pricing problem.

Second, after solving n pricing problems, for each feasibility cover Q whose corresponding decision variable χ_Q is in the basis of the last solution of the RMP, we calculate the upper and lower bound of their corresponding TSP-GL. We then add each of the selected feasibility covers Q to a priority queue, which sorts them based on their upper bounds.

Third, in both our B&P and hybrid approaches, we check whether the feasibility cover's lower bound is smaller than our incumbent's upper bound for the first $m = 5$ feasibility covers in the priority queue. If this holds, we solve the corresponding TSP-GL. Otherwise, we continue to the next element of the priority queue without solving the TSP-GL. We do not test additional feasibility covers in an iteration, even if we solve no corresponding TSP-GLs.

Fourth, determining the set of all feasibility covers that are not minimal (Ψ) for Constraints (12) is computationally not tractable. Accordingly, we can implement them as lazy constraints. In our computational effectiveness analysis, we did not see a computational advantage of adding these constraints to the pricing problem and sometimes, they even worsened the lower bound of our B&P and hybrid approaches. Accordingly, we excluded them from the results of our computational analysis. Note that our problem reformulation in Section 4.3 holds for feasibility covers and does not require minimal feasibility covers. Intuitively, there is a clear trade-off between using feasibility covers and minimal feasibility covers. On the one hand, we expect that solving the pricing problems for minimal feasibility covers will take more time. On the other hand, we expect that the bound produced when only minimal feasibility covers are returned by the pricing problem will be stronger. In a limited computational study that we do not include in the paper due to length considerations, we observed that our approaches perform better when the pricing problems were not restricted to producing minimal feasibility covers.

Fifth, due to the structure of our Benders approach's feasibility cuts (17), we can cache all feasibility cuts for a set of nodes. If we solve the TSP-GL for a new feasibility cover whose set of nodes was included in another already solved TSP-GL, we initialize our Benders approach with the corresponding feasibility cuts. Finally, we can terminate our Benders approach if the objective value of Problem 16 is greater than the current upper bound of our algorithm.

4.5. Computational analysis

Our analysis is fourfold. First, we analyze the computational efficiency of our algorithms. Second, we study the behavior of upper and lower bounds of our different approaches to understand what drives the complexity of our problem. Third, we analyze the benefits of our stochastic planning approach compared to a deterministic planning approach. Fourth, we aim to understand how the predefined service level and the feasibility frequency impact the design cost.

To do so, we first compare the bounds of our B&P, heuristic, and hybrid approach against a MIP benchmark, which solves the deterministic equivalent of our STSP-GL (Problem 2) with a standard solver. Second, we analyze the convergence of bounds of our three approaches against the same MIP benchmark. Third, we compare our stochastic approach against a deterministic approach on multiple instances, varying in graph size, service level, and feasibility frequency. Fourth, we show the difference in design cost on instances with varying service level and feasibility frequency.

4.5.1. Instances

We use the TSPGL_LIB data set to test our algorithms. The TSPGL_LIB instance set is an open source data set that builds on the TSPLIB data set. We focus on the *complete and uniform* demand distributions in which the demand matrix is complete and demand d_s^{hk} is generated uniformly on a given interval. For an in-depth explanation of the instance generation, we refer to Errico et al. (2017). For analysis (i), we solve instances on complete graphs with 17, 21, 29, 42, 51, 70, and 76 nodes to analyze our algorithms on different graph sizes and analyze our approaches on instances with 20 and 50 scenarios for $\theta = 0.95$ and $\rho = 0.05$ as well as $\theta = 0.9$ and $\rho = 0.1$. We generate five instances considering random seeds for each network dimension, scenario size, and service/feasibility level. Accordingly, this TSPGL_LIB-based instance set consists of 140 different instances. Furthermore, we set $\alpha = 0.25$. For analysis (ii), we focus on one representative example of bound convergence on a graph with 51 nodes for 20 and 50 scenarios. Other graphs from the TSPGL_LIB data set show similar results to those displayed here. For analysis (iii), we solve instances on complete graphs with 17, 21, 29, 42, 51, and 70 with 20 scenarios for $\theta \in [0.05, 0.1]$ and $\rho \in [0.9, 0.95]$. We again generate five instances considering random seeds for each network dimension, service, and feasibility level. Accordingly, this TSPGL_LIB-based instance set consists of 120 different instances. For analysis (iv), we again focus on one representative example with 51 nodes and 50 scenarios to analyze the impact of service and feasibility level on the design cost. In this context, we solve the STSP-GL for $\theta \in [0.8, 0.85, 0.9, 0.95, 1.0]$ and $\rho \in [0.0, 0.05, 0.1, 0.15, 0.2]$. Other graphs from the TSPGL_LIB data set again show similar results to those displayed here.

4.5.2. Computational setting

We set the maximum runtime of all algorithms to 60 minutes. In our B&P and hybrid approach, we limit the computation time for each RMP to 40 minutes and for each pricing problem to 20 minutes. Furthermore, for fair comparison, we terminate the RMP and pricing problem when reaching an optimality gap of two percent. This is necessary, as it may happen that both the RMP and pricing problem cannot find an optimal solution within the maximum computation time. Furthermore, we terminate all algorithms except our heuristic if an optimality gap of two percent is reached.

All our experiments have been conducted on a standard desktop computer equipped with two `Intel(R) Core(TM) i9-9900, 3.1 GHz CPU` and a total of `20 GB` of `RAM`, running `Ubuntu 20.04`. We have implemented all algorithms in `Python (3.8.11)` using `Gurobi 9.5`. Our source code can be found on https://github.com/tumBAIS/STSP-GL.

4.5.3. Computational effectiveness of proposed methods

To understand whether the proposed methods are computationally efficient, we benchmark them against a MIP solver solving Problem 2. We do this by aggregating results on instances according to different parameter values. Namely, the number of scenarios, service level, and feasibility frequency.

Tables 1 - 4 show average upper bounds (UB), lower bounds (LB), and optimality gaps (Gap) for the MIP benchmark, our B&P and hybrid approaches. Additionally, it shows the number of instances for which no primal feasible solution could be found (No UB) for the MIP benchmark and the upper bounds (UB) for our heuristic approach. All data displays the average over five randomly generated instances with a maximum computation time of one hour. We also tested our algorithms with a maximum computation time of two hours, but all our three algorithms and the MIP benchmark rarely improved upper or lower bounds in this additional hour.

Tables 1 - 4 show that for smaller instances with less than 50 nodes, the upper bounds of our B&P, heuristic, and hybrid approaches are nearly always 0-3% better than our MIP benchmark. The only outliers are the instances with 42 nodes in Table 4, in which the upper bounds of our B&P and hybrid approaches are 13% and of our heuristic are 8% better. We also see that our B&P and hybrid approaches find good lower bounds for these instances even though they are nearly always 2-3% worse than the lower bounds of our MIP benchmark. The only large outliers are the instances with 17 nodes in Table 4, in which the lower bounds of our B&P and hybrid approaches are 10% worse than our MIP benchmark. For larger instances with more than 50 nodes, our B&P and hybrid approaches find much better upper bounds than our MIP benchmark ranging from improvements between 10-44% in the instances with 51 and 76 nodes and improvements between 74-85% in instances with 70 nodes.

Furthermore, our B&P and hybrid approaches find much better lower bounds than our MIP benchmark for the two largest instances with 70 and 76 nodes, with improvements ranging between 8-55% compared to our MIP benchmark, but cannot close the optimality gap of 5-34%. Additionally, our MIP benchmark cannot find feasible solutions for at least three of the five problems of our largest instance with 76 nodes in all settings.

If we compare our different approaches over the different instance settings, we see that either our B&P or our hybrid approach performs better than our heuristic approach on all problems except the ones with 76 nodes, $\theta = 0.95$, $\rho = 0.05$ and 50 scenarios. Furthermore, we see that for most problems, our B&P and hybrid approaches perform similarly in terms of upper and lower bounds. The main difference in upper and lower bounds is in Table 3, where our hybrid approach finds worse lower bounds than our B&P approach and the fact that our hybrid approach finds better upper bounds than our B&P approach for the largest instances with 76 nodes in all tables. One explanation for the latter observation is that our hybrid approach finds more minimal feasibility covers it can evaluate

Table 1 TSPGL_LIB-based instances with 20 scenarios $\theta = 0.95$, $\rho = 0.05$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	20	3231	3187	1.38	0	3230	3230	0.00	3230	3230	0.00	3230
21	20	4258	4239	0.45	0	4253	4215	0.91	4252	4218	0.82	4252
29	20	2531	2489	1.67	0	2524	2412	4.67	2524	2426	4.05	2523
42	20	1046	985	6.29	0	1036	990	4.99	1036	975	6.42	1050
51	20	668	566	18.28	0	591	560	5.59	591	560	5.65	613
70	20	5086	686	637.77	0	925	885	4.89	927	821	12.95	988
76	20	864	561	51.18	3	720	671	7.35	718	667	7.70	778

Table 2 TSPGL_LIB-based instances with 50 scenarios $\theta = 0.95$, $\rho = 0.05$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	50	3279	3224	1.72	0	3276	3169	3.46	3276	3276	0.00	3276
21	50	4259	4224	0.83	0	4257	4205	1.24	4255	4225	0.73	4255
29	50	2536	2495	1.64	0	2530	2407	5.11	2526	2399	5.33	2530
42	50	1079	1024	5.42	0	1075	964	11.58	1071	978	9.55	1073
51	50	694	596	16.36	0	615	598	2.72	615	600	2.60	629
70	50	6079	692	779.12	0	962	885	8.90	966	866	11.57	992
76	50	933	451	63.18	4	828	702	18.06	784	697	12.55	778

Table 3 TSPGL_LIB-based instances with 20 scenarios $\theta = 0.9$, $\rho = 0.1$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	20	3132	3096	1.16	0	3130	2896	8.31	3131	2955	6.34	3203
21	20	3945	3904	1.03	0	3937	3706	6.33	3937	3763	4.78	3987
29	20	2166	2129	1.75	0	2167	2068	4.85	2167	2047	5.86	2238
42	20	983	808	21.74	0	947	947	0.00	945	845	12.74	981
51	20	582	456	27.83	0	527	484	9.37	520	467	11.66	581
70	20	3154	516	511.39	0	827	671	24.71	836	630	32.71	953
76	20	1123	403	171.74	3	631	576	10.04	630	539	16.88	738

Table 4 TSPGL_LIB-based instances with 50 scenarios $\theta = 0.9$, $\rho = 0.1$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	50	3269	3223	1.43	0	3268	2892	13.03	3268	2913	12.18	3277
21	50	3967	3905	1.59	0	3963	3786	4.80	3964	3792	4.64	4007
29	50	2191	2152	1.82	0	2200	2098	4.93	2203	2092	5.35	2293
42	50	1109	806	36.75	1	962	962	0.00	962	904	6.97	1014
51	50	689	471	46.55	0	552	506	9.33	554	492	12.56	595
70	50	6081	322	243816.66	0	886	672	31.68	896	671	33.49	955
76	50	832	255	97.54	4	785	571	37.80	742	567	30.97	744

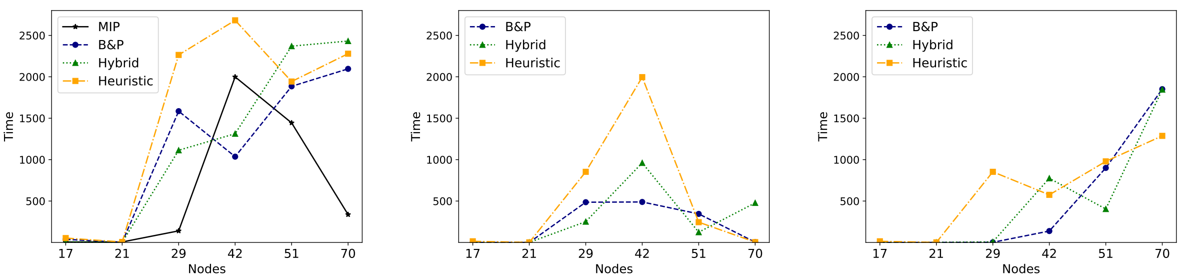
before solving the corresponding TSP-GL of one of them. As the instances grow in size, solving the corresponding TSP-GL becomes computationally expensive, i.e., determining the upper and lower bounds of more minimal feasibility covers leads to a higher probability of choosing a good minimal feasibility cover to evaluate.

Furthermore, we see that the number of scenarios as well as the reduction of the service level and feasibility frequency drives complexity, i.e., leads to worse optimality gaps in our B&P and hybrid approaches. The reason for that is, that with a growing number of scenarios as well as shrinking service level and feasibility frequency, the number of minimal feasibility covers grows significantly and more TSP-GLs have to be solved.

Figure 6 shows average computation times until the final upper bound is found (a), average computation times until a better upper bound than the best MIP bound is found in our B&P, hybrid, and heuristic approaches (b), and average computation times until an upper bound is found that is not more than 1% worse than the final upper bound (c) for five instances with 20 scenarios with $\theta = 0.95$ and $\rho = 0.05$. The results for instances with 50 scenarios or $\theta = 0.9$ and $\rho = 0.1$ are similar to those shown here. We do not add the results for instances with 76 nodes, as the MIP does not find solutions for most of them. We see that our B&P approach, on average, finds better upper bounds than our MIP in under 500 seconds for all instances. Additionally, we see that there may be a significant difference between finding an upper bound that is within one percent of the final upper bound and finding the final upper bound, e.g., instances with 29 nodes with 1585 seconds (Figure 6a) versus 2 seconds (Figure 6c). We observe similar behavior for our hybrid and heuristic approach. Additionally, we see that except for the largest instances with 70 nodes, either our B&P or our hybrid approaches find better upper bounds than the MIP faster than our heuristic approach.

4.5.4. Convergence of bounds

In this section, we analyze the convergence of the upper and lower bounds of our B&P, heuristic, and hybrid approaches with our MIP benchmark on the TSPGL_LIB dataset. Note that our heuristic has no lower bound. Figure 7 shows the average convergence of upper and lower bounds for instances



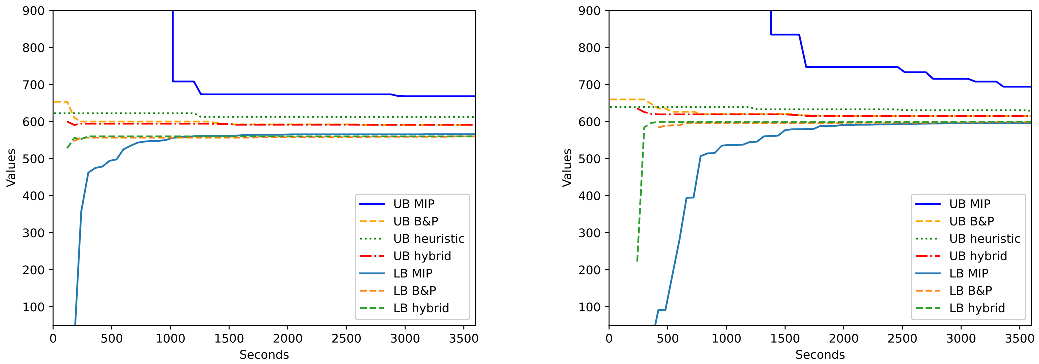
(a) Computation times until final upper bound

(b) Computation times until better upper bound than best MIP solution

(c) Computation times until upper bound that is not more than 1% worse than final upper bound

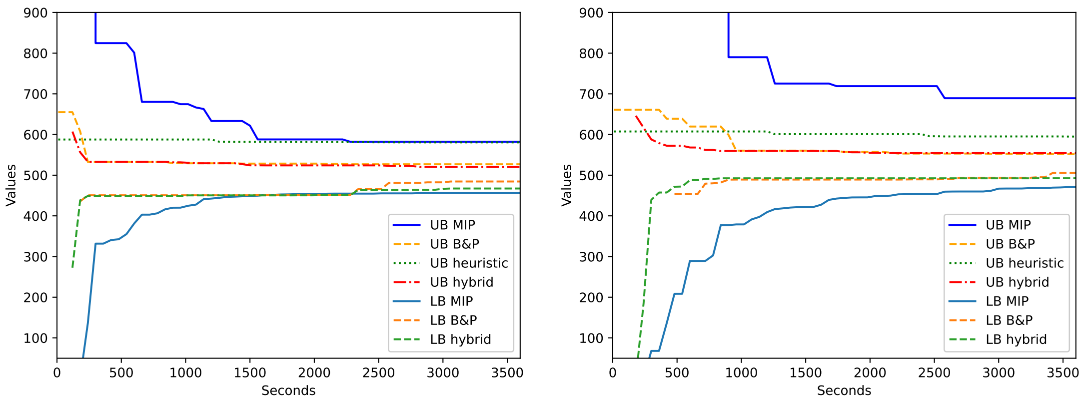
Figure 6 **Computation times of TSPGL_LIB-based instances with 20 scenarios** $\theta = 0.95$, $\rho = 0.05$

with 51 nodes and a service level of $\theta = 0.95$ and $\rho = 0.05$. We see that our B&P and hybrid approaches find better upper bounds faster than our MIP benchmark for instances with 20 scenarios over the entire computation time and quickly find better upper bounds than our heuristic approach. Our heuristic approach also finds better upper bounds than our MIP benchmark. We also see that all methods ultimately converge to similar lower bounds. For instances with 50 scenarios, we again find upper bounds with our B&P and hybrid approaches much faster than our MIP benchmark. Our heuristic approach finds good upper bounds even quicker than our B&P and hybrid approaches but cannot improve significantly over the computation time. Furthermore, we see that with more scenarios, our B&P and hybrid approaches find upper bounds much faster than our MIP benchmark but cannot close the optimality gap.



(a) Convergence of upper and lower bound for 51 nodes and 20 scenarios (b) Convergence of upper and lower bound for 51 nodes and 50 scenarios

Figure 7 Average bound convergence in instances with 20 and 50 scenarios for $\theta = 0.95$, $\rho = 0.05$



(a) Convergence of upper and lower bound for 51 nodes and 20 scenarios (b) Convergence of upper and lower bound for 51 nodes and 50 scenarios

Figure 8 Average bound convergence instances with 20 and 50 scenarios for $\theta = 0.9$, $\rho = 0.1$

Figure 8 shows the average convergence of upper and lower bounds for instances with 51 nodes and a service level of $\theta = 0.9$ and $\rho = 0.1$. We see that for 20 scenarios, our B&P and hybrid approaches find better upper bounds faster than our MIP benchmark. Similar to the higher service level, our heuristic approach finds upper bounds fast but cannot improve much during the rest of the run. Still, contrary to the higher service level instances with 20 scenarios, our MIP benchmark cannot find better bounds than our heuristic approach. Looking at the lower bounds of our B&P and hybrid approaches compared to our MIP benchmark, we see that for a lower service level, our B&P and hybrid approaches find equally good lower bounds to our MIP benchmark. For instances with 50 scenarios, our MIP benchmark takes much longer than our B&P, heuristic, and hybrid approaches to find feasible solutions. Additionally, our B&P and hybrid approaches find better lower bounds faster than our MIP benchmark. As our MIP benchmark especially struggles to find good upper bounds and the initial lower bounds of our B&P and hybrid approaches only show minor improvements with respect to computation time, none of the algorithms can close the optimality gap within one hour.

With this analysis of bound convergence, we see that our B&P, heuristic, and hybrid approaches are more robust against an increase in scenarios than our MIP benchmark and can find feasible solutions of good quality much faster.

4.5.5. Value of Stochastic Solution

It is common for solutions determined by deterministic models to derive better objective values than solutions determined by stochastic models when we evaluate them in a deterministic setting. At the same time, the deterministic solutions are often infeasible or derive worse objective values than stochastic solutions when we evaluate them in a stochastic setting. In this section, we show that this property also holds for the STSP-GL. Table 5 shows the comparison between a deterministic approach and our stochastic approach. Specifically, we compare our stochastic approach to the deterministic setting in which we solve the STSP-GL for only one scenario in which we calculate the mean demand for each request over all original scenarios. In this context, we test the two different approaches on different instances with $\theta \in [0.95, 0.9]$, $\rho \in [0.05, 0.1]$, and 20 scenarios. We solve five instances for each individual configuration. Table 5 shows the design cost, percentage of nodes in the tour ($\bar{N}$), average demand of served demand ($\bar{d}$), average percentage of scenarios in which the service level is not fulfilled ($\bar{\rho}$), and the number of deterministic solutions which are infeasible in our stochastic setting (Inf.).

We see that the deterministic approach finds tours with much lower design costs than our stochastic approach as, on average, the tours include fewer nodes. At the same time, as is expected for deterministic approaches in a stochastic setting, we see that for all instances, the solutions of the

Table 5 Comparison between deterministic and stochastic approach

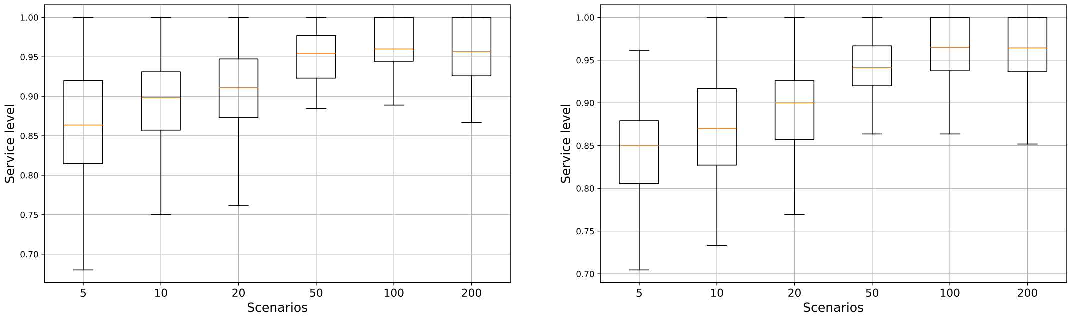
Instance			Deterministic approach					Stochastic approach			
Nodes	θ	ρ	Design cost	$\bar{N}$	$\bar{d}$	$\bar{\rho}$	Inf.	Design cost	$\bar{N}$	$\bar{d}$	$\bar{\rho}$
17	0.95	0.05	2846.1	0.91	0.96	0.31	5 / 5	3083.7	0.99	0.97	0.05
21	0.95	0.05	3714.3	0.90	0.95	0.43	5 / 5	4060.5	1.00	0.98	0.05
29	0.95	0.05	2077.5	0.90	0.95	0.45	5 / 5	2464.8	0.99	0.98	0.04
42	0.95	0.05	937.8	0.90	0.95	0.41	5 / 5	989.4	0.93	0.95	0.05
51	0.95	0.05	543.6	0.85	0.95	0.42	5 / 5	585.3	0.90	0.95	0.05
70	0.95	0.05	888.0	0.89	0.95	0.47	5 / 5	916.2	0.92	0.95	0.05
17	0.95	0.10	2846.1	0.91	0.96	0.31	5 / 5	3083.7	0.99	0.94	0.10
21	0.95	0.10	3714.3	0.90	0.95	0.43	5 / 5	4060.5	1.00	0.96	0.10
29	0.95	0.10	2077.5	0.90	0.95	0.45	5 / 5	2365.5	0.97	0.96	0.10
42	0.95	0.10	937.8	0.90	0.95	0.41	5 / 5	983.1	0.92	0.94	0.10
51	0.95	0.10	540.6	0.85	0.95	0.42	5 / 5	573.3	0.89	0.94	0.10
70	0.95	0.10	888.0	0.89	0.95	0.47	5 / 5	911.4	0.91	0.93	0.10
17	0.90	0.05	2611.8	0.85	0.91	0.42	5 / 5	3018.3	0.96	0.95	0.05
21	0.90	0.05	3380.4	0.86	0.90	0.41	5 / 5	3908.1	0.96	0.93	0.05
29	0.90	0.05	1812.3	0.83	0.90	0.44	5 / 5	2114.1	0.90	0.91	0.05
42	0.90	0.05	876.0	0.80	0.90	0.43	5 / 5	924.0	0.85	0.91	0.05
51	0.90	0.05	485.7	0.76	0.90	0.43	5 / 5	520.8	0.80	0.90	0.05
70	0.90	0.05	797.4	0.77	0.90	0.44	5 / 5	828.0	0.81	0.90	0.05
17	0.90	0.10	2611.8	0.85	0.91	0.42	5 / 5	2999.1	0.95	0.92	0.10
21	0.90	0.10	3380.4	0.86	0.90	0.41	5 / 5	3762.0	0.92	0.91	0.10
29	0.90	0.10	1812.3	0.83	0.90	0.44	5 / 5	2069.4	0.87	0.89	0.10
42	0.90	0.10	876.0	0.80	0.90	0.43	5 / 5	904.5	0.84	0.88	0.10
51	0.90	0.10	486.9	0.76	0.90	0.43	5 / 5	512.4	0.78	0.89	0.10
70	0.90	0.10	792.6	0.77	0.90	0.41	5 / 5	814.5	0.80	0.88	0.10

deterministic approach can not fulfill the service level for more than 30 percent of all scenarios. This makes all deterministic solutions infeasible for our stochastic setting, as for them to be feasible, $\bar{\rho}$ would need to be smaller than ρ, i.e., 0.05 and 0.1. Furthermore, we see that the percentage of served demand ($\bar{d}$) is often smaller in the deterministic solutions compared to our stochastic approaches. In this context, it is interesting that $\bar{d}$ is significantly higher than $\theta * (1 - \rho)$ which we expected in the stochastic approach. This phenomenon suggests that to fulfill our feasibility frequency, we have to serve more demand than required by our service level. Furthermore, except for the instances with 29 nodes, $\theta = 0.95$ and $\rho = 0.05$, $\bar{\rho}$ is as expected equal to $(1 - \rho)$ for all stochastic solutions.

Summarizing, our stochastic approach accounts for uncertain passenger demand by including more nodes in the tour, increasing the design cost. Additionally, significantly more demand is served in our stochastic solutions compared to the deterministic solutions.

4.5.6. Analysis of the number of chosen scenarios

Our model utilizes a set of scenarios to approximate the demand distribution. In this context, it remains a central question how many scenarios are necessary to approximate the demand distribution accurately such that the resulting route remains feasible in an out-of-sample comparison. To



(a) Service levels for instance with 42 nodes (b) Service levels for instance with 51 nodes

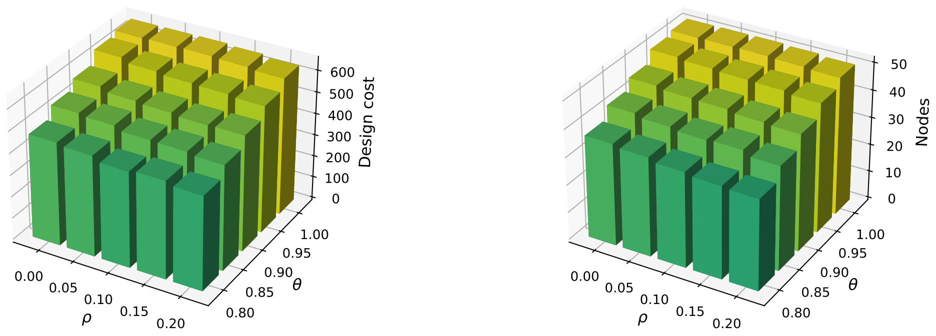
Figure 9 Service levels of 100 new instances on determined route for $\theta = 0.9$ and $\rho = 0.05$

answer this question, we solve instances with 5, 10, 20, 50, 100, and 200 scenarios with our B&P approach. Furthermore, we generate 100 new demand-scenarios and compute the service level based on the determined route. Figure 9 shows the service levels for these new scenarios for instances with 42 and 51 nodes with $\theta = 0.9$ and $\rho = 0.05$. Our assumption is that with sufficiently many scenarios to approximate the demand distribution, 95 demand-scenarios should have a service level higher than 0.9. While routes determined with 50 scenarios are just barely under the threshold of 95 demand-scenarios with a service level of at least 0.9, we see that the worst service levels of demand-scenarios can be higher than with 100 and 200 scenarios. Other instance configurations yield similar results.

4.5.7. Analysis of service and feasibility level impact on design cost

Our model presumes a service level, i.e., the market share served and the frequency with which it is served. To understand which drives costs, we aggregate results on instances according to different parameter values for the service level θ and the frequency of meeting it $(1-\rho)$.

Figure 10 shows the difference in design costs and number of nodes in the tour for service levels $\theta \in [0.8, 0.85, 0.9, 0.95, 1.0]$ and feasibility levels $\rho \in [0.0, 0.05, 0.1, 0.15, 0.2]$ on the TSPGL_LIB instance with 51 nodes with 50 scenarios. We see that the design cost of a route depends more on the desired market size θ than on the frequency of meeting it $(1-\rho)$, i.e., reducing θ by 0.05 on average decreases design costs by 8.5% while reducing $1-\rho$ by 0.05 on average only reduces design costs by 2%. We see a similar reduction in the number of nodes in the tour. Explicitly, reducing θ by 0.05 on average decreases the number of nodes in the tour by 9% while reducing $1-\rho$ by 0.05 on average only reduces the number of nodes in the tour by 2%. This is not surprising as the number of nodes in a tour highly correlates to the design costs of the tour. Consequently, a DAS operator aiming to reduce the design costs in the tactical planning problem should focus on the desired service level instead of on the frequency of meeting it.



(a) Average design cost (b) Average number of nodes

Figure 10 Average design cost and number of nodes for varying θ and ρ values on instance with 51 nodes

4.6. Conclusion and future work

With this work, we introduced the STSP-GL, which can be used to find a general ordering of stops in the tactical planning of DAS under uncertain passenger demand. Furthermore, we proposed an algorithmic framework to determine near-optimal solutions. To do so, we modeled the STSP-GL as a stochastic program and determined its deterministic equivalent. This allowed us to utilize a B&P, hybrid, and heuristic approach, which decompose the STSP-GL into multiple TSP-GLs. We performed extensive computational experimentation to analyze how our different algorithms perform on different graph sizes, number of scenarios, service, and feasibility levels. For this purpose, we generated a large set of instances from the TSPGL_LIB dataset. We show that our B&P and hybrid approaches find better upper bounds than our MIP benchmark within 500 seconds for nearly all instance types but struggle with closing the optimality gap. Especially for larger graph sizes, our B&P and hybrid approaches improve the MIP benchmark's upper bounds by up to 85%. Furthermore, we showed that our heuristic approach finds feasible solutions within a couple of seconds but due to the size of the set of minimal feasibility covers is not able to significantly improve on the initial upper bounds during the remainder of the computation time. Additionally, we saw that the additional local search and exploration of our hybrid approach makes it more robust against higher numbers of scenarios compared to our B&P approach. Furthermore, we showed that even though the design costs of solutions derived through a deterministic approach are lower compared to solutions of a stochastic approach, they are not feasible in a stochastic setting, i.e., the stochastic approach includes more nodes in the tours to meet the demand with a higher probability. Finally, we showed that the impact of the service level on the design cost is higher than the impact of the feasibility level. Here, we are able to, on average, reduce the design costs by 8.5% when lowering the service level by 5% while only reducing the design costs by 2% when lowering the feasibility frequency by 5%.

Further research may expand on the current methodology of our heuristic to develop a more sophisticated hybrid approach to outperform our B&P and hybrid approaches. Additionally, our current distribution of requests is nearly uniform, i.e., the probability of request A having positive demand in a scenario is similar to request B having positive demand in a scenario. Investigating the impact of different request distributions on the design and routing costs remains an interesting avenue for future research.

Acknowledgments

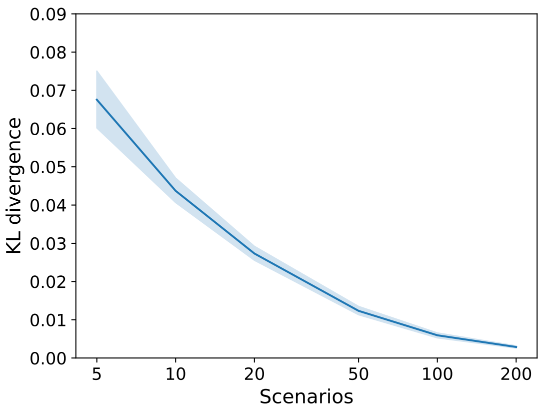
This work was supported by the German Federal Ministry of Education and Research under Grant 03ZU1105FA.

Appendix A Kullback-Leibler divergence

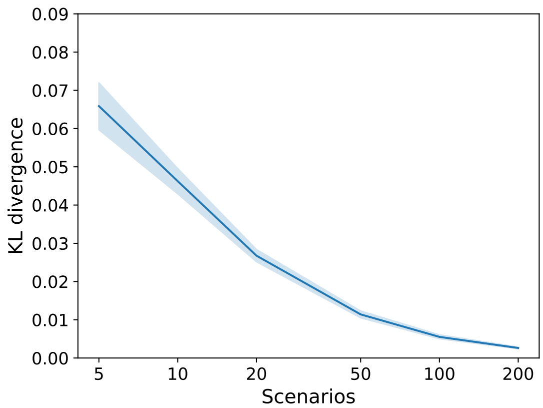
In this section, we analyze how well a set of scenarios represents the distribution of demands in the TSPGL_LIB instance set. To do so, we compute the Kullback–Leibler divergence (KLD) for instances with 17, 21, 29, 42, 51, and 70 nodes for 5, 10, 20, 50, 100, and 200 scenarios on five different seeds. Figure 11 shows the KLD for these instances with different numbers of scenarios. Here, we compare the average KLD over all request between the mean over all seeds of the sample-based and the original demand distribution. We see that on average over all instances with 50 scenarios, we are able to reduce the KLD by 83.4%, with 100 scenarios by 92.1%, and with 200 scenarios by 96.1% compared to the KLD for only five scenarios. As our MIP benchmark often does not find any feasible solutions for instances with more than 50 scenarios and our solutions found perform well for 50 scenarios in an out-of-sample comparison, we restrict the computational analysis of our results to a maximum of 50 scenarios.

Appendix B Sensitivity analysis on parameter α

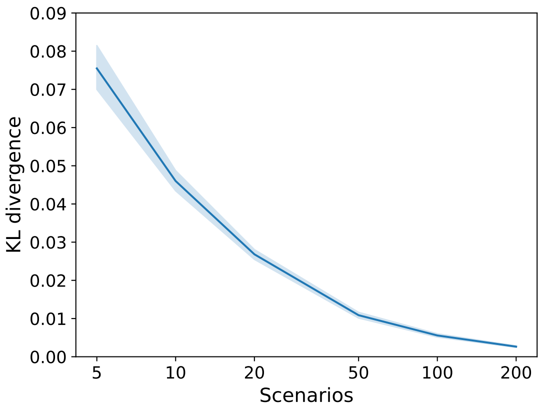
In this section, we show results for $\alpha \in [0.5, 0.75]$. Tables 6 - 9 ($\alpha = 0.5$) and Tables 10 - 13 ($\alpha = 0.75$) show that with an increasing value of α the optimality gap of our B&P and hybrid approaches get worse. This is mainly due to worse lower bounds. At the same time, our B&P and hybrid approaches consistently perform better than our MIP benchmark on nearly all instances. Interestingly, our heuristic consistently finds the best solutions for the largest instances. This is mainly due to the fact that for larger instances, generating feasibility covers through our column generation becomes increasingly difficult, and our heuristic is able to find more feasibility covers faster than our B&P and hybrid approaches. Additionally, due to the increased complexity of our TSP-GLs through an increase in parameter α, we are only able to solve fewer subproblems during our computation time. As our heuristic finds more feasibility covers before solving their corresponding TSP-GLs, it has a higher chance of generating a good feasibility cover whose upper bounds are already relatively good.



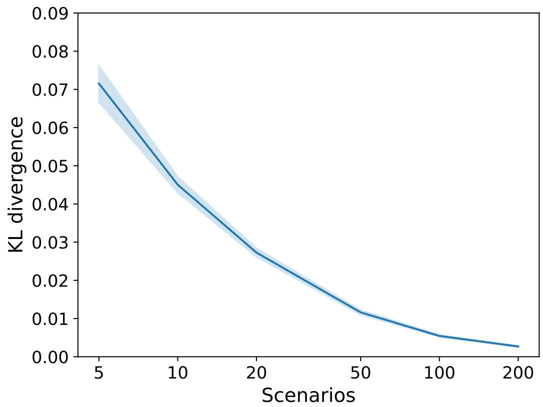
(a) KLD for 17 nodes



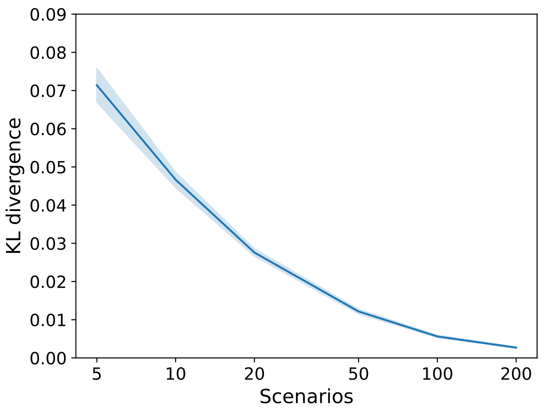
(b) KLD for 21 nodes



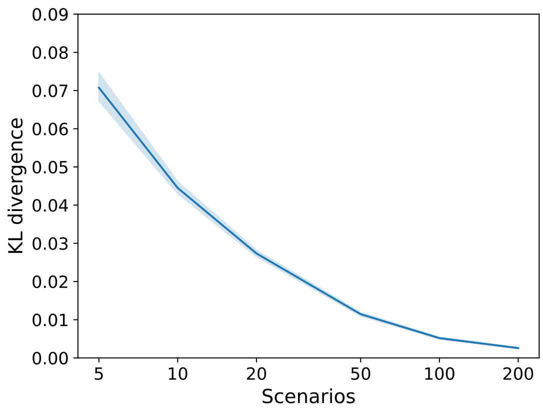
(c) KLD for 29 nodes



(d) KLD for 42 nodes



(e) KLD for 51 nodes



(f) KLD for 70 nodes

Figure 11 Kullback-Leiber divergence (KLD) between scenario-based and original demand distribution

Table 6 **TSPGL_LIB-based instances with 20 scenarios** $\theta = 0.95$, $\rho = 0.05$, $\alpha = 0.5$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	20	2346	2302	1.92	0	2346	2169	8.24	2344	2119	10.76	2346
21	20	3091	3034	1.88	0	3090	3077	0.44	3090	3090	0.00	3090
29	20	1835	1710	7.31	0	1711	1687	1.44	1785	1693	5.40	1712
42	20	771	706	9.32	0	730	658	10.99	733	657	11.66	746
51	20	470	387	21.38	0	398	386	3.25	406	386	5.23	410
70	20	3678	295	176355.41	0	619	578	7.18	619	581	6.54	664
76	20	746	392	98.14	2	538	462	16.57	530	463	14.29	520

Table 7 **TSPGL_LIB-based instances with 50 scenarios** $\theta = 0.95$, $\rho = 0.05$, $\alpha = 0.5$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	50	2383	2337	1.98	0	2374	2135	11.25	2372	2110	12.44	2375
21	50	3097	3044	1.76	0	3097	3060	1.23	3096	3086	0.34	3097
29	50	1847	1699	8.71	0	1708	1678	1.81	1849	1697	8.97	1706
42	50	834	720	15.88	0	746	673	10.98	744	693	7.37	745
51	50	468	409	14.36	0	412	410	0.47	437	414	5.66	423
70	50	4409	1	440825.07	0	677	606	11.73	690	611	12.80	666
76	50	1108	78	186.37	4	618	478	29.49	564	484	16.78	523

Table 8 **TSPGL_LIB-based instances with 20 scenarios** $\theta = 0.9$, $\rho = 0.1$, $\alpha = 0.5$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	20	2262	2222	1.80	0	2262	2007	12.65	2258	2044	10.54	2329
21	20	2859	2813	1.64	0	2859	2646	8.10	2860	2645	8.18	2894
29	20	1571	1521	3.29	0	1528	1421	7.54	1533	1424	7.66	1591
42	20	783	576	35.80	0	686	544	26.06	685	545	25.84	718
51	20	462	316	46.50	0	357	313	14.11	356	313	14.00	391
70	20	3691	146	264036.97	0	574	454	26.51	570	449	26.92	642
76	20	734	284	156.50	1	456	371	22.78	464	370	25.45	497

Table 9 **TSPGL_LIB-based instances with 50 scenarios** $\theta = 0.9$, $\rho = 0.1$, $\alpha = 0.5$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	50	2367	2321	2.02	0	2366	1994	18.70	2373	1994	19.02	2382
21	50	2883	2831	1.84	0	2880	2709	6.42	2880	2707	6.53	2916
29	50	1600	1501	6.61	0	1528	1435	6.49	1547	1433	8.01	1593
42	50	826	579	42.63	0	689	566	21.73	690	574	20.27	727
51	50	482	328	47.17	0	373	340	9.80	370	342	8.18	402
70	50	4414	1	441335.08	0	731	468	56.45	653	464	40.89	638
76	50	627	180	111.07	4	596	377	59.92	527	372	41.96	496

Table 10 **TSPGL_LIB-based instances with 20 scenarios** $\theta = 0.95$, $\rho = 0.05$, $\alpha = 0.75$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	20	1400	1357	3.17	0	1358	1265	7.45	1359	1233	10.29	1362
21	20	1921	1865	3.00	0	1868	1749	6.90	1866	1717	8.66	1870
29	20	1136	876	29.68	0	876	876	0.00	1150	956	20.31	880
42	20	532	385	38.18	0	388	384	0.85	476	390	22.10	394
51	20	303	200	51.50	0	202	202	0.00	245	210	16.88	209
70	20	1863	289	533.36	0	316	316	0.00	400	328	21.76	334
76	20	395	233	70.03	3	320	249	28.47	330	250	31.76	263

Table 11 **TSPGL_LIB-based instances with 50 scenarios** $\theta = 0.95$, $\rho = 0.05$, $\alpha = 0.75$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	50	1420	1348	5.34	0	1349	1248	8.06	1353	1260	7.39	1357
21	50	1934	1861	3.92	0	1866	1752	6.51	1867	1771	5.45	1865
29	50	1148	862	33.18	0	868	868	0.00	1163	962	20.86	869
42	50	579	387	49.61	0	390	390	0.00	489	403	21.42	390
51	50	329	204	61.27	0	208	208	0.00	256	226	13.21	214
70	50	2740	1	273887.60	0	398	338	17.63	417	343	21.53	334
76	50	542	217	147.31	3	389	258	51.43	351	249	42.56	265

Table 12 **TSPGL_LIB-based instances with 20 scenarios** $\theta = 0.9$, $\rho = 0.1$, $\alpha = 0.75$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	20	1353	1330	1.73	0	1331	1187	12.20	1333	1188	12.28	1362
21	20	1759	1724	2.02	0	1728	1554	11.17	1733	1561	11.03	1756
29	20	974	793	22.82	0	799	796	0.41	994	810	22.76	824
42	20	468	343	36.33	0	363	335	8.54	364	341	7.04	378
51	20	281	176	59.94	0	182	174	4.53	201	176	14.07	200
70	20	1838	177	55365.05	0	290	266	9.05	319	269	18.53	327
76	20	377	187	100.83	1	275	194	42.59	278	200	39.62	254

Table 13 **TSPGL_LIB-based instances with 50 scenarios** $\theta = 0.9$, $\rho = 0.1$, $\alpha = 0.75$

Instance		MIP				B&P			Hybrid			Heuristic
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	UB
17	50	1402	1352	3.70	0	1354	1209	12.04	1354	1211	11.83	1369
21	50	1788	1737	2.94	0	1738	1586	9.61	1738	1599	8.71	1757
29	50	1021	787	29.73	0	790	790	0.00	1035	809	27.97	818
42	50	505	345	46.59	0	364	347	4.90	364	349	4.45	386
51	50	310	181	71.56	0	190	189	0.67	234	188	24.33	201
70	50	2748	1	274652.63	0	425	268	59.62	407	266	52.99	326
76	50	511	168	210.90	4	375	186	105.52	313	188	68.38	254

Appendix C An alternative branching strategy

This section discusses the consequences of an alternative branching strategy. Specifically, instead of directly branching on cover variables χ_Q, we introduce branching constraints

$$\sum_{Q \in \mathcal{X'}_{\theta,\rho}} r_Q^{hk} \chi_Q \geq 1 \tag{19}$$

and

$$\sum_{Q \in \mathcal{X'}_{\theta,\rho}} r_Q^{hk} \chi_Q \leq 0 \tag{20}$$

for $(h,k) \in \bigcup_{Q \in \bar{\mathcal{X}}'_{\theta,\rho}} Q \setminus \bigcap_{Q \in \bar{\mathcal{X}}'_{\theta,\rho}} Q$ with $\bar{\mathcal{X}}'_{\theta,\rho} := \{Q \in \mathcal{X'}_{\theta,\rho} : 0 < \chi_Q < 1\}$. This strategy branches on the decision whether to serve request (h,k) (Constraint (19)) or not (Constraint (20)). Here (h,k) is not in at least one feasibility cover $Q \in \bar{\mathcal{X}}'_{\theta,\rho}$. Consequently, this leads to integer cover variables χ_Q. After determining integer cover variables, the resulting subproblems can again be solved via Benders decomposition to derive integer edge variables. To utilize column generation in the subproblems resulting from branching constraints (19) and (20), we have to add constraints $r^{hk} \geq 1$ or $r^{hk} \leq 0$ to pricing problem 11 according to the previous branching decisions and update the objective function (11a) of the pricing problem to.

$$\min_{l,r} \quad 2\sum_{i \in N} \iota_i l_i + \sum_{(h,k) \in D} r^{hk}(\epsilon_h^{hk} - \epsilon_k^{hk} + b^{hk}) - \beta \tag{21}$$

Here, we define b as

$$b^{hk} = \begin{cases} \text{dual variable of (19)} & \text{if } (h,k) \in B^+, \\ \text{dual variable of (20)} & \text{if } (h,k) \in B^-, \\ 0 & \text{else} \end{cases} \tag{22}$$

where B^+ and B^- are branching decisions on requests $(h,k) \in D$ resulting in constraints (19) and (20) respectively.

There are two major complexity drivers for Problem 6. First, determining all feasibility covers, even for only solving the root node of the problem, is hard. Second, in each iteration of our column generation, we have to determine optimal solutions for all passenger flow variables f_{ij}^{hk} for each arc and each request. By directly branching on the feasibility cover variables χ_Q, utilizing our upper and lower bounds for the resulting subproblems, and solving the subproblems via Benders decomposition, we tackle both of these complexity drivers. First, we only have to determine optimal passenger flow variables once for each generated feasibility cover, i.e., in the restricted master problem of our column generation. Second, through the upper and lower bounds of our subproblems, we cut parts of our branching tree early in our algorithm. Third, by utilizing Benders decomposition for our subproblems, we do not have to consider passenger flow variables in our subproblems.

If on the other hand, we branch via constraints (19) and (20), even though this would lead to a more balanced search tree, we face several problems. First, we can not generate all feasibility covers to solve the root node to optimality. Second, in each of the following branching steps, we have to determine optimal passenger flow variables. This may happen up to $2^{|Edges|}$ many times and is very costly. Second, by not directly branching on cover variables χ_Q, we can only utilize our upper and lower bounds of the TSP-GL subproblems after determining integer cover variables late in the search tree. Since we can only determine a subset of feasibility covers in the root node of our branch tree, determining whether a feasibility cover leads to a good solution early in the branch tree and, based on this information, possibly generating additional feasibility covers early is essential for finding good primal solutions. Table 14 shows the results for the alternative branching strategy (Alternative) for the instances of Tables 1-4 compared to our chosen branching strategy (B&P) where we branch directly on the feasibility covers variables χ_Q. We see that even though for small instances the alternative branching strategy leads to optimal solutions, it rarely finds integral solutions for larger instances.

For these reasons, in this work we only focus on directly branching on the feasibility covers variables χ_Q.

Table 14 Instance of Tables 1-4 solved with alternative branching

		$\theta = 0.95,\ \rho = 0.05$							$\theta = 0.9,\ \rho = 0.1$						
Instance		Alternative				B&P			Alternative				B&P		
Nodes	$\|S\|$	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]	UB	LB	Gap [%]	No UB	UB	LB	Gap [%]
17	20	3230	3230	0.0	-	3230	3230	0.0	3130	3130	0.0	-	3130	2896	8.31
17	50	3276	3276	0.0	-	3276	3169	3.46	3268	3268	0.0	-	3268	2892	13.03
21	20	4253	4253	0.0	-	4253	4215	0.91	3937	3937	0.0	-	3937	3706	6.33
21	50	4256	4256	0.0	-	4257	4205	1.24	3963	3963	0.0	-	3963	3786	4.80
29	20	2530	2435	3.89	-	2524	2412	4.67	2215	2124	4.67	-	2167	2068	4.85
29	50	2526	2455	2.89	-	2530	2407	5.11	2291	2106	8.8	-	2200	2098	4.93
42	20	1047	954	9.81	-	1036	990	4.99	964	799	20.81	-	947	947	0.0
42	50	1083	994	8.97	2	1075	964	11.58	-	839	-	5	965	965	0.0
51	20	617	567	8.02	3	591	560	5.59	-	471	-	5	527	484	9.37
51	50	-	607	-	5	615	598	2.72	-	509	-	5	552	506	9.33
70	20	-	837	-	5	925	885	4.89	-	671	-	5	827	671	24.71
70	50	-	876	-	5	962	885	8.90	-	672	-	5	886	672	31.68
76	20	-	678	-	5	720	671	7.35	-	542	-	5	631	576	10.04
76	50	-	708	-	5	828	702	18.08	-	543	-	5	785	571	37.80

5 Operational route planning under uncertainty for Demand Adaptive Systems

With an increasing need for more flexible mobility services, we consider an operational problem arising in the planning of Demand Adaptive Systems. Motivated by the decision of whether to accept or reject passenger requests in real time in a Demand Adaptive Systems, we introduce the operational route planning problem of Demand Adaptive Systems. To this end, we propose an algorithmic framework that allows an operator to plan which passengers to serve in a Demand Adaptive Systems in real-time. To do so, we model the operational route planning problem as a Markov decision process and utilize a rolling horizon approach to approximate the Markov decision process via a two-stage stochastic program in each timestep to decide on the next action. Furthermore, we determine the deterministic equivalent of our approximation through sample-based approximation. This allows us to decompose the deterministic equivalent of our two-stage stochastic program into several full information planning problems, which can be solved in parallel efficiently. Additionally, we propose a consensus-based heuristic and a myopic approach. We perform extensive computational experiments based on real-world data provided to us by the public transportation provider of Munich, Germany. We show that our exact decomposition yields the best results in under five seconds, and our heuristic approach reduces the serial computation time by 17 - 57% compared to our exact decomposition, with a solution quality decline of less than one percent. From a managerial perspective, we show that by switching a fixed-line bus route to a Demand Adaptive Systems, an operator can increase profit by up to 49% and the number of served passengers by up to 35% while only increasing the travel distance of the bus by 14%. Furthermore, we show that an operator can reduce their cost per passenger by 43 - 51% by increasing route flexibility and that incentivizing passengers to walk slightly longer distances reduces the cost per passenger by 83-85%. This chapter is based on an article published as: Lienkamp, B., Hewitt, M., Parmentier, A., & Schiffer, M. (2025). Operational route planning under uncertainty for Demand Adaptive Systems. preprint arXiv:2503.07812

5.1 Introduction

One of the development goals outlined in the United Nations' Agenda 2030 for cities (UN2 2015) is to preserve sustainable, affordable, and accessible mobility. Public transportation systems are essential for achieving this goal as they provide access to essential services, education, and employment opportunities for individuals with varying socio-economic backgrounds. Public transportation systems enhance land use efficiency and mitigate urban sprawl, fostering the development of interconnected and compact communities, and reinforcing social bonds. Building and reinforcing extensive public transportation networks enables cities to enhance organizational efficiency, increase ecological sustainability, and broaden societal inclusivity.

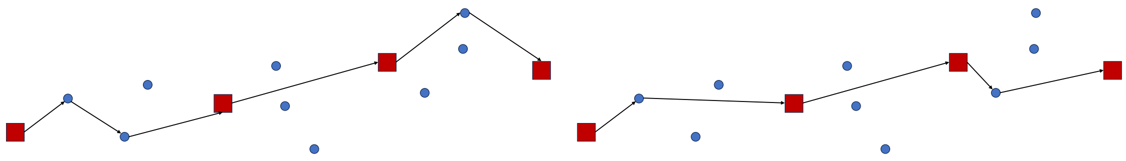
In a dense urban area, traditional fixed-line public transportation systems typically have high utilization, particularly during daytime hours, due to a concentrated population that does not have its own personal transportation. In contrast, in sparse rural areas, these fixed-line systems, usually operated by busses, exhibit lower utilization in general, reflecting the reduced population density and corresponding lower demand for such services (VDV 2020). This contrast between urban and rural areas shows that the efficiency of public bus transportation systems with fixed routes and schedules depends on the respective operating environment.

Demand Responsive Systems, originally introduced as Dial-a-Ride, offered door-to-door services, especially catering to users with reduced mobility (Wilson et al. 1971; Ioachim et al. 1995; Toth & Vigo 1996). These systems constitute an opportunity to increase efficiency in low-demand scenarios because they do not operate on fixed bus routes and timetables. Accordingly, they provide increased flexibility and can adjust to varying individual transportation needs, allowing for more personalized services while still enabling resource sharing and, thus, a high utilization. However, a drawback of Dial-a-Ride lies in the higher operational costs associated with providing customized services and accommodating specific travel requests (Ho et al. 2018). Moreover, the real-time nature of pickups and drop-offs, involving significant route changes during operation, can result in unpredictable travel times for passengers, making it less suitable for individuals with time-sensitive commitments.

Demand Adaptive Systems (DASs), as proposed by Malucelli et al. (1999) and Quadrifoglio et al. (2007), integrate Demand Responsive Systems into the framework of scheduled bus transportation. In the context of DASs, a designated bus line follows a conventional transit route with predetermined compulsory stops, adhering to a fixed schedule with specified time windows for departures at those compulsory stops. In addition to these compulsory stops, passengers have the flexibility to make requests at predefined non-compulsory stops, prompting the system to accommodate detours in the vehicle's route (see Figure 1). In the following, we refer to these non-compulsory stops as optional stops.

The fundamental premise of DASs is that the consistency and predictability of public transportation are necessary features. These attributes enable passengers to plan their journeys, allow integration with other modes of transportation, and provide accessibility to the service without necessitating advanced bookings as is the case in Demand Responsive Systems. Here, the reliability inherent in a regular schedule is deemed a crucial asset for public transit, enhancing the overall user

Figure 1: A DAS line with compulsory stops (red) and optional stops (blue)



(a) Route variant 1 **(b) Route variant 2**

experience and optimizing the utilization of transportation infrastructure. Given that premise, a DAS introduces a more personalized transportation system, capable of serving a larger proportion of passengers compared to traditional fixed bus lines, all at a reduced cost compared to on-demand door-to-door services (Errico et al. 2013). This combination of flexibility and predictability effectively combines the strengths of conventional fixed bus lines and Dial-a-Ride, mitigating their respective limitations.

Effectively planning the operations of a DAS is a challenging task, given the need to find a balance between the fixed and flexible attributes of traditional and on-demand systems. Additionally, the long-term viability of many public transportation systems depends at least in part on their revenue (Kustar et al. 2024). This planning process entails two phases: a tactical service design phase, wherein bus routes with compulsory and optional stops, as well as schedules, are initially established, and a subsequent operational phase, wherein continuous adjustments to bus routes and schedules in response to user requests are performed. The complexities associated with planning DASs operations have been investigated in existing literature, and Errico et al. (2013) provide a comprehensive review of the methodological aspects involved in this task.

This paper focuses on the operational phase. More precisely, the problem of determining in real time whether to accept or reject passenger requests of a DAS to maximize the operators' profit, which is pivotal for operating a DAS economically. This problem setting is similar to a dynamic vehicle routing problem (Pillac et al. 2013) with the distinction, that we consider part of the route, i.e., the compulsory stops, to be fixed with a given schedule. Crainic et al. (2005) analyze this problem under the assumption that the operator has full information, i.e., all passenger requests are known a priori to the acceptance or rejection decision, and develop and compare different meta-heuristics to solve that problem. As this assumption entails long response times towards the passengers requesting service, we explicitly consider that passenger requests occur over time and are not known in advance. This leads to a dynamic and stochastic optimization problem and, thus, short response times whether a passenger request is accepted or rejected.

In conclusion, the planning process of the operations of semi-flexible transit systems and DASs has been studied from tactical and operational perspectives under several system assumptions utilizing simulations, continuous approximations, and combinatorial optimization. So far, operational DAS problems have only been considered under full demand information in Crainic et al. (2005), which does not allow an operator to immediately give feedback to a passenger about the acceptance or rejection decision. Relatedly, the literature on dynamic vehicle routing problems has not considered the setting in which customers are inserted into a pre-determined route and schedule.

To close the research gap outlined above, we propose an algorithmic framework to solve the operational route planning problem of a DAS. To this end, we model the operational route planning problem under uncertainty as a Markov decision process (MDP) and utilize a rolling horizon approach to approximate the MDP via a two-stage stochastic program in each timestep to decide whether to accept or reject a passenger. We leverage a sample-based approximation of that stochastic program to formulate multiple deterministic problems that can be solved in parallel. Additionally, we propose a heuristic that leverages a consensus-based metric similar to Pisinger (2024) to determine which requests to accept.

We utilize this algorithmic framework for a real-world case study in the city of Munich and show that our parallelized decomposition approach outperforms our heuristic and myopic approaches in terms of solution quality. Additionally, we see that our exact decomposition yields the best results in under five seconds, and our heuristic approach can reduce the serial computation time by half compared to our exact decomposition, with a solution quality decline of less than one percent. We also show that even considering a limited number of future demand scenarios suffices to obtain a good solution quality, especially in high flexibility scenarios. From a managerial perspective, we show that by switching a fixed-line bus route to a DAS, an operator can increase revenue by up to 49% and the number of served passengers by up to 35% while only increasing the travel distance of the bus by 14%. Lastly, we show that even though an operator can reduce their cost per passenger by 43 - 51% by increasing route flexibility, incentivizing passengers to walk 250 meters instead of only 100 meters to their boarding and alighting stops reduces this cost by 83-85%.

Contribution: Our contributions are threefold. First, we present a new model of a real-time operational planning problem for DASs that explicitly recognizes their dynamic and stochastic nature. Second, we propose multiple algorithms for solving instances of this model that produce high-quality solutions with some yielding solutions in just seconds with only a small degradation in solution quality. Third, we study the performance of a DAS, as compared to a fixed-line bus route, in multiple settings to illustrate the impacts of such systems on both transit system health and passenger experience.

Organization: The remainder of this paper is as follows. Section 5.2 reviews related literature. We specify our problem setting in Section 5.3 and develop our methodology in Section 5.4. In Section 5.5, we describe our instance generation and present our numerical studies. Section 5.6 concludes this paper by summarizing its main findings.

5.2 Literature Review

Semi-flexible transit systems, which include DAS, have been studied from several perspectives under multiple different assumptions. Errico et al. (2013) categorize the planning decisions necessary to operate a semi-flexible transit system into strategic, tactical, and operational.

Strategic planning decisions focus on the definition of the service area, the itineraries in terms of potential compulsory and optional stops, the level of service, i.e., frequencies and fleet size, and whether to use fixed bus routes, DASs, or another type of public transportation system. In this context, Daganzo (1984) compares the cost between fixed route transit and two Dial-a-Ride variants to determine which type of transportation system is better suited to serve a given service area. Quadrifoglio & Li (2009) analyze the critical demand density below which their demand-responsive policy outperforms a fixed-route policy, and Li & Quadrifoglio (2009) develop an analytical model which determines the optimal number of independently operated zones, the service area should be divided into, to balance customer service quality and vehicle operating costs. Pratelli & Schoen (2001) propose a mathematical formulation to define the service area of a semi-flexible line based on the total aggregated travel time of passengers spent on board during deviations from the original main route, the walking times, and the increase of waiting times at compulsory stops. Qiu et al. (2015) study a so-called demi-flexible operating policy of a feeder system in Zhengzhou City (China). This feeder system can deviate from its fixed route to offer door-to-door service. The demi-flexible operating policy allows rejected passengers to utilize curb-to-curb stops of accepted passengers to carry out their trips. The authors show that under certain assumptions, this demi-flexible policy offers advantages in terms of affordability and efficiency compared to fixed-route and fully flexible route policies. Viergutz & Schmidt (2019) perform a simulation-based study in a rural area in Germany and show that stop-based and door-to-door Demand Responsive Systems (DRS) are uneconomical in their rural service area because of their high operational costs. Bischoff et al. (2019) perform a simulation-based study in the city of Cottbus (Germany) to determine the fleet size of a stop-based and a door-to-door DRS and show that for autonomous vehicles, each of the two DRS services is significantly cheaper during peak hours compared to fixed-line policies.

Tactical planning decisions focus on (i) which compulsory and optional stops to include in the design of a semi-flexible line, (ii) in which sequence the compulsory stops should be visited, (iii) the tour segments, i.e., between which two compulsory stops can an optional stop be visited, and (iv) the design of a so-called master schedule, which determines the time-windows of the compulsory stops. Errico et al. (2013) categorize tactical planning according to the used solution method into continuous approximations and combinatorial optimization. As our methodology belongs to the later solution method, we only review combinatorial optimization approaches for DAS in this paragraph. Errico (2008) and later Errico et al. (2011) introduce the Single-line DAS Design Problem (SDDP), which assumes the set of stops to be served by the DAS line, the transportation demand, and the travel time between all stop pairs to be known. Based on this information, the SDDP determines the compulsory stops, their sequence, the DAS segments, and the master schedule of the compulsory stops. The authors propose two decompositions of the SDDP consisting of the selection of the compulsory stops, the General Minimum Latency Problem (GMLP), which is a sequencing problem of the compulsory and optional stops, and the master schedule problem. Errico et al. (2011) address the selection of the compulsory stops and compare the two decompositions. Errico et al. (2017) propose the traveling salesman problem with generalized latency (TSP-GL), a variant

of the GMLP where the objective function combines the vehicle routing cost and the passenger's travel times, and develop a Benders' decomposition approach to solve it efficiently. Lienkamp et al. (2023) build on this methodology and introduce the stochastic TSP-GL, incorporating stochastic passenger demand into the problem setting and solve it via a Branch-and-Price approach. They show that an operator can derive more robust solutions by considering stochastic passenger demand than in a deterministic setting. Crainic et al. (2012) proposes a mathematical description and a solution method based on probabilistic approximations to solve the master schedule problem.
Lastly, operational planning determines vehicle and driver schedules as well as real-time adjustments to the DAS route. There are two main operational problem categories: dynamic online and dynamic offline problems. Vansteenwegen et al. (2022) give an extensive overview of this categorization. Dynamic online problems assume that requests can occur during the planning horizon and after the bus starts its trip. Here, Pei et al. (2019) utilize a tabu search approach to reduce the number of visited stops in a fixed-bus line, maximizing the total system income, which is the income minus operating and passenger time costs. They show that during off-peak hours, their approach outperforms fixed-line operations. Quadrifoglio et al. (2007) propose an insertion heuristic for Mobility Allowance Shuttle Transit (MAST), a special case of semi-flexible transit systems, which is described in detail in Errico et al. (2013). They show that their approach can be used to automate a MAST in Los Angeles (USA) and that their heuristic results are close to optimal solutions.

Dynamic offline strategies assume that requests can only be issued before the bus starts its trip. In this context, Malucelli et al. (1999) introduce DAS with three operational policies differing in user requirements. They propose linear integer formulations for these policies, which they solve through Lagrangian approaches and column generation, and combine these with heuristic approaches. Malucelli et al. (2001) and Crainic et al. (2005) introduce a GRASP-based memory-enhanced constructive heuristic and a tabu search meta-heuristic to solve the operational problem of DAS under full information. Additionally, they combine the two heuristic approaches, resulting in multiple hybrid approaches, and show that for simple instances, a tabu search-based approach performs best. In contrast, a hybrid approach yields the best results for more complex instances. Qiu et al. (2014) study a MAST system in Los Angeles (USA) and show that their dynamic stop strategy can reduce user cost by up to 30%. We refer to Errico et al. (2013) and Vansteenwegen et al. (2022) for an in-depth overview of semi-flexible and demand-responsive systems.

More generally, the stochastic dynamic vehicle routing problem (SDVRP) is an extension of the vehicle routing problem where both uncertainty, e.g., random customer demands or travel times, and real-time changes are considered. There are four types of methods commonly applied to solve SDVRPs. Reoptimization approaches only use information that is currently available to the operator (Branchini et al. 2009; Gendreau et al. 2006). Policy function approximations aim to imitate effective decision-making in practice (Branke et al. 2005; Thomas & White III 2004). Cost function approximations learn the value of a policy via repeated simulations and training (Riley et al. 2020; Al-Kanj et al. 2020). Look-ahead algorithms sample stochastic information and derive a policy based on the sampled scenarios (Schilde et al. 2014; Azi et al. 2012). We refer to Soeffker et al. (2022) and Ulmer et al. (2020) for an in-depth overview of modeling stochastic dynamic vehicle routing problems.

In summary, semi-flexible transit systems and DAS have been studied from strategic, tactical, and operational perspectives under several system assumptions utilizing simulations, continuous approximations, and combinatorial optimization. So far, operational DAS route planning problems have only been considered in a static setting under full demand information in Crainic et al. (2005), which does not allow an operator to give feedback to a passenger about the acceptance or rejection decision in a dynamic way. Furthermore, stochastic dynamic vehicle routing problems do not consider fixed routes and schedules.

5.3 Problem description and mathematical formulations

A DAS route consists of *compulsory stops* at which the bus performing the route has to arrive at predefined *time windows*. We define a set of *optional stops* for each pair of consecutive compulsory stops that may be served by the vehicle when driving from one compulsory stop to the next. Travel times between all stops and service times at stops are known. Passengers seek transportation between pick-up and drop-off locations. To do so, they issue *requests*, which induce sets of pick-up and drop-off stops within a given walking distance from their corresponding pick-up and drop-off location determined by the operator. These pick-up and drop-off stops are subsets of all compulsory and optional stops. If we accept only passenger requests whose pick-up and drop-off stops consist of compulsory stops, the vehicle's route consists only of the compulsory stops, whose order is predetermined and can be inferred from the time windows at compulsory stops. If we accept a passenger request whose induced pick-up or drop-off stops contain an optional stop, the vehicle may reroute such that the route includes an optional stop. We assume an uncapacitated vehicle, i.e., a passenger request is only rejected if doing so renders visiting a compulsory stop during its time window impossible or the cost of doing so is too high.

Each passenger request, if served, induces a utility that may correspond to the fare charged for the particular request but may also account for service quality or priority indicators, e.g., if a request was already rejected in the previous route (Crainic et al. 2005). In this work, we consider the utility to be the fare charged. We aim to maximize the profit of our tour, i.e., the difference between the revenue of accepted passenger requests and cost.

To explain the routing aspects of our problem, including the potential and impact of accepting or rejecting passenger requests, we first formulate the full information problem setting, which we will later utilize to decompose our operational route planning problem under uncertainty into multiple full information problems. Second, we introduce the operational route planning problem under uncertainty as an MDP.

5.3.1 Full information problem setting

In our full information setting, information about all requests is known before taking any acceptance or rejection decision, and determining the vehicle's route. Similar to Crainic et al. (2005), we define the set of all compulsory stops as $H = \{f_1, f_2, ..., f_{n+1}\}$ and assume our route starts at compulsory stop f_1 and ends at compulsory stop f_{n+1}. Furthermore, there are predefined time windows $[a_h, b_h]$ for each compulsory stop $f_h, h = 1, \ldots, n+1$ at which the vehicle has to serve stop f_h. If the vehicle arrives earlier than a_h at f_h, the vehicle must wait. We associate a set F_h of optional stops to each pair of consecutive compulsory stops $\langle f_h, f_{h+1}\rangle$. The vehicle only serves an optional stop if we accept a request inducing pick-up or drop off at the stop. Sets F_h are mutually disjoint. We define *segment* h for each pair $\langle f_h, f_{h+1}\rangle$ as the directed graph $G_h = (N_h, A_h)$, where $N_h = F_h \cup \{f_h, f_{h+1}\}$ is the node set, and $A_h \subseteq N_h \times N_h$ is the set of arcs connecting the stops. Then, $G = \bigcup_h G_h$ is the entire graph. Travel costs c_{ij} and travel times τ_{ij}, which include service time at node i, for each arc $(i, j) \in A$ are given and positive.

We denote the *request set* as R, where $r \in R$ is defined as a pair $\langle s(r), d(r)\rangle$ of on and off-boarding stops. Here, $s(r), d(r) \subseteq \bigcup_h N_h$ are the sets of all stops within a certain proximity of the origin or destination coordinates of request $r \in R$ respectively. For each $r \in R$, let $n_1 \in s(r)$ and $n_2 \in d(r)$ with n_1 and n_2 being an optional stop and let $h(n_1)$ and $h(n_2)$ denote the on and off-boarding segments of r, respectively. We assume that $h(n_1) < h(n_2)$ holds, i.e., n_1 and n_2 do not belong to the same segment. Accordingly, we do not have to consider precedence constraints between stops within the same segment. Additionally, we implicitly handle precedence constraints regarding on and off-boarding stops of each request by the sequencing of compulsory stops.

Figure 2 shows an example of a problem setting with compulsory stops f_h (red), optional stops (blue), time windows $[a_h, b_h]$, segments F_h (grey), segment arcs A_h (black), and on-/off-boarding arcs (dotted grey) defining the on and off-boarding stops. Additionally, there exists the direct path only including the compulsory stops (yellow), and two passenger requests represented by houses and work buildings. Figure 3 shows two possible route realizations of this problem setting. Let y_r be the decision variables that indicate whether we accept request $r \in R$ ($y_r = 1$) or not ($y_r = 0$). In the first route realization, both requests are served, i.e., $y_{r_1} = y_{r_2} = 1$. In the second route

Figure 2: Planned bus route with compulsory and optional stops before requests received

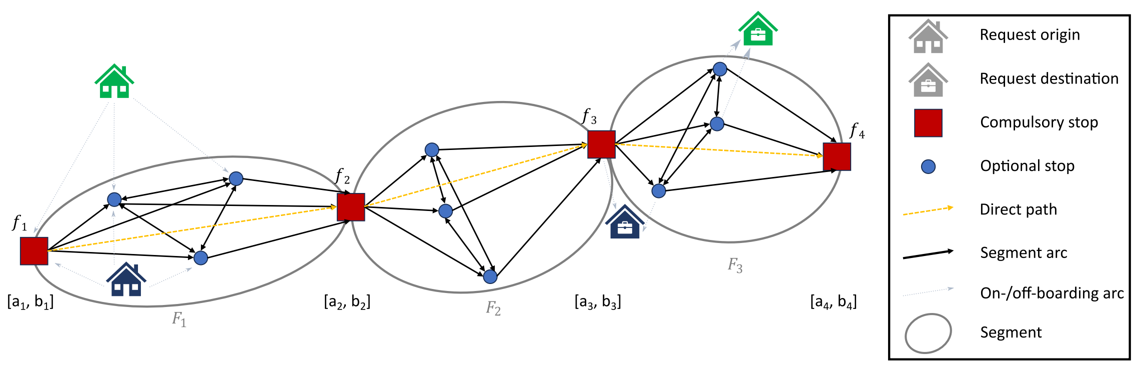
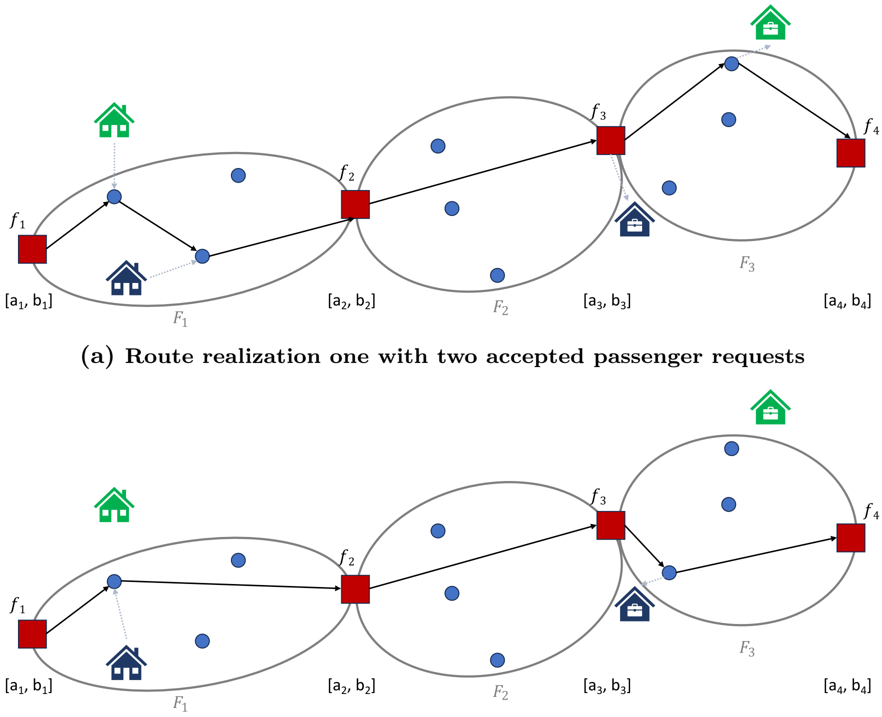


Figure 3: Different bus routes for different accepted passenger requests



(a) Route realization one with two accepted passenger requests

(b) Route realization two with one accepted passenger request

realization, only one request is served, i.e., $y_{r_1} = 1$ and $y_{r_2} = 0$. Note that even though the second request (green) may enter the bus at the same station as in the first route variant, it can not reach its destination.

We associate a *utility* $u(r) > 0$ with each request $r \in R$, and aim to maximize the profit of our tour, i.e., the difference between the utility of accepted passenger requests and cost. In this context, we use decision variables x_{ij} to indicate whether arc $(i, j) \in A$ is used in the vehicle's tour ($x_{ij} = 1$) or not ($x_{ij} = 0$). Furthermore, decision variable t_h is the starting time from all compulsory stops $f_h, h = 1, \ldots, n+1$ and $\delta^i_{s(r)}$ ($\delta^i_{d(r)}$) is a constant of value 1 if $i \in s(r)$ ($i \in d(r)$) and 0 otherwise.

We define d_i as

$$d_i = \begin{cases} 1, & \text{if } i = f_1, \\ -1, & \text{if } i = f_{n+1}, \\ 0, & \text{otherwise.} \end{cases}$$

With this notation, we formulate the DAS full information problem as follows.

$$\max \quad \sum_{r \in R} u(r) y_r - \sum_{i,j \in A} c_{ij} x_{ij} \tag{5.3.1a}$$

$$y_r \leq \sum_{i \in N} \sum_{j \in N^+(i)} \delta^i_{s(r)} x_{ij} \qquad \forall r \in R \qquad (5.3.1b)$$

$$y_r \leq \sum_{j \in N} \sum_{i \in N^-(j)} \delta^j_{d(r)} x_{ij} \qquad \forall r \in R \qquad (5.3.1c)$$

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = d_i \qquad \forall i \in N \qquad (5.3.1d)$$

$$t_h + \sum_{i,j \in A_h} \tau_{ij} x_{ij} \leq t_{h+1} \qquad h = 1, \ldots, n \qquad (5.3.1e)$$

$$a_h \leq t_h \leq b_h \qquad h = 1, \ldots, n+1 \qquad (5.3.1f)$$

$$\sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{ij} \leq |Q| - 1 \qquad \forall Q \subseteq N_h, |Q| \geq 2, h = 1, \ldots, n \qquad (5.3.1g)$$

$$y_r \in \{0, 1\} \qquad \forall r \in R \qquad (5.3.1h)$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in A \qquad (5.3.1i)$$

Objective function (5.3.1a) maximizes the operators profit according to the request utility $u(r)$ and the arc costs c_{ij}. Constraints (5.3.1b) and (5.3.1c) ensure that a request $r \in R$ can only be served if one stop out of the set of on-boarding stops $s(r)$ and one stop of the set of off-boarding stops $d(r)$ are visited by the vehicle. Constraints (5.3.1d) ensure flow conservation at all optional and compulsory stops. Note that we do not need to specify that all compulsory stops are included in the vehicle's tour as the graph structure implies that each compulsory stop f_h, $h = 1, \ldots, n$ only has ingoing arcs $N^-(f_h)$ from segment h and outgoing arcs $N^+(f_h)$ to segment f_{h+1}, i.e., a path from f_1 to f_{n+1} must include all compulsory stops. Constraints (5.3.1e) define the starting times t_h at all compulsory stops. Constraints (5.3.1f) ensure that the time windows at the compulsory stops are met. Constraints (5.3.1g) ensure that no subtours within the segments exist and are ensured through lazy constraints. Finally, Constraints (5.3.1h) and (5.3.1i) define the domains of y_r and x_{ij}.

5.3.2 Dynamic and stochastic problem setting

In the operational route planning problem under uncertainty, passengers issue requests one after another before the vehicle begins its route. We model decisions as being made according to a discrete time horizon $\theta = 1, \ldots, T-1$, at which one new request r_θ may arrive, and we must decide whether to accept or reject that request. Similar to the full information setting, we assume each request to induce a pair of on and off-boarding stop sets $\langle s(r_\theta), d(r_\theta) \rangle$ with a utility $u(r_\theta)$. At decision time $T < a_1$, we have to determine the optimal route serving all accepted requests and respecting all time windows at compulsory stops.

We model our dynamic operational route planning problem under uncertainty as a finite-horizon MDP with the following state spaces, action spaces, and value functions.

System states: At any event-based decision time $\theta = 1, \ldots, T$, the system state χ_θ consists of previously accepted passenger requests S_θ that have to be fulfilled, a new passenger request r_θ, and decision time θ.

Action spaces: At decision time $\theta = 1, \ldots, T-1$, our action space $\mathcal{A}(\chi_\theta)$ entails either accepting or rejecting the new request r_θ. If, due to the time windows at the compulsory stops, there is no route that can serve all previously accepted passengers χ_θ and the current request, the action space will be reduced to rejecting the new request. This ensures feasibility at decision time T. At decision time T, our action space $\mathcal{A}_T(\chi_\theta)$ is the set of all routes serving all accepted requests χ_θ and respecting all time windows at compulsory stops.

Value functions: We define our value function at decision times $\theta = 1, \ldots, T-1$ as follows.

$$V_\theta(\chi_\theta) = \max_{a \in \mathcal{A}(\chi_\theta)} \big\{ R(\chi_\theta, a) + \mathbb{E}\big[V_{\theta+1}(X(\chi_\theta, a, W_{\theta+1}))\big] \big\} \tag{5.3.2}$$

Here, $R(\chi_\theta, a)$ is the payoff from taking action $a \in \mathcal{A}(\chi_\theta)$, i.e., accepting a new request r_θ yields a reward of $u(r_\theta)$, and the system state changes from χ_θ to $\chi_{\theta+1} = X(\chi_\theta, a, W_{\theta+1})$ through transition function $X(\cdot)$ when action a is taken and the random variable $W_{\theta+1}$ is realized. Here $W_{\theta+1}$ is the random variable of a new request $r_{\theta+1}$, arriving between decision time θ and $\theta+1$. As we only have to decide the DAS route after all passenger request decisions have been taken, our value function at decision time T consists of the cost of serving all accepted passenger requests and respecting the compulsory stop's time windows. Accordingly, our value function at decision time T is as follows.

$$V_T(\chi_T) = \max_{a \in \mathcal{A}_T(\chi_T)} R_T(a) \tag{5.3.3}$$

Here, $R_T(a) < 0$ is the cost of the DAS route chosen through action a.
Policy: Let S_Θ denote the set of all possible system states. A policy $\delta : S_\Theta \to \mathcal{A}(\chi_\Theta)$ is a mapping that assigns to any system state χ_Θ a decision $a \in \mathcal{A}(\chi_\Theta)$.

Full information upper bound: We can compute a full-information upper bound by supposing that all requests $\bigcup_\theta r_\theta$ are already known at timestep $\theta = 1$ and solve this problem via our full information model (Problem 5.3.1). In this context, we define a sample path for a given decision time θ as one realization of requests at each subsequent decision time θ' with $\theta < \theta' < T$. We will use these sample paths in the methodology we present next to approximate our dynamic and stochastic problem with one that is stochastic but static as we presume knowledge of the distribution of sample paths.

5.4 Methodology

In this section, we first show how we utilize a rolling horizon look-ahead approach in which the decision of whether to accept or reject a request at a given decision time is made based on an estimate of its impact on both the ability to accept future requests and routing costs. This estimate is based on an approximation in which future requests remain uncertain but are static, as they are revealed simultaneously. To compute this estimate, we formulate a two-stage stochastic program in which the first stage reflects the decision to accept or reject the current request. The second stage problem presumes immediate and complete information for a given sample path and thus knowledge of all future requests that can be accepted or rejected. This second stage problem has an objective that reflects the revenues gained from already accepted requests and accepting future requests in periods before decision time T, and routing costs incurred at the last decision time T.

We then show how to utilize a sample-based approximation to approximate the expectation of the objective function of the second-stage problem through a multi-scenario deterministic model. Then, we show how to decompose our sample-based approximation into several full information models of the operational route planning problem (Section 5.4.1.1). Afterwards, we propose a consensus-based heuristic approach that reduces the number of full information problems needed to solve the operational route planning problem under uncertainty by 50% at each decision time θ (Section 5.4.1.2). Finally, we introduce a straightforward myopic approach (Section 5.4.2) for benchmarking purposes.

Rolling horizon framework

We use a rolling horizon framework to solve our MDP model. In general, the rolling horizon framework is a decision-making strategy in which a problem is repeatedly solved over a limited planning horizon that moves forward in time. At each step, a decision is implemented, and the horizon is shifted forward, allowing for continuous adaptation to new information and changing conditions. One approach to solve the operational route planning problem under uncertainty is to use a rolling horizon look-ahead framework in which the dynamic and stochastic nature of future requests is approximated as static but stochastic. Specifically, we decompose the problem of maximizing the total operators' profit at the end of the problem time horizon T into multiple subsequent stochastic optimization problems at each decision time θ at which a new request r_θ arises. In these stochastic programs, we approximate the dynamic and stochastic future passenger requests through static and stochastic sample paths. Another approach is to ignore all information about future passenger requests and use a myopic strategy.

5.4.1 Full information approximation

In the following, we use a static approximation to capture the dynamic nature of our problem. To do so, at each decision time θ, we assume a probability distribution that describes the likelihood of each potential set of passenger requests occurring. We then optimize our decision at θ given this distribution by solving a two-stage stochastic program.

Two-stage stochastic program

We define y_{r_θ} as the accept or reject decisions of the new passenger request r_θ at decision time θ. Additionally, let S_θ be all passenger requests accepted up until decision time θ, and let r_θ be the new passenger request as defined in Section 5.3. Furthermore, we define $\boldsymbol{\xi_\theta}$ as the probability distribution of possible sample paths at decision time θ conditioned on the already accepted requests S_θ and

$R(\boldsymbol{\xi_\theta})$ as a realization of $\boldsymbol{\xi_\theta}$. Note that each realization $R(\boldsymbol{\xi_\theta})$ is a sample path as defined in Section 5.3.2, i.e., a set of requests that consists of all already accepted requests S_θ, the new passenger request r_θ, and possible future requests. Consequently, it holds that $r_\theta \in R(\boldsymbol{\xi_\theta})$ and $r \in R(\boldsymbol{\xi_\theta})$ for $r \in S_\theta$. Problems 5.4.1 and 5.4.2 define the first and second stages of our stochastic program for each decision time θ that we solve when using our full information approximation. Note that instead of including the utility of serving all previously accepted requests and potentially serving our current passenger request in the first stage of our stochastic programs, we choose to include this utility in the first sum of the objective function of the second stages $Q(S_\theta, y_{r_\theta}, \boldsymbol{\xi_\theta})$ (see Problem 5.4.2). Even though this utility is deterministic, i.e., independent from the distribution of future passenger requests, this notation allows for an easier decomposition into independent full information problems in Section 5.4.1.1.

$$\max \quad \mathbb{E}[Q(S_\theta, y_{r_\theta}, \boldsymbol{\xi_\theta})] \tag{5.4.1a}$$

$$y_{r_\theta} \in \{0,1\} \tag{5.4.1b}$$

$$Q(S_\theta, y_{r_\theta}, \boldsymbol{\xi_\theta}) = \max \quad \sum_{r \in \xi_\theta} u(r) y_r - \sum_{i,j \in A} c_{ij} x_{ij} \tag{5.4.2a}$$

$$\begin{aligned}
y_r &= 1 && \forall r \in S_\theta && (5.4.2b)\\
y_r - \sum_{i \in N} \sum_{j \in N^+(i)} \delta^i_{o(r)} x_{ij} &\leq 0 && \forall r \in R(\boldsymbol{\xi_\theta}) && (5.4.2c)\\
y_r - \sum_{j \in N} \sum_{i \in N^-(j)} \delta^j_{d(r)} x_{ij} &\leq 0 && \forall r \in R(\boldsymbol{\xi_\theta}) && (5.4.2d)\\
\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} &= d_i && \forall i \in N && (5.4.2e)\\
t_h + \sum_{i,j \in A_h} \tau_{ij} x_{ij} &\leq t_{h+1} && h = 1, \ldots, n && (5.4.2f)\\
a_h \leq t_h &\leq b_h && h = 1, \ldots, n+1 && (5.4.2g)\\
\sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{ij} &\leq |Q| - 1 && \forall Q \subseteq N_h, |Q| \geq 2, h = 1, \ldots, n && (5.4.2h)\\
y_r &\in \{0,1\} && \forall r \in R(\boldsymbol{\xi_\theta}) && (5.4.2i)\\
x_{ij} &\in \{0,1\} && \forall i,j \in A && (5.4.2j)
\end{aligned}$$

Problem 5.4.1 maximizes the expected profit based on the decision y_{r_θ} at decision time θ, all previsously accepted passengers S_θ, and the probability distribution of future passenger requests

$\boldsymbol{\xi_\theta}$. The objective function (5.4.2a) aims to maximize the expected operator profit under the assumption that passenger request r_θ is either accepted ($y_{r_\theta} = 1$) or rejected ($y_{r_\theta} = 0$) in the first stage, considering previously accepted requests and future passenger requests at decision time θ. Constraints (5.4.2b) ensure all requests accepted before decision time θ are served. Similarly to Problem (5.3.1), Constraints (5.4.2c) and (5.4.2d) ensure that a request $r \in R(\boldsymbol{\xi_\theta})$ can only be served if one stop out of the set of on-boarding stops $s(r)$ and one stop of the set of off-boarding stops $d(r)$ are visited by the vehicle. Constraints (5.4.2e) ensure flow conservation at all optional and compulsory stops. Constraints (5.4.2f) define the starting times t_h at all compulsory stops, and Constraints (5.4.2g) ensure that the time windows at the compulsory stops are met. Constraints (5.4.3i) ensure that no subtours within the segments exist and are ensured through lazy constraints. Finally, Constraints (5.4.2i) and (5.4.2j) define the domains of y_r and x_{ij}.

Sample-based approximation of the two-stage stochastic program

We utilize a sample-based approximation to tackle the computational complexity arising from computing expected second stage costs of the stochastic program. Let Ω define a set of possible request scenarios with the probability of occurrence of p_σ for all $\sigma \in \Omega$. Here, for each $\sigma \in \Omega$ it holds that $S_\theta \subseteq \sigma$ and $r_\theta \in \sigma$. We can then approximate Problem 5.4.1 through a multi-scenario deterministic model, in which we approximate the random variable $\boldsymbol{\xi_\theta}$ through a set of scenarios Ω with corresponding request sets R^σ_θ for each scenario $\sigma \in \Omega$. Problem 5.4.3 shows the sample-based approximation of Problem 5.4.2.

$$Q^\Omega(S_\theta, y_{r_\theta}, \Omega) = \max \quad \sum_{\sigma\in\Omega} p_\sigma \Big(\sum_{r^\sigma \in R^\sigma_\theta} u(r^\sigma) y_{r^\sigma} - \sum_{i,j\in A} c_{ij} x^\sigma_{ij} \Big) \tag{5.4.3a}$$

$$y_{r^\sigma_\theta} = y_{r_\theta} \tag{5.4.3b}$$

$$y_{r^\sigma} = 1 \qquad \forall r^\sigma \in S_\theta, \forall \sigma \in \Omega \tag{5.4.3c}$$

$$y_{r^\sigma} - \sum_{i\in N} \sum_{j\in N^+(i)} \delta^i_{o(r^\sigma)} x^\sigma_{ij} \leq 0 \qquad \forall r^\sigma \in R^\sigma_\theta, \forall \sigma \in \Omega \tag{5.4.3d}$$

$$y_{r^\sigma} - \sum_{j\in N} \sum_{i\in N^-(j)} \delta^j_{d(r^\sigma)} x^\sigma_{ij} \leq 0 \qquad \forall r^\sigma \in R^\sigma_\theta, \forall \sigma \in \Omega \tag{5.4.3e}$$

$$\sum_{j\in N^+(i)} x^\sigma_{ij} - \sum_{j\in N^-(i)} x^\sigma_{ji} = d_i \qquad \forall i \in N, \forall \sigma \in \Omega \tag{5.4.3f}$$

$$t^\sigma_h + \sum_{i,j\in A_h} \tau_{ij} x^\sigma_{ij} \leq t^\sigma_{h+1} \qquad h = 1, \ldots, n, \forall \sigma \in \Omega \tag{5.4.3g}$$

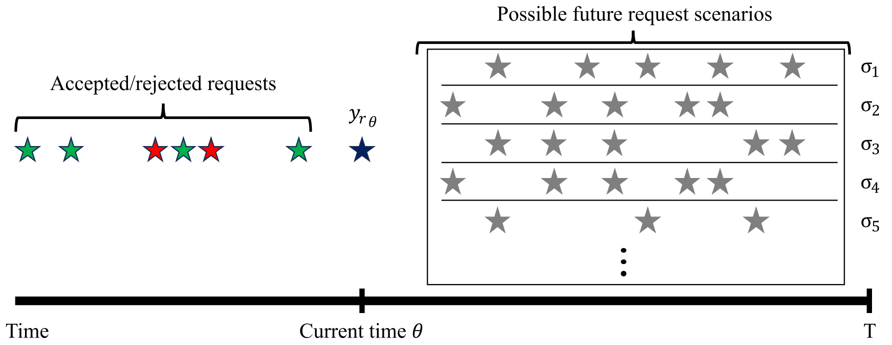
$$a_h \leq t^\sigma_h \leq b_h \qquad h = 1, \ldots, n+1 \tag{5.4.3h}$$

$$\sum_{i\in Q} \sum_{j\neq i, j\in Q} x^\sigma_{ij} \leq |Q| - 1 \qquad \forall Q \subseteq N_h, |Q| \geq 2, h = 1, \ldots, n, \forall \sigma \in \Omega \tag{5.4.3i}$$

$$y_{r^\sigma} \in \{0, 1\} \qquad \forall r^\sigma \in R^\sigma_\theta, \forall \sigma \in \Omega \tag{5.4.3j}$$

$$x^\sigma_{ij} \in \{0, 1\} \qquad \forall i, j \in A, \forall \sigma \in \Omega \tag{5.4.3k}$$

Figure 4: Visualization of our sample-based approximation where we consider possible future request scenarios at the same time



$$y_{r_\theta} \in \{0,1\} \tag{5.4.3l}$$

The objective function (5.4.3a) calculates the expected operators' profit over all scenarios $\sigma \in \Omega$. Additionally, Constraint (5.4.3b) ensures, that over all scenarios, request r_θ is either always accepted ($y_{r_\theta} = 1$) or never accepted ($y_{r_\theta} = 0$). Here, we introduce variables $y_{r_\theta^\sigma}$ as this notation enables us to describe our heuristic approach in Section 5.4.1.2.

Figure 4 visualizes the rational of this sample-based approximation approach in which we consider all possible future request scenarios at the same time.

5.4.1.1 Decomposition of sample-based approximation

Solving the sample-based approximation (Problem 5.4.3) is time-consuming. As we aim to take a decision fast and immediately inform the passenger whether we accept or reject their request, we show how to decompose the sample-based approximation (Problem 5.4.3) into multiple independent full information planning problems (Problems 5.3.1) in the following. Note that even though the following proof of the decomposition is rather intuitive, it is necessary to substantiate our scenario-based decomposition and consensus-based heuristic. In the following, we branch on variable y_{r_θ} and thus, implicitly, on the second stage variables $y_{r_\theta^\sigma}$, $\forall \sigma \in \Omega$. Accordingly, Constraints (5.4.3b) automatically hold $\forall \sigma \in \Omega$, and we can decompose $Q^\Omega(S_\theta, y_{r_\theta}, \Omega)$ by scenario as follows.

$$Q^\Omega(S_\theta, y_{r_\theta}, \Omega) = \max\{\sum_{\sigma \in \Omega} p_\sigma Q^\sigma(S_\theta, y_{r_\theta}, \sigma) : y_{r_\theta} \in \{0,1\}\} \tag{5.4.4}$$

Here, we define $Q^\sigma(S_\theta, y_{r_\theta}, \sigma)$ as the optimal objective value of Problem 5.4.5 with $y_{r_\theta} = y_{r_\theta^\sigma}$ if Problem 5.4.5 is feasible and $-\infty$ otherwise. Through Theorem 5.4.1, we show that Problem 5.4.5 being feasible is independent of scenario σ. As we assume that serving no requests is always feasible, we ensure that $Q^\Omega(S_\theta, y_{r_\theta}, \Omega) > -\infty$ for all $\theta < T$. Accordingly, we always ensure a feasible solution at each decision time θ.

Theorem 5.4.1. *Problem 5.4.5 is infeasible for $\sigma \in \Omega$ with $y_{r_\theta} = 1$ if and only if Problem 5.4.5 is infeasible for $S_\theta \cup r_\theta$ with $y_{r_\theta} = 1$.*

Proof. Let $\sigma = S_\theta \cup r_\theta \cup \mathcal{F}$ with $\mathcal{F}$ being a sample of possible future requests. We define $\Lambda(\sigma)$ as the feasible region of Problem 5.4.5 for $\sigma \in \Omega$ and $\Lambda(S_\theta \cup r_\theta)$ as the feasible region for $S_\theta \cup r_\theta$ with $y_{r_\theta} = 1$. For the Theorem to hold, we have to show that $\Lambda(\sigma) = \emptyset$ if and only if $\Lambda(S_\theta \cup r_\theta) = \emptyset$. We assume that serving no customers is always feasible. Accordingly, $\Lambda(S_\theta \cup r_\theta)$ is a facet of $\Lambda(\sigma)$ with $\Lambda(\sigma) \cap \{y_r = 0 : r \in \mathcal{F}\} = \Lambda(S_\theta \cup r_\theta)$. It follows directly that $\Lambda(\sigma) = \emptyset \Leftrightarrow \Lambda(S_\theta \cup r_\theta) = \emptyset$. □

Additionally, we define Problem 5.4.5 as follows.

$$Q^\sigma(S_\theta, y_{r_\theta}, \sigma) = \max \quad \sum_{r^\sigma \in R_\theta^\sigma} u(r^\sigma) y_{r^\sigma} - \sum_{i,j \in A} c_{ij} x_{ij}^\sigma \tag{5.4.5a}$$

$$y_{r_\theta^\sigma} = y_{r_\theta} \tag{5.4.5b}$$

$$y_{r^\sigma} = 1 \qquad \forall r^\sigma \in S_\theta \tag{5.4.5c}$$

$$y_{r^\sigma} - \sum_{i \in N} \sum_{j \in N^+(i)} \delta^i_{o(r^\sigma)} x_{ij}^\sigma \leq 0 \qquad \forall r^\sigma \in R_\theta^\sigma \tag{5.4.5d}$$

$$y_{r^\sigma} - \sum_{j \in N} \sum_{i \in N^-(j)} \delta^j_{d(r^\sigma)} x_{ij}^\sigma \leq 0 \qquad \forall r^\sigma \in R_\theta^\sigma \tag{5.4.5e}$$

$$\sum_{j \in N^+(i)} x_{ij}^\sigma - \sum_{j \in N^-(i)} x_{ji}^\sigma = d_i \qquad \forall i \in N \tag{5.4.5f}$$

$$t_h^\sigma + \sum_{i,j \in A_h} \tau_{ij} x_{ij}^\sigma \leq t_{h+1}^\sigma \qquad h = 1, \ldots, n \tag{5.4.5g}$$

$$a_h \leq t_h^\sigma \leq b_h \qquad h = 1, \ldots, n+1 \tag{5.4.5h}$$

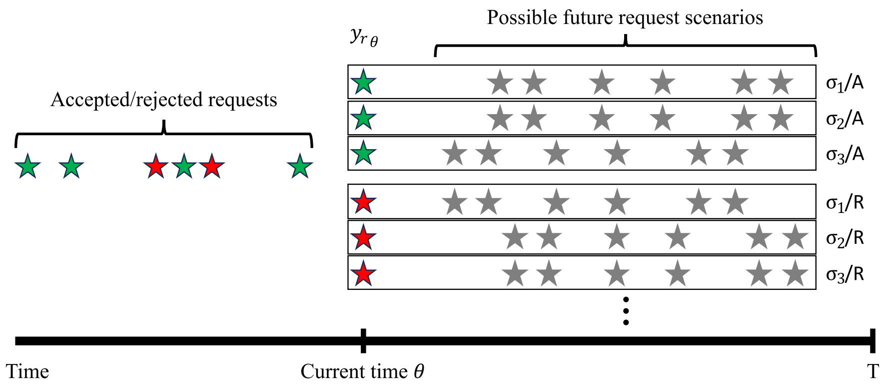
$$y_{r^\sigma} \in \{0, 1\} \qquad \forall r^\sigma \in R_\theta^\sigma \tag{5.4.5i}$$

$$x_{ij}^\sigma \in \{0, 1\} \qquad \forall i, j \in A \tag{5.4.5j}$$

Note that in Problem 5.4.5 y_{r_θ} and σ are fixed. Consequently, the objective function (5.4.5a) calculates the operators' profit for the request set R_θ^σ of scenario σ under the assumption that decision y_{r_θ} has been taken (5.4.5b) and all requests accepted before decision time θ are served (5.4.5c). Furthermore, Constraints (5.4.5d) and (5.4.5e) ensure that a request $r^\sigma \in R_\theta^\sigma$ can only be served if one stop out of the set of on-boarding stops $s(r)$ and one stop of the set of off-boarding stops $d(r)$ are visited by the vehicle. Constraints (5.4.5f) ensure flow conservation at all optional and compulsory stops.

Constraints (5.4.5g) define the starting times at all compulsory stops, and Constraints (5.4.5h) ensure that the time windows at the compulsory stops are met. Finally, Constraints (5.4.5i) and (5.4.5j) define the dimensions of y_{r^σ} and x_{ij}^σ. Accordingly, Problem 5.4.5 is now equivalent to our

Figure 5: Visualization of our sample-based approximation decomposition into different subproblems where we independently solve different scenarios σ_i with acceptance (A) or rejection (R) of request r_θ



full information Problem 5.3.1 with all previously accepted requests' variables $r^\sigma \in S_\theta$ fixed to one in Constraints (5.4.5c) and our current request variable y_{r_θ} either fixed to zero or one.

Figure 5 visualizes how our decomposition allows us to solve all scenarios $\sigma \in \Omega$ independently in a full information problem setting, once assuming we reject the current passenger request ($y^\sigma_{r_\theta} = 0$) and once assuming we accept it ($y^\sigma_{r_\theta} = 1$). Afterward, we either accept or reject the current passenger request depending on whether $\sum_{\sigma\in\Omega} p_\sigma Q^\sigma(S_\theta, 1, \sigma) > \sum_{\sigma\in\Omega} p_\sigma Q^\sigma(S_\theta, 0, \sigma)$ (accept) or not (reject).

Note that solving all full information subproblems independently is easily parallelizable. Indeed, we will show in Section 5.5 that this decomposition allows us to make a decision much faster than solving Problem 5.4.3 directly.

5.4.1.2 Consensus-based heuristic approach

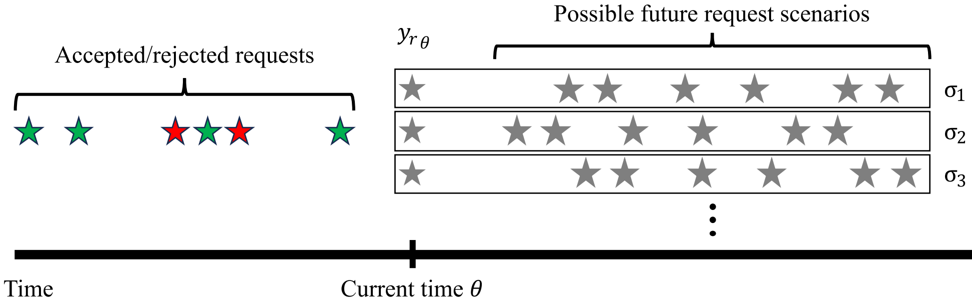
Our decomposition of the operational route planning problem under uncertainty allows us to solve multiple full information problems, once for accepting the current passenger request and once for rejecting it. Alternatively, Figure 6 visualizes how we can also solve the individual full information problems without fixing the current passenger requests variable $y_{r^\sigma_\theta}$, i.e., neglecting Constraint (5.4.5b), and take a weighted majority vote to decide on whether to accept or reject the current passenger request.

Such a weighted majority vote counteracts instances in which there exist scenarios that achieve disproportionately large profits. Our conjecture is that in our original decomposition, these scenarios may influence the acceptance decision substantially, even if the probability of occurrence is low. Deciding whether to accept or reject passenger request r_θ based on a weighted majority vote decouples the decision from the absolute objective value.

$$y_{r_\theta} = \begin{cases} 1, & \text{if } \sum_{\sigma\in\Omega} p_\sigma \mathbb{1}_{\{y_{r^\sigma_\theta}=1\}} \geq \frac{|\Omega|}{2}, \\ 0, & \text{otherwise} \end{cases} \tag{5.4.6}$$

Here, for $\sigma \in \Omega$ we define the indicator function $\mathbb{1}_{\{y_{r^\sigma_\theta}=1\}}$ as follows

Figure 6: Visualization of our heuristic approach where we independently solve different scenarios σ_i without fixing the decision for r_θ



$$\mathbb{1}_{\{y_{r_\theta^\sigma}=1\}} = \begin{cases} 1, & \text{if } y_{r_\theta^\sigma} = 1, \\ 0, & \text{if } y_{r_\theta^\sigma} = 0 \end{cases} \tag{5.4.7}$$

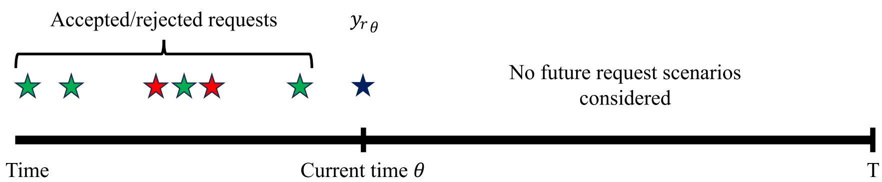
Similar to our previous decomposition approach, solving the independent subproblems of our heuristic approach is easily parallelizable.

5.4.2 Myopic approach

So far, we always considered future passenger requests and approximated them through a stochastic distribution. Instead, we can also make a decision whether to accept or reject passenger request r_θ based on a myopic approach. Figure 7 visualizes how in such an approach, we neglect all information about future passenger requests and accept passenger requests r_θ only if this immediately increases our profit. To do so, we solve Problem 5.4.5 only for one scenario $\sigma = S_\theta \cup r_\theta$. Accordingly, we decide whether to accept or reject passenger request r_θ based on Equation (5.4.8) in accordance to Constraint (5.4.5b).

$$y_{r_\theta} = \begin{cases} 1, & \text{if } y_{r_\theta^\sigma} = 1, \\ 0, & \text{otherwise} \end{cases} \tag{5.4.8}$$

Figure 7: Visualization of our myopic approach where we do not consider future request scenarios



5.5 Computational analysis

Our analysis is fivefold. First, we analyze the computational efficiency of our Two-stage stochastic program (2S-SP), our Consesus-based heuristic approach (CHA), and the Myopic approach. To do so, we first compare the computation times, profits, and percentages of passengers served in solutions produced by those approaches on two different DAS lines with different degrees of route flexibility. Second, we perform a sensitivity analysis on the number of scenarios in our 2S-SP and CHA. Here, we analyze the change in computation times, profits, and served passengers attained through our 2S-SP and CHA approaches under different numbers of future demand scenarios. Third, we evaluate the solution quality of our approaches compared to an optimal solution under full information. To do so, we compare our 2S-SP, CHA, and Myopic approaches to the profits obtained and passengers served under full information, solved with our full information model (Problem 5.3.1). Fourth, we analyze the benefits and drawbacks of converting a fixed-line bus system to a DAS. To do so, we compare both systems' profits, served passengers, and travel distances to show how an increase in travel distances translates to an increase in profits and served passengers. Fifth, we aim to understand how the passenger's willingness to walk to nearby stops and the flexibility of the bus route influences the profit and number of served passengers. Here, we show the differences in profits and number of served passengers under varying passenger walking willingness and route flexibility.

Instances

We use data provided by Stadtwerke München GmbH (SWM), the public transportation provider of Munich. This SWM data consists of the average number of passengers boarding and aligning at the different stops of two different fixed bus routes over the entire day. The data was collected between April and June 2022. Based on this data, we generate between five and 40 demand scenarios in three steps. First, we randomly determine an OD-matrix between the stops of the original fixed route through iterative proportional fitting (Rüschendorf 1995) based on the SWM data. This OD-matrix indicates for each pair of compulsory stops the number of passengers seeking transportation between those stops. Here, the origins and destinations of all passengers lie at the stops of the original fixed bus route. Second, based on the determined OD-matrix, we randomly map passenger origins and destinations onto buildings within 300 meters of their respective original origin and destination. These buildings correspond to the pick-up and drop-off locations of a passenger request and, in combination with the walking distance, later define the pick-up and drop-off sets of the passenger requests. Note that if there are multiple requests between the same origin and destination in the OD-matrix, we individually map each request to their own pick-up and drop-off locations. Third, to be able to model a point process in which the requests arrive in the system over time, we use random sampling to obtain a request time between zero and 10,800 seconds (3 hours) before the route's departure time for each passenger request.

Besides the generation of passenger demand, we also utilize the SWM data to determine the compulsory and optional stops of our route. To do so, we utilize a *compulsory stop factor* (CSF), which determines the degree of flexibility of our DAS line. This CSF lies between zero and one. The first and last stops of the original fixed bus route are always marked as compulsory stops as they often lie at critical transfer hubs or are essential for operations, e.g., driver breaks. Additionally, we order the stops of the original fixed route based on their passenger demand utilization and mark a percentage of them as compulsory. This percentage corresponds to the CSF. Furthermore, we allow each intersection within 600 meters of the original fixed bus route to be an optional stop. Here, we merge optional stops within 200 meters of each other.

Computational setting

All our experiments have been conducted on a standard desktop computer equipped with two `Intel(R) Core(TM) i9-9900, 3.1 GHz CPU` and a total of `4 GB` of `RAM`, running `Ubuntu 20.04`. We have implemented all algorithms in `Python (3.8.11)` using `Gurobi 10.0.0`. Our source code can be found on https://github.com/tumBAIS/Route_Planning_DAS.

5.5.1 Computational effectiveness of the proposed methods

The profit of our operational route planning problem under uncertainty is driven by two quantities. The revenue of accepted passenger requests and travel costs. To see what drives solution quality and understand the computational effectiveness of our proposed methods, we compare our proposed methods on two different routes based on the achieved profit, served passengers (SP) out of all issued requests, computation times in serial computation (ser.), and a theoretical lower bound of parallel computation times (par.). Here, the theoretical lower bounds of parallel computation times for 2S-SP and CHA are given by the maximum runtime of the independent subproblems for each request decision. We include this theoretical lower bound as parallel computation is a typical approach for dynamic decision-making in stochastic problem settings. We compare the different parameters by aggregating the results of solving five different instances for each instance configuration. To do so, we compare our proposed methods with five scenarios in our 2S-SP and CHA approaches, a uniform passenger utility of 750, and a CSF $\in [0.2, 0.4, 0.6, 0.8, 1]$. We analyze our results for different CSF and different numbers of scenarios in our 2S-SP and CHA approaches. Note that in our 2S-SP, we solve our decomposition Problem 5.4.4. We were not able to solve Problem 5.4.3 within four hours for all decision times for our configurations.

5.5.1.1 Computational effectiveness for varying CSF

In this section, we focus on how the degree of flexibility of a DAS line, i.e., the CSF, changes the computational effectiveness of our three approaches. Table 1 shows that for Route 55, our serial-computed CHA approach is between 22 - 54% faster than our 2S-SP approach, and our Myopic approach is between 88 - 98% faster than our CHA. Comparing the theoretical lower bound of parallel computation, we see that our 2S-SP and CHA approaches perform similarly. Note that our 2S-SP still needs to solve twice the number of subproblems as our CHA. We also see that with increasing flexibility of a DAS line, i.e., decreasing CSF, the parallel computation times of our 2S-SP and CHA approaches increase significantly.

Looking at our approaches' profits, we see that our 2S-SP achieves the best profit for all CSFs except 1.0, followed by our CHA approach. As expected, our Myopic approach performs worse than

Table 1: Comparison between 2S-SP, CHA, and Myopic with different compulsory stop factors for Route 55 with 5 future demand scenarios

	2S-SP				CHA				Myopic		
CSF	Profit	SP [%]	ser. [s]	par. [s]	Profit	SP	ser.	par.	Profit	SP	ser.
0.20	25864.38	83.02	4.97	0.98	25792.82	82.87	2.28	0.89	25316.64	82.10	0.25
0.40	23456.57	77.31	2.07	0.36	23290.62	76.85	1.41	0.42	22166.70	75.00	0.14
0.60	22960.73	75.46	1.13	0.20	22956.19	75.46	0.88	0.24	22429.49	75.22	0.09
0.80	21472.71	72.53	0.25	0.03	21366.03	72.07	0.17	0.04	18658.58	63.85	0.02
1.00	20656.19	69.91	0.10	0.01	20561.83	69.60	0.07	0.01	20731.19	70.06	0.01

Table 2: Comparison between 2S-SP, CHA, and Myopic with different compulsory stop factors Route 155 with 5 future demand scenarios

	2S-SP				CHA				Myopic		
CSF	Profit	SP [%]	ser. [s]	par. [s]	Profit	SP	ser.	par.	Profit	SP	ser.
0.20	19181.68	89.92	3.99	0.61	19011.13	89.42	1.72	0.47	18214.41	87.43	0.16
0.40	18575.22	87.41	1.44	0.23	18494.20	86.90	1.19	0.31	17806.07	84.13	0.11
0.60	16878.16	81.11	0.56	0.06	16792.81	80.60	0.32	0.09	16807.90	80.86	0.05
0.80	16498.82	79.09	0.12	0.01	16498.82	79.09	0.06	0.01	16360.81	78.84	0.01
1.00	15282.68	74.81	0.05	0.01	15209.15	74.56	0.03	0.01	15209.15	74.56	0.00

our 2S-SP and CHA approaches but surprisingly outperforms our 2S-SP and CHA at a CSF of 1.0. The reason for that is a higher problem complexity for instances with high flexibility where our 2S-SP and CHA approaches benefit from utilizing historical data. In low flexibility instances, the DAS route is mostly fixed due to the time windows at the compulsory stops. We also see that our 2S-SP approach serves more passengers than our CHA and Myopic approaches.

Table 2 shows that for Route 155, in serial computation, our CHA approach is between 17 - 57% faster than our 2S-SP approach, and our Myopic approach is between 83 - 91% faster than our CHA. Again, the theoretical lower bounds of parallel computation of our 2S-SP and CHA approaches perform similarly and again increase with the growing flexibility of the DAS line. Looking at our approaches' profits, we see that similar to Route 55, 2S-SP achieves the best profit for all CSFs. This can analogously be explained by the higher number of passengers served.

Observation 1. *Regarding solution quality and the number of served passengers, our 2S-SP approach yields the best results, followed by our CHA and finally our Myopic approach. Additionally, our Myopic approach is the fastest in serial computation, followed by our CHA approach, which is between 17 - 57% faster than our 2S-SP approach. Both our 2S-SP and CHA approaches have similar theoretical lower bounds of parallel computation.*

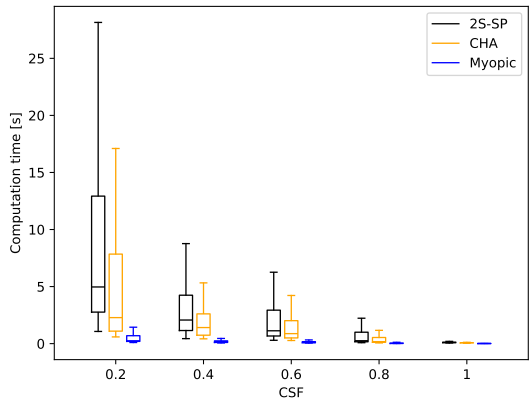
Observation 2. *For each algorithmic approach and instance the serial and parallel computation times of our three approaches increase significantly with growing route flexibility, i.e., a decrease of CSF.*

5.5.1.2 Distribution of computation times for varying CSF

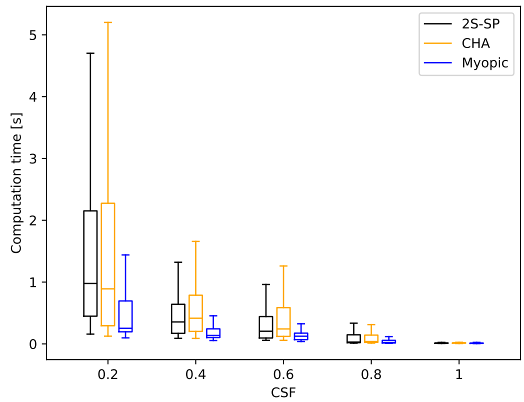
Figure 8 shows the distribution of serial and parallel computation times for our approaches for instances of Route 55. Note that the parallel computation times of our Myopic approach are equivalent to the serial computation times.

Building on Table 1, we see that our Myopic approach has stable computation times of under two seconds for all instances and under one second for instances with a CSF greater than 0.2. Our 2S-SP and CHA approaches are not as stable in serial and parallel computation times and range between $0 - 28$ seconds for serial computation and $0 - 5$ seconds for parallel computation. The reason for that is, first, the different numbers of scenarios in our 2S-SP and CHA approaches and, second, significant variance in computation times for the different subproblems of the requests.

Figure 8: Difference in serial and parallel computation times based on varying CSF for Route 55

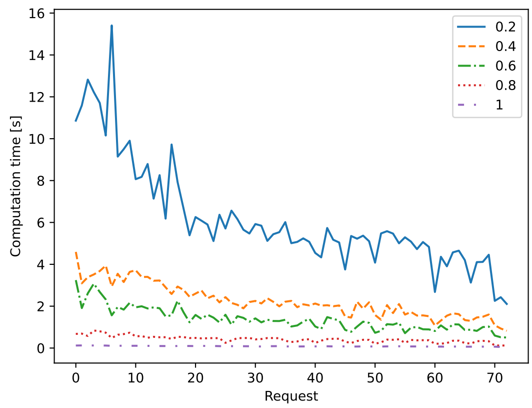


(a) Serial computation times

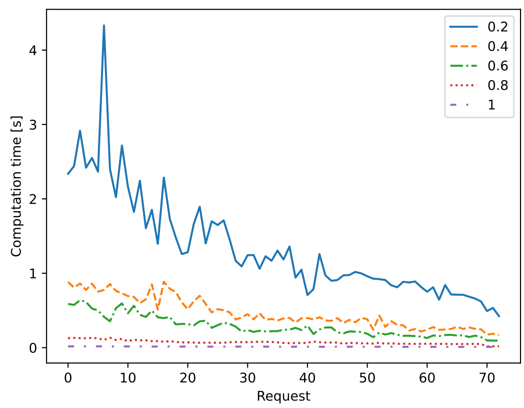


(b) Parallel computation times

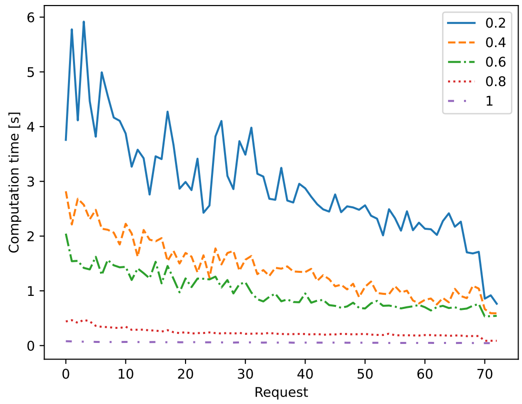
Figure 9: Development of average serial and parallel computation times based on different CSF for Route 55



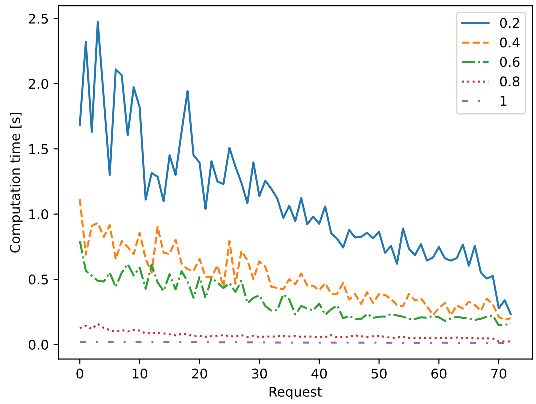
(a) Serial computation times for different CSF (2S-SP)



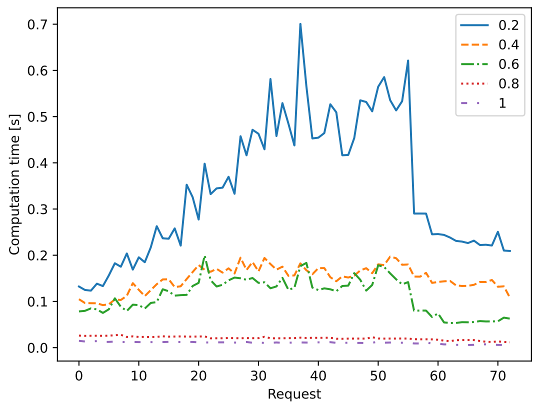
(b) Parallel computation times for different CSF (2S-SP)



(c) Serial computation times for different CSF (CHA)



(d) Parallel computation times for different CSF (CHA)



(e) Serial computation times for different CSF (Myopic)

Figure 9 shows the difference in computation times for the different subproblems of the requests. We see that the more decisions on requests have already been taken, the faster our approaches can solve the subproblem of the current passenger request. Additionally, the later we have to decide on accepting or rejecting a request, the fewer future requests we have to consider in the resulting subproblems. The reduction in computation times over the course of our operational planning problem under uncertainty holds for both serial and parallel computation times for our 2S-SP and CHA approaches.

Interestingly, our Myopic approach shows exactly the opposite development of computation times, i.e., an increase in computation times over the course of our operational planning problem under uncertainty. This suggests that the reduction in future scenarios is the main driver for the reduction in computation times for our 2S-SP and CHA approaches.

Observation 3. *Over the course of our operational planning problem under uncertainty, the subproblems of our 2S-SP and CHA approaches get easier to solve, while the subproblems of our Myopic approach get harder to solve.*

5.5.1.3 Computational effectiveness for varying numbers of scenarios

In this section, we focus on how different numbers of scenarios in our 2S-SP and CHA approaches change the computational effectiveness of our approaches. Note that as our Myopic approach is independent of the number of scenarios, we report only one data point in the tables.

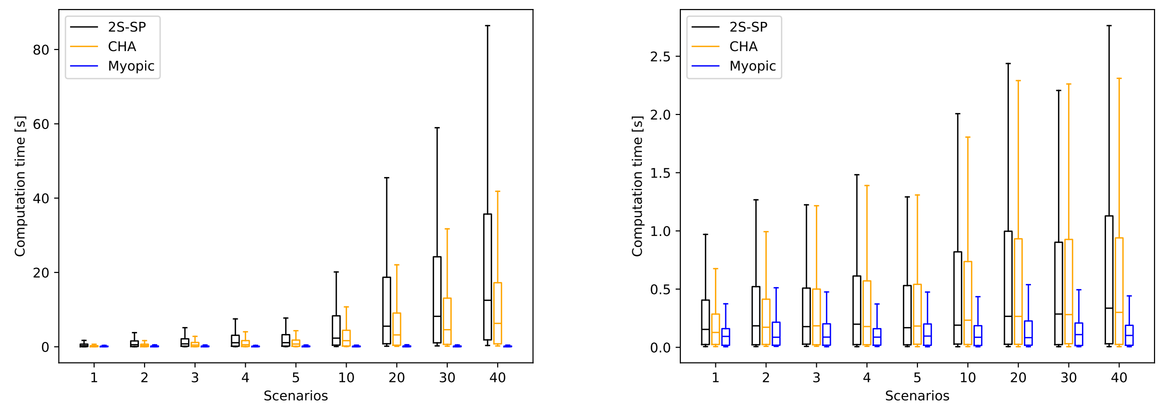
As expected, Table 3 shows a nearly linear increase in serial computation times for our 2S-SP and CHA approaches for Route 55. We also expected nearly constant parallel computation times in our 2S-SP and CHA approaches over the different numbers of scenarios. This is the case for both our 2S-SP and CHA approaches. We also see that over all CSFs, even with a limited number of scenarios, our 2S-SP and CHA approaches improve the results of our Myopic approach by more than four percent. Even with an increase in scenarios, the solution quality of our 2S-SP and CHA approaches does not increase significantly.

Figure 10 shows the distribution for the development of serial and parallel computation times for Route 55. Here, we see an increase in serial computation times and serial computation time variances when the number of scenarios increases, especially with our 2S-SP approach. As our Myopic approach is independent of the number of scenarios, its computation times remain constant.

Table 3: Comparison between 2S-SP, CHA, and Myopic with different number of future scenarios Route 55

	2S-SP				CHA				Myopic		
Scenarios	Profit	SP	ser.	par.	Profit	SP	ser.	par.	Profit	SP	ser.
5	22882.12	75.65	1.10	0.17	22793.50	75.37	0.73	0.18			
10	22905.57	75.74	2.32	0.19	22760.35	75.31	1.65	0.23			
20	22972.39	75.93	5.54	0.27	22778.48	75.25	3.19	0.27	21915.37	73.37	0.08
30	22980.84	75.81	7.76	0.27	22828.88	75.49	4.59	0.28			
40	23021.10	75.79	12.10	0.32	22842.33	75.46	6.29	0.30			

Figure 10: Development of serial and parallel computation times based on different numbers of future scenarios for Route 55



(a) Serial computation times **(b) Parallel computation times**

Apart from five scenarios, the parallel computation times and parallel computation time variances remain stable.

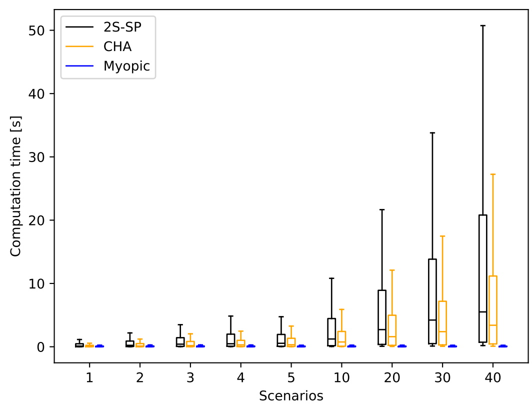
Table 4 confirms our observations of a nearly linear increase in serial computation times for our 2S-SP and GA approaches and nearly constant parallel computation times for Route 155. Additionally, we observe that a limited number of scenarios suffices for our 2S-SP and CHA approaches to determine better solutions than our Myopic approach. We elaborate on the quality of our operational planning problem policies under uncertainty in the next subsection, where we compare them to full information policies.

Figure 11 shows that for Route 155, the serial computation time variance of our 2S-SP approach increase more with a growing number of scenarios than for our CHA approach. Contrary to our observations for Route 55, we see that for Route 155, the parallel computation times and parallel computation time variances in our 2S-SP approach slightly increase with a growing number of scenarios. Starting from 20 scenarios, the parallel computation times and parallel computation times variances stay nearly constant for our CHA approach. In both Routes 55 and 155, we see that the parallel computation times of our CHA approach are lower than our 2S-SP approach. This suggests that our subproblems are easier solvable without fixing the decision of a current request.

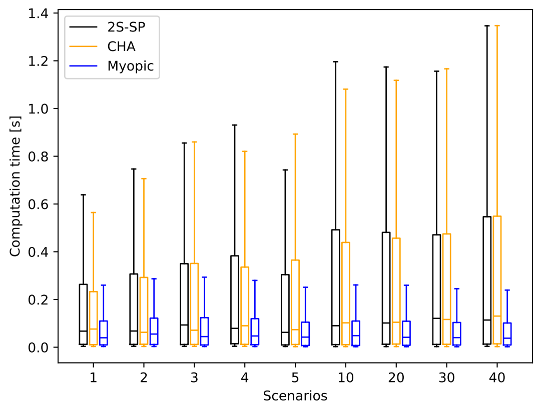
Table 4: Comparison between 2S-SP, CHA, and Myopic with different number of future scenarios Route 155

	2S-SP				CHA				Myopic		
Scenarios	Profit	SP [%]	ser. [s]	par. [s]	Profit	SP	ser.	par.	Profit	SP	ser.
5	17283.31	82.47	0.55	0.06	17201.22	82.12	0.32	0.07			
10	17259.57	82.37	1.23	0.09	17237.89	82.22	0.76	0.10			
20	17267.28	82.42	2.71	0.10	17228.35	82.22	1.59	0.10	16852.43	81.01	0.04
30	17264.25	82.47	4.22	0.12	17245.32	82.27	2.39	0.12			
40	17280.81	82.52	5.51	0.11	17260.91	82.32	3.40	0.13			

Figure 11: Development of serial and parallel computation times based on different numbers of scenarios for Route 155



(a) Serial computation times



(b) Parallel computation times

Observation 4. *The serial computation times of our 2S-SP and CHA approaches grow nearly linear with the number of scenarios. Additionally, the median parallel computation time stays constant with a growing number of scenarios. Note that this comes at the expense of additional memory usage. As our Myopic approach is independent of the number of scenarios, its serial computation times remain constant.*

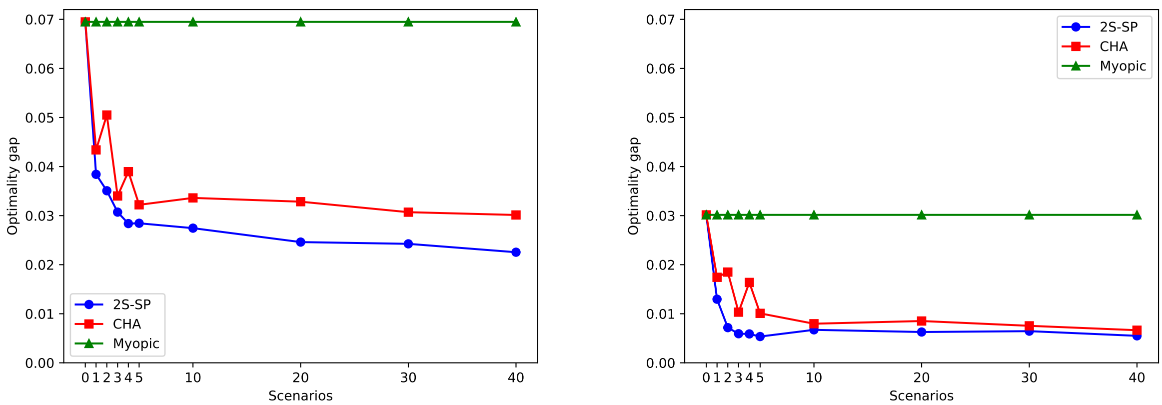
Observation 5. *The variance in parallel computation time is lower for our CHA compared to our 2S-SP approach. Accordingly, not fixing the decision of whether to accept or reject a passenger request in our subproblems leads to more stable computation times.*

5.5.1.4 Performance analysis of our approaches

Our 2S-SP and CHA models use a set of scenarios to approximate the distribution of future passenger requests. In this context, one major question that arises is how many scenarios are necessary to approximate future demands so that we achieve a good approximation of future passenger demands and thus can maximize the operators' profit accordingly. To answer this question, we use our results from the previous section and compare them against the optimal full information policy obtained through Problem 5.3.1. Specifically, we compare our solutions obtained in a dynamic stochastic problem setting to the deterministic problem setting in which all requests are revealed at the same time.

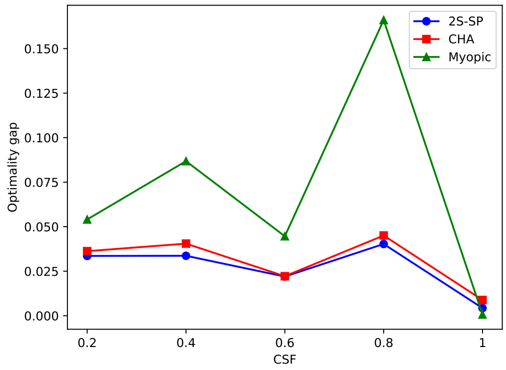
Figure 12 shows the optimality gap, i.e., *1 - (objective of our policy under uncertainty / objective of full information policy)*, between our policies under uncertainty and optimal full information policies based on the number of scenarios for Routes 55 and 155 over all CSF configurations. Note that our Myopic approach is independent of the number of scenarios. We see that even with only five scenarios, both our 2S-SP and CHA approaches obtain good solutions. Additionally, we see that increasing the number of scenarios to more than five shows no significant improvements.

Figure 12: Difference in profit based on the number of scenarios for Route 55 and 155

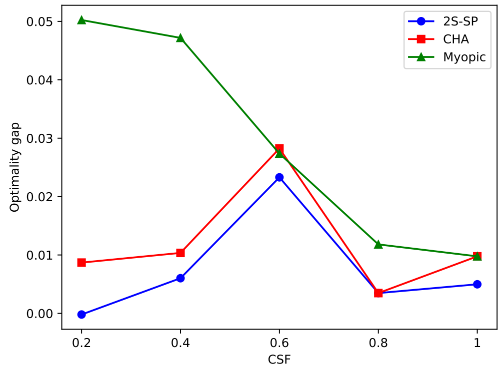


(a) **Optimality gap for Route 55** (b) **Optimality gap for Route 155**

Figure 13: Difference in profit based on varying CSFs for Route 55 and 155



(a) Profit for Route 55



(b) Profit for Route 155

Figure 13 shows the difference in profit for varying CSFs for Routes 55 and 155 over all number of scenarios compared to the optimal full information policy. We see that especially for routes with higher flexibility, i.e., CSF of 0.2 and 0.4 our 2S-SP and CHA approaches perform much better than our Myopic approach. Additionally, we also see that for very low and very high route flexibility, i.e., CSF of 0.2, 0.8, and 1, especially our 2S-SP and CHA approaches are close to the optimal full information policy. The reason for that is that with very high flexibility, most of the passengers can be served, and with very low flexibility, the route is mostly fixed. Both cases are relatively easy to handle. As a DAS combines flexibility with predictability, especially a CSF between $0.4 - 0.8$ is of interest. Here, our Myopic approach struggles to find good policies for Route 55.

Observation 6. *Even with a limited amount of scenarios, our 2S-SP and CHA approaches can determine good policies under uncertainty. Especially in high and low flexibility instances, our 2S-SP and CHA approaches perform better than our Myopic approach. Additionally, with decreasing flexibility, all our approaches can determine close to optimal full information policies.*

5.5.2 Practical insights

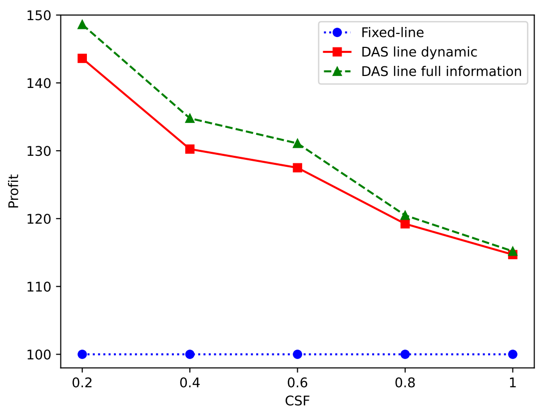
From the perspective of a public transportation system operator, a natural question is whether it is beneficial to convert a fixed-line bus route to a DAS. To answer this question, our analysis in this section is twofold. First, we compare the average profits, percentages of passengers served, and travel times between the fixed-line bus routes, a DAS under uncertainty, and a DAS in the full information setting. Second, we analyze the impact of passenger willingness to walk and route flexibility on profit, served passengers, and the cost per passenger.

5.5.2.1 Comparison between fixed-line and a DAS

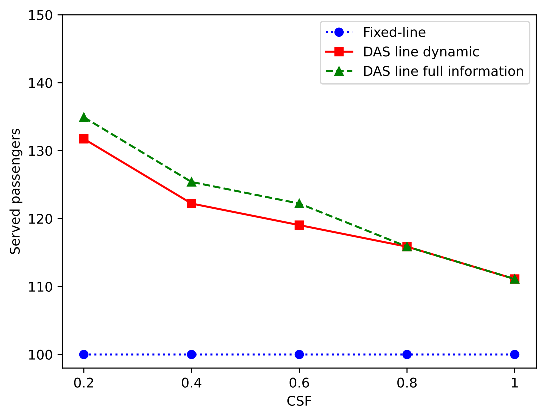
One of our main motivations for the operational planning problem under uncertainty is to be able to compare fixed-line public bus transportation systems with a DAS on an operational level. To do so, we solve Problem 5.3.1 with fixed arcs between the compulsory stops for a CSF of one, on five different instances. This gives us average profits, percentages of passengers served, and travel times for a fixed-line bus transportation system. Afterward, we determine the same metrics for the full information DAS setting through Problem 5.3.1 without fixed arcs and for the DAS setting under uncertainty through Problem 5.4.4. In all three settings, the walking distance is 250 meters, the utility is 750, and in the two DAS settings, the CSF $\in [0.2, 0.4, 0.6, 0.8, 1]$. Note that we only consider the fixed-line bus setting over the compulsory stops of the original routes, i.e., only for a CSF of one.

Figure 14 shows the comparison between fixed-line and DAS settings. Here, the metrics of the solution of the fixed-line setting are scaled to the baseline of 100, e.g., a profit of 115 for the dynamic DAS setting constitutes a 15% increase in profit compared to the fixed-line setting. We see that even for a CSF of one, both DAS settings generate 15% more profit, serve 11% more passengers,

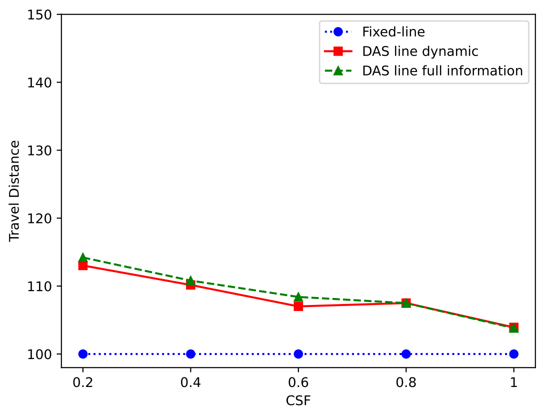
Figure 14: Increase in profit, served passengers, and travel times compared to a fixed-line baseline setting as 100 for Route 55



(a) Difference in profits between fixed-line and a DAS settings



(b) Difference in served passengers between fixed-line and a DAS settings



(c) Difference in travel times between fixed-line and a DAS settings

while having only 4% more travel time. With growing flexibility, i.e., a decrease in CSFs, these metrics increase to 44 - 49% more profit, serving 32 - 35% more passengers, with an increase of 13 - 14% in travel time. Over all CSF, we see that in our route, an increase of one percent in travel distance leads to an increase of 2.7 - 4.7% in profits and 2.2 - 3.7% in passengers served in our setting under uncertainty. Additionally, an increase of one percent in travel distance leads to an increase of 2.9 - 5% in profits and 2.3 - 3.7% in passengers served in our full information setting. An analysis of Route 155 yields similar results.

Observation 7. *Switching from a fixed-line setting to a DAS increases profits and passengers served at the cost of longer travel distances of the bus and thus its passengers. With a travel distance increase of one percent, a DAS can achieve up to 4.7% more profit, serving up to 3.7% more passengers in a setting under uncertainty.*

5.5.2.2 Impact of passenger walking willingness and route flexibility

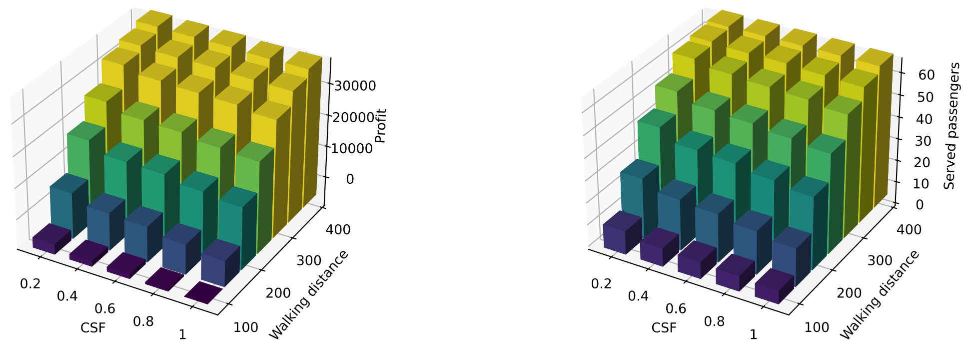
Our model allows us to analyze the number of passengers served and costs per served passenger based on different DAS assumptions, i.e., different CSFs, which indicates the degree of flexibility and the walking distance passengers are willing to walk to and from stops. Our analysis in this section is twofold. First, we analyze the impact of those two parameters on the number of passengers served and achievable profit. Second, we investigate how CSF and walking distance influence the cost per passenger served. To do so, we compare the solutions of our exact approach with five scenarios for a uniform passenger utility of 750, a CSF $\in [0.2, 0.4, 0.6, 0.8, 1]$, and a walking distance $\in [100, 150, 250, 300, 350, 400]$ on Route 55 and 155.

Profit and passengers served

Our model presumes a CSF indicating the flexibility of the DAS line and a threshold distance passengers are willing to walk to nearby stops. To understand which parameter drives profit and the number of served passengers, we aggregate results on instances according to the previously mentioned parameter values for the CSF and the walking distance.

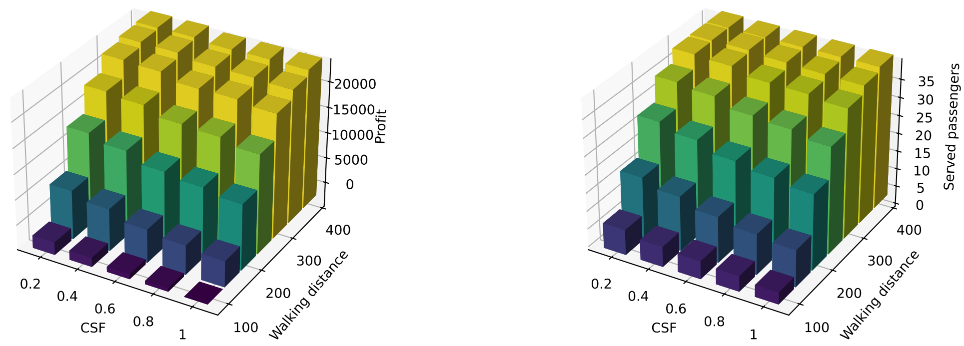
Figure 15 shows that for Route 55, the profit increases from -8,549 to 36,961 with increasing route flexibility and passenger walking willingness, and the number of served passengers increases from 6 to 65.1. Here, walking distances larger than 300 meters allow the operator to serve between 86-100% of all passenger requests. We see that especially for walking distances up to 250 meters, decreasing the CSF from 1 to 0.2 increases the profit by 59% from 11,453 to 18,221 and the number of served passengers by 19% from 45.4 to 53.8. At the same time, increasing the walking distance from 100 to 250 meters increases the profit from -7,133 to 22,906, and passengers served by 620% from 7.9 to 49.1. This shows that even though a higher degree of flexibility allows an operator to serve more passengers and thus increase profit, incentivizing passengers to walk slightly larger distances to and from their stops represents a larger lever to increase profit.

Figure 15: Development of operator profit and number of served passengers based on route flexibility (CSF) and walking distance for Route 55



(a) Average operator profit **(b) Average number of served passengers**

Figure 16: Development of operator profit and number of served passengers based on route flexibility (CSF) and walking distance for Route 155



(a) Average operator profit **(b) Average number of served passengers**

Figure 16 shows that our previous observations also holds for Route 155. We see that with increasing route flexibility and passenger walking willingness, the profit increases from -4,283 to 23,812, and the number of served passengers increases from 3.3 to 39.7. Here, walking distances larger than 300 meters allow the operator to serve between 91-100% of all issued passenger requests. We also see that for shorter walking distances up to 250 meters, decreasing the CSF from 1 to 0.2 increases the profit by 25% from 15,283 to 19,127 and the number of served passengers by 24% from 28.6 to 35.6. At the same time, increasing the walking distance from 100 to 250 meters increases the profit from -3,093 to 17,230, and passengers served by 654% from 5.0 to 32.7.

Observation 8. *With increasing route flexibility and a higher willingness to walk, operators can serve more passengers and generate higher profits. Solely increasing route flexibility does not suffice to significantly improve the number of served passengers. To achieve this, incentivizing passengers to walk slightly longer distances to and from their stops, e.g., through dynamic pricing, is necessary.*

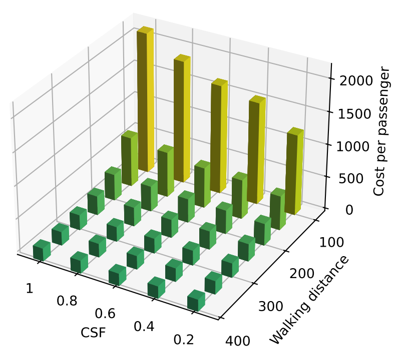
Cost per passenger

Apart from operator profit and the number of served passengers, we additionally want to analyze the development of the cost per passenger based on varying passenger walking willingness and route flexibility. To do so, we define the cost per passenger as the routing cost of the vehicle divided by the number of served passengers.

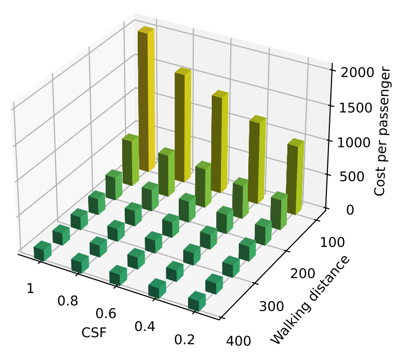
Figure 17 shows that by increasing the passenger walking distance and the flexibility of the route, the cost per passenger for the two routes reduces from 2,175 to 182 and 2,048 to 150, respectively. We see that even though the flexibility of the route reduces the cost per passenger between 3-43% and 5-51%, especially for lower walking distances under 250 meters, the passengers' willingness to walk has an even higher impact on the cost per passenger. Here, we see that increasing the walking distance from 100 to 250 meters reduces the cost per passenger for the two routes on average by 83% from 1,712 to 284 and by 85% from 1,449 to 223, respectively. Generally, over the different CSF configurations, the cost per passenger in both routes reduces between 88-90% for a walking distance increase from 100 to 400 meters.

Observation 9. *If passengers are willing to walk 250 meters instead of 100 meters to and from stops, the cost per passenger reduces by 83-85%. Increasing the flexibility of a route by reducing the CSF from 1 to 0.2 only reduces the cost per passenger by 43-51%.*

Figure 17: Development of cost per passenger based on route flexibility (CSF) and walking distance



(a) Route 55



(b) Route 155

5.6 Conclusion and future work

We introduced the operational planning problem under uncertainty of a Demand Adaptive System. Furthermore, we proposed an algorithmic framework that allows an operator to plan in real-time which passengers to serve in a Demand Adaptive System line. To this end, we formulated the operational route planning problem's full information version as a mixed-integer program, modeled the operational route planning problem under uncertainty as a Markov decision process, and utilized a rolling horizon approach to compute a policy via a two-stage stochastic program in each timestep to decide whether to accept or reject a passenger. Furthermore, we determined the deterministic equivalent of our approximation through sample-based approximation. This allowed us to decompose the deterministic equivalent of our two-stage stochastic program into several full information planning problems that can be solved in parallel efficiently. Additionally, we proposed a heuristic and a myopic approach.

We performed extensive computational experimentation to analyze how our different algorithms perform on different routes, the number of future passenger demand scenarios, passenger walking willingness, and route flexibility. For this purpose, we generated a large set of instances based on real-world data provided to us by Stadtwerke München GmbH, a Munich public transportation provider. In a comparison between full information policies and policies under uncertainty, we showed that in our case study, a limited amount of future request scenarios suffices to obtain good policies under uncertainty, especially in routes with very low and high flexibility.

From a managerial perspective, we show that by switching a fixed-line bus route to a Demand Adaptive System, an operator can increase revenue by up to 49% and the number of served passengers by up to 35% while only increasing the travel distance of the bus by 14%. Furthermore, we showed that even though we can achieve an increase in passengers served between 19-24% by making a route more flexible, increasing passengers' willingness to walk from 100 to 250 meters increases this number by 620-654%. Additionally, such an increase in passenger willingness to walk also reduced the cost per passenger served by 83-85%, while an increase in the flexibility of the route only reduced it by 43-51%.

Furthermore, we showed that our exact decomposition consistently derives the best solutions in terms of operator revenues and served passengers, followed by our heuristic approach and, lastly, our myopic approach. This performance comes at the cost of higher serial computation time and doubles the resources needed in parallel computation compared to our heuristic approach, which is 17-57% faster. Additionally, we saw an increase in problem complexity, i.e., serial computation time with growing route flexibility.

Future research may expand on the current methodology of our two-stage stochastic program by developing a machine-learning enhanced policy that reduces the computational effort during operations. Additionally, our current passenger distribution is based on the assumption that passengers utilizing a Demand Adaptive System have a similar demand distribution as passengers utilizing fixed-line bus systems. Investigating the impact of different passenger request distributions on the operators' revenue remains an interesting avenue for future research. Finally, considering passenger behavior, e.g., dynamic pricing models that influence passenger acceptance or rejection decisions, poses an interesting area for future study.

Acknowledgment

This work was supported by the German Federal Ministry of Education and Research under Grant 03ZU1105FA. Additionally, we thank Stadtwerke München GmbH for the pleasant collaboration and for providing us with the necessary data.

6 Conclusion

6.1 Main contributions

Intermodal public transportation systems are an essential lever for providing an efficient and cost-effective way to meet people's growing travel demands. Designing and operating today's public transportation systems requires careful planning from strategic, tactical, and operational perspectives to allow for an integration of more flexible transportation services, such as Demand Adaptive Systems (DAS). Additionally, the integration of stochasticity into the planning processes, instead of solely focusing on average passenger demands, allows an operator to consider a broader range of demand scenarios, leading to more robust planning decisions. Against this background, this thesis contributes novel models and solution methods that enable intermodal public transportation operators to conduct an in-depth analysis of optimal passenger flows and consider stochasticity in the tactical and operational planning of DAS.

Chapter 3 introduced an algorithmic framework to determine the passenger flow system optimum of an intermodal transportation system. To do so, we modeled a transportation system as a partially time-expanded digraph, which implicitly encodes problem-specific constraints related to the intermodality of the problem. This allowed us to find the intermodal transportation system's passenger flow optimum by solving a standard integer minimum-cost Multi-Commodity Network Flow problem. We solved this problem with a column generation-based Price-and-Branch approach, for which we introduced a pricing filter to reduce the number of solved pricing problems. Furthermore, we proposed a distance approximation to utilize the A^* algorithm to solve the pricing problems more efficiently. We applied our methodology to a real-world case study for the city of Munich. Our results show that our pricing filter reduces the computation time of our algorithm by up to 60%, and the application of the A^*-based pricing algorithm additionally reduces it by up to 90%. Our algorithm is able to solve unimodal instances with 23,688 trips in 11 minutes and intermodal instances with 56,295 trips in 48 minutes, whereas straightforward mixed-integer program (MIP) models fail to solve instances with more than a few hundred trips. Our algorithmic framework allows us to analyze the passenger utilization of individual routes based on the calculated passenger flows which is helpful for short-term network design decisions.

Chapter 4 introduced the stochastic TSP with generalized latency (STSP-GL), which can be used to find a general ordering of stops in the tactical planning of DAS under uncertain passenger demand. Furthermore, we proposed an algorithmic framework to determine near-optimal solutions. To do so, we modeled the STSP-GL as a stochastic program and determined its deterministic equivalent. This allowed us to utilize a Branch-and-Price (B&P), hybrid, and heuristic approach, which decompose the STSP-GL into multiple TSP-GLs. We performed extensive computational experimentation to analyze how our different algorithms perform on different graph sizes, number of scenarios, service, and feasibility levels. For this purpose, we generated a large set of instances from the TSPGL_LIB dataset. We showed that our B&P and hybrid approaches find better upper bounds than our MIP benchmark within 500 seconds for nearly all instance types but struggle with closing the optimality gap. Especially for larger graph sizes, our B&P and hybrid approaches improved the MIP benchmark's upper bounds by up to 85%. Furthermore, we showed that our heuristic approach finds feasible solutions within a couple of seconds, but due to the size of the set of minimal feasibility covers, is not able to significantly improve on the initial upper bounds during the remainder of the computation time. Furthermore, we showed that even though the design costs of solutions derived through a deterministic approach are lower compared to solutions of a stochastic approach, they are not feasible in a stochastic setting, i.e., the stochastic approach includes more nodes in the tours to meet the demand with a higher probability. Finally, we showed that the impact of the service level on the design cost is higher than the impact of the feasibility level. Here, we were able to reduce the design costs by 8.5% on average when lowering the service level by 5% while only reducing the design costs by 2% when lowering the feasibility frequency by 5%.

Chapter 5 introduced the operational planning problem under uncertainty of a DAS. Furthermore, we proposed an algorithmic framework that allows an operator to plan which passengers to serve in a DAS line in real-time. To this end, we formulated the operational route planning problem's full information version as a mixed-integer program, modeled the operational route planning problem under uncertainty as a Markov decision process, and utilized a rolling horizon approach to approximate the Markov decision process via a two-stage stochastic program in each timestep to decide whether to accept or reject a passenger. Furthermore, we determined the deterministic equivalent of our approximation through sample-based approximation. This allowed us to decompose the deterministic equivalent of our two-stage stochastic program into several full information planning problems, which we showed how to solve in parallel efficiently. Additionally, we proposed a heuristic approach and a myopic approach.

We performed extensive computational experimentation to analyze how our different algorithms perform on different routes, the number of future passenger demand scenarios, passenger walking willingness, and route flexibility. For this purpose, we generated a large set of instances based on real-world data provided to us by Stadtwerke München GmbH, a Munich public transportation provider. In a comparison between full information policies and policies under uncertainty, we showed that in our case study, a limited amount of future request scenarios suffices to obtain good policies under uncertainty, especially in routes with very low and high flexibility.

From a managerial perspective, we show that by switching a fixed-line bus route to a DAS, an operator can increase revenue by up to 49% and the number of served passengers by up to 35% while only increasing the travel distance of the bus by 14%. Furthermore, we showed that even though we can achieve an increase in passengers served between 19-24% by making a route more flexible, increasing passengers' willingness to walk from 100 to 250 meters increases this number by 620-654%. Additionally, such an increase in passenger willingness to walk also reduced the cost per passenger served by 83-85%, while an increase in the flexibility of the route only reduced it by 43-51%.

Furthermore, we showed that our exact decomposition consistently derives the best solutions in terms of operator revenues and served passengers, followed by our heuristic approach and, lastly, our myopic approach. This performance comes at the cost of higher serial computation time and doubles the resources needed in parallel computation compared to our heuristic approach, which is 17-57% faster. Additionally, we saw an increase in problem complexity, i.e., serial computation time, with growing route flexibility.

In summary, this thesis contributes to new models, methods, and managerial insights into relevant real-world problems closely related to the analysis of intermodal transportation systems and the tactical and operational planning of DASs. Specifically, it focused on finding the travel time system optimum of passenger flows in a large-scale intermodal transportation system, proposed an efficient decomposition to solve the STSP-GL, a problem arising in the tactical planning of DAS, and developed an algorithmic framework to solve the operational planning problem under uncertainty of a DAS.

6.2 Limitations and future research

Although this thesis provides comprehensive algorithmic frameworks that address optimizing passenger flows in large-scale intermodal transportation systems, solving the STSP-GL, and the operational planning problem under uncertainty of a DAS, some problem cases remained out of the scope of this thesis.

First, even though our P&B framework in Chapter 3 finds optimal solutions for all instances of our case study, it does not possess an optimality guarantee when deciding integer solutions. To this end, extending our framework to a B&P framework that guarantees optimality for integer solutions remains an interesting avenue for future research. Moreover, our current problem setting does not consider additional path constraints, e.g., limits on passenger transfers, which remains another interesting direction for future research. Lastly, we implemented our column generation approach, including our A^* algorithm, in Python. A re-implementation in Julia or C++, which allows for an efficient parallelization of the pricing problem, may speed up this algorithmic framework significantly.

Second, while the hybrid approach in Chapter 4 determines good upper bounds, closing the optimality gap remains a challenging task. This may be tackled with a more sophisticated heuristic approach to determine promising feasibility covers. Additionally, our current distribution of requests is nearly uniform, i.e., the probability of request A having positive demand in a scenario is similar to request B having positive demand in a scenario. Investigating the impact of different request distributions on the design and routing costs remains an interesting avenue for future research.

Third, in Chapter 5 research may expand on the current methodology of our two-stage stochastic program by developing a machine-learning enhanced policy that reduces the computational effort during operations. Additionally, our current passenger distribution is based on the assumption that passengers utilizing a DAS have a similar demand distribution as passengers utilizing fixed-line bus systems. Investigating the impact of different passenger request distributions on the operators' revenue remains an interesting avenue for future research. Finally, considering passenger behavior, e.g., dynamic pricing models that influence passenger acceptance or rejection decisions, poses an interesting area for future study.

Bibliography

2015. UN Agenda. `https://sdgs.un.org/2030agenda`. Accessed: 2024-01-29.

2020. VDV Die Verkehrsunternehmer. `https://www.vdv.de/mobilitaet-im-laendlichen-raum-staerken.aspx`. Accessed: 2025-01-12.

Agatz, N., A.L. Erera, M.W.P. Savelsbergh, X. Wang. 2011. Dynamic ride-sharing: A simulation study in metro Atlanta. *Procedia-Social and Behavioral Sciences* **17** 532–550.

Al-Kanj, L., J. Nascimento, W. Powell. 2020. Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. *European Journal of Operational Research* **284**(3) 1088–1106.

Al-Khayyal, F., S-J Hwang. 2007. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. *European Journal of Operational Research* **176**(1) 106–130.

Alavidoost, MH, M. Tarimoradi, MH Zarandi. 2018. Bi-objective mixed-integer nonlinear programming for multi-commodity tri-echelon supply chain networks. *Journal of intelligent manufacturing* **29**(4) 809–826.

Alonso-González, M. J., O. Cats, N. van Oort, S. Hoogendoorn-Lanser, S. Hoogendoorn. 2021. What are the determinants of the willingness to share rides in pooled on-demand services? *Transportation* **48**(4) 1733–1765.

Alonso-Mora, J., S. Samaranayake, A. Wallar, E. Frazzoli, D. Rus. 2017a. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* **114**(3) 462–467.

Alonso-Mora, J., A. Wallar, D. Rus. 2017b. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3583–3590.

Amberg, B., B. Amberg, N. Kliewer. 2019. Robust efficiency in urban public transportation: Minimizing delay propagation in cost-efficient bus and driver schedules. *Transportation Science* **53**(1) 89–112.

Atamtürk, A. 2001. Flow pack facets of the single node fixed-charge flow polytope. *Operations Research Letters* **29**(3) 107–114.

Atamtürk, A. 2002. On capacitated network design cut–set polyhedra. *Mathematical Programming* **92** 425–437.

Atamtürk, A., M. Zhang. 2007. Two-stage robust network flow and design under demand uncertainty. *Operations Research* **55**(4) 662–673.

Ayar, B., H. Yaman. 2012. An intermodal multicommodity routing problem with scheduled services. *Computational optimization and applications* **53**(1) 131–153.

Azi, N., M. Gendreau, J.-Y. Potvin. 2012. A dynamic vehicle routing problem with multiple delivery routes. *Annals of Operations Research* **199** 103–112.

Banerjee, S., C. Hssaine, Q. Luo, S. Samaranayake. 2021. Plan your system and price for free: Fast algorithms for multimodal transit operations. *Available at SSRN 3972423* .

Barnhart, C., C. A. Hane, E. L. Johnson, G. Sigismondi. 1994. A column generation and partitioning approach for multi-commodity flow problems. *Telecommunication Systems* **3**(3) 239–258.

Barnhart, C., C.A. Hane, P.H. Vance. 2000. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research* **48**(2) 318–326.

Barnhart, C., E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations Research* **46**(3) 316–329.

Beliën, J., E. Demeulemeester. 2008. A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research* **189**(3) 652–668.

Bertsimas, D., S. S. Patterson. 2000. The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science* **34**(3) 239–255.

Birge, J., F. Louveaux. 2011. *Introduction to Stochastic Programming*. Springer.

Birge, J. R. 1982. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical Programming* **24** 314–325.

Bischoff, J., K. Führer, M. Maciejewski. 2019. Impact assessment of autonomous DRT systems. *Transportation Research Procedia* **41** 440–446.

Bischoff, J., M. Maciejewski. 2016. Simulation of city-wide replacement of private cars with autonomous taxis in Berlin. *Procedia computer science* **83** 237–244.

Boland, N., M. Hewitt, L. Marshall, M. Savelsbergh. 2017. The continuous-time service network design problem. *Operations Research* **65** 1303–1321.

Branchini, R.M., V.A. Armentano, A. Løkketangen. 2009. Adaptive granular local search heuristic for a dynamic vehicle routing problem. *Computers & Operations Research* **36**(11) 2955–2968.

Branke, J., M. Middendorf, G. Noeth, M. Dessouky. 2005. Waiting strategies for dynamic vehicle routing. *Transportation science* **39**(3) 298–312.

Brunetta, L., M. Conforti, M. Fischetti. 2000. A polyhedral approach to an integer multicommodity flow problem. *Discrete Applied Mathematics* **101**(1-3) 13–36.

Canel, C., B. Khumawala, J. Law, A. Loh. 2001. An algorithm for the capacitated, multi-commodity multi-period facility location problem. *Computers & Operations Research* **28**(5) 411–427.

Chouman, M., T. G. Crainic, B. Gendron. 2017. Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science* **51**(2) 650–667.

Cordeau, J.F., G. Laporte. 2007. The dial-a-ride problem: models and algorithms. *Annals of Operations Research* **153**(1) 29–46.

Costa, A. 2005. A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research* **32**(6) 1429–1450.

Costa, A., J.-F. Cordeau, B. Gendron. 2009. Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications* **42** 371–392.

Costa, A., J.-F. Cordeau, B. Gendron, G. Laporte. 2012. Accelerating Benders decomposition with heuristicmaster problem solutions. *Pesquisa Operacional* **32** 03–20.

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

Crainic, T. G. 2019. Parallel metaheuristics and cooperative search. M. Gendreau, J.-Y. Potvin, eds., *Handbook of Metaheuristics*. Springer, 419–451.

Crainic, T. G., F. Errico, F. Malucelli, M. Nonato. 2012. Designing the master schedule for demand-adaptive transit systems. *Annals of Operations Research* **194**(1) 151–166.

Crainic, T. G., A. Frangioni, B. Gendron. 2001. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics* **112**(1-3) 73–99.

Crainic, T. G., M. Gendreau. 2007. A scatter search heuristic for the fixed-charge multicommodity flow network design problem. *Metaheuristics: Progress in Complex Systems Optimization* 25–40.

Crainic, T. G., M. Gendreau, J. M. Farvolden. 2000. A simplex-based tabu search method for capacitated network design. *INFORMS Journal on Computing* **12**(3) 223–236.

Crainic, T. G., M. Gendreau, B. Gendron. 2021. Fixed-Charge Network Design Problems. T. G. Crainic, M. Gendreau, B. Gendron, eds., *Network Design with Applications to Transportation and Logistics*. Springer, 15–28.

Crainic, T. G., Y. Li, M. Toulouse. 2006. A first multilevel cooperative algorithm for capacitated multicommodity network design. *Computers & Operations Research* **33**(9) 2602–2622.

Crainic, T.G., X. Fu, M. Gendreau, W. Rei, S. Wallace. 2011. Progressive hedging-based metaheuristics for stochastic network design. *Networks* **58**(2) 114–124.

Crainic, T.G., F. Malucelli, M. Nonato, F. Guertin. 2005. Meta-heuristics for a class of demand-responsive transit systems. *INFORMS Journal on Computing* **17**(1) 10–24.

Daganzo, C. 1984. Checkpoint dial-a-ride systems. *Transportation Research Part B: Methodological* **18**(4-5) 315–327.

d'Antonio, G., A. Frangioni. 2009. Convergence analysis of deflected conditional approximate subgradient methods. *SIAM Journal on Optimization* **20**(1) 357–386.

Dantzig, G., P. Wolfe. 1960. Decomposition principle for linear programs. *Operations Research* **8**(1) 101–111.

Delahaye, D., S. Chaimatanan, M. Mongeau. 2019. Simulated annealing: From basics to applications. M. Gendreau, J.-Y. Potvin, eds., *Handbook of Metaheuristics*. Springer, 1–35.

Dijkstra, E. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* **1**(1) 269–271.

Erera, A.L., J.C. Morales, M. Savelsbergh. 2005. Global intermodal tank container management for the chemical industry. *Transportation Research Part E: Logistics and Transportation Review* **41**(6) 551–566.

Erhardt, G. D., S. Roy, D. Cooper, B. Sana, M. Chen, J. Castiglione. 2019. Do transportation network companies decrease or increase congestion? *Science advances* **5**(5) 1–11.

Errico, F. 2008. The design of flexible transit systems: Models and solution methods. Ph.D. thesis, Polytechnic University of Milan, Italy.

Errico, F., T. G. Crainic, F. Malucelli, M. Nonato. 2011a. The design problem for single-line demand adaptive transit systems. *CIRRELT-2011-65, Centre interuniversitaire de recherche sur les réseaux dentreprise, la logistique et le transport, Université de Montréal* .

Errico, F., T. G. Crainic, F. Malucelli, M. Nonato. 2011b. An unifying framework and review of semi-flexible transit systems. CIRRELT.

Errico, F., T. G. Crainic, F. Malucelli, M. Nonato. 2013. A survey on planning semi-flexible transit systems: Methodological issues and a unifying framework. *Transportation Research Part C: Emerging Technologies* **36** 324–338.

Errico, F., T. G. Crainic, F. Malucelli, M. Nonato. 2017. A Benders decomposition approach for the symmetric TSP with generalized latency arising in the design of semi-flexible transit systems. *Transportation Science* **51**(2) 706–722.

Errico, F., T.G. Crainic, F. Malucelli, M. Nonato. 2021. The single-line design problem for demand-adaptive transit systems: a modeling framework and decomposition approach for the stationary-demand case. *Transportation Science* **55**(6) 1300–1321.

Farahani, R., E. Miandoabchi, W. Szeto, H. Rashidi. 2013. A review of urban transportation network design problems. *European Journal of Operational Research* **229**(2) 281–302.

Fischetti, M., A. Lodi. 2003. Local branching. *Mathematical Programming* **98** 23–47.

Ford Jr, L.R., D.R. Fulkerson. 1958a. Constructing maximal dynamic flows from static flows. *Operations Research* **6**(3) 419–433.

Ford Jr, L.R., D.R. Fulkerson. 1958b. A suggested computation for maximal multi-commodity network flows. *Management Science* **5**(1) 97–101.

Frangioni, A., G. Gallo. 1999. A bundle type dual-ascent approach to linear multicommodity min-cost flow problems. *INFORMS Journal on Computing* **11**(4) 370–393.

Frangioni, A., B. Gendron, E. Gorgone. 2017. On the computational efficiency of subgradient methods: a case study with Lagrangian bounds. *Mathematical Programming Computation* **9** 573–604.

Frangioni, A., E. Gorgone. 2014. Bundle methods for sum-functions with "easy" components: applications to multicommodity network design. *Mathematical Programming* **145** 133–161.

Gendreau, M., F. Guertin, J.-Y. Potvin, R. Séguin. 2006. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part C: Emerging Technologies* **14**(3) 157–174.

Gendreau, M., J.-Y. Potvin. 2019. Tabu search. M. Gendreau, J.-Y. Potvin, eds., *Handbook of Metaheuristics*. Springer, 37–55.

Gendron, B. 2011. Decomposition methods for network design. *Procedia-Social and Behavioral Sciences* **20** 31–37.

Gendron, B. 2019. Revisiting Lagrangian relaxation for network design. *Discrete Applied Mathematics* **261** 203–218.

Gendron, Bernard, Teodor Gabriel Crainic, Antonio Frangioni. 1999. Multicommodity capacitated network design. *Telecommunications network planning*. Springer, 1–19.

Ghamlouche, I., T. G. Crainic, M. Gendreau. 2004. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals of Operations Research* **131** 109–133.

Ghamlouche, I., T.G. Crainic, M. Gendreau. 2003. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research* **51**(4) 655–667.

Gkiotsalitis, K., F. Alesiani. 2019. Robust timetable optimization for bus lines subject to resource and regulatory constraints. *Transportation Research Part E: Logistics and Transportation Review* **128** 30–51.

Goodson, J., B. Thomas, J. Ohlmann. 2017. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research* **258**(1) 216–229.

Graehler, M., R. A. Mucci, G. D. Erhardt. 2019. Understanding the recent transit ridership decline in major US cities: Service cuts or emerging modes. *98th Annual Meeting of the Transportation Research Board, Washington, DC*.

GTFS. 2022. GTFS data Germany. `https://gtfs.de/en/`. Accessed: 2022-08-10.

Gunnerud, V., B. Foss, K. McKinnon, B. Nygreen. 2012. Oil production optimization solved by piecewise linearization in a branch & price framework. *Computers & Operations Research* **39**(11) 2469–2477.

Haneveld, Klein, Maarten H Van der Vlerk. 2020. *Stochastic programming*. Springer.

Hewitt, M., G. L. Nemhauser, M. Savelsbergh. 2010. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing* **22**(2) 314–325.

Higle, J. L., S. W. Wallace. 2003. Sensitivity analysis and uncertainty in linear programming. *Interfaces* **33**(4) 53–60.

Hintermann, B., B. Schoeman, J. Molloy, T. Schatzmann, C. Tchervenkov, K. Axhausen. 2023. The impact of COVID-19 on mobility choices in Switzerland. *Transportation Research Part A: Policy and Practice* .

Ho, S., W. Szeto, Y.-H. Kuo, J. Leung, M. Petering, T. Tou. 2018. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological* **111** 395–421.

Horni, A., , K. Nagel, K.W. Axhausen. 2016. *The multi-agent transport simulation MATSim*. Ubiquity Press.

Hu, T. C. 1963. Multi-commodity network flows. *Operations Research* **11**(3) 344–360.

Hu, W. 2017. Your Uber car creates congestion. should you pay a fee to ride? *The New York Times* .

Huang, D., Y. Gu, S. Wang, Z. Liu, W. Zhang. 2020. A two-phase optimization model for the demand-responsive customized bus network design. *Transportation Research Part C: Emerging Technologies* **111** 1–21.

Huang, Z., B. Loo, K. Axhausen. 2023. Travel behaviour changes under work-from-home (WFH) arrangements during COVID-19. *Travel Behaviour and Society* **30** 202–211.

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

Ioachim, I., J. Desrosiers, Y. Dumas, M. Solomon, D. Villeneuve. 1995. A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science* **29**(1) 63–78.

Kall, P., S. Wallace. 1994. *Stochastic Programming*. Springer.

Katayama, N., S. Yurimoto. 2011. Combining capacity scaling and local branch approaches for the logistics network design problem. *Proceedings 21st International Conference on Production Research, ICPR 2011, Stuttgart, Germany.*

Khouja, M., Z. Michalewicz, M. Wilmot. 1998. The use of genetic algorithms to solve the economic lot size scheduling problem. *European Journal of Operational Research* **110**(3) 509–524.

Kustar, A., A. Davidson, T. Hein Tun, B. Welle, S. Muralidaran. 2024. A fare look: Funding urban public transport operations. `https://www.wri.org/research/fare-look-funding-urban-public-transport-operations`. Accessed: 2024-12-18.

Lanzetti, N., M. Schiffer, M. Ostrovsky, M. Pavone. 2021. On the interplay between self-driving cars and public transportation. *arXiv preprint arXiv:2109.01627* .

Lawler, E.L., D.E. Wood. 1966. Branch-and-bound methods: A survey. *Operations Research* **14**(4) 699–719.

Levin, M. W., K. M. Kockelman, S. D. Boyles, T. Li. 2017. A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application. *Computers, Environment and Urban Systems* **64** 373–383.

Li, J., M. Zhang. 2011. Fundamental diagram of traffic flow: new identification scheme and further evidence from empirical data. *Transportation research record* **2260**(1) 50–59.

Li, S., L. Liu, L. Yang, Z. Gao. 2019. Robust dynamic bus controls considering delay disturbances and passenger demand uncertainty. *Transportation Research Part B: Methodological* **123** 88–109.

Li, X., L. Quadrifoglio. 2009. Optimal zone design for feeder transit services. *Transportation Research Record* **2111**(1) 100–108.

Lienkamp, B., M. Hewitt, M. Schiffer. 2023. Branch-and-price for the stochastic TSP with generalized latency. *arXiv preprint arXiv:2311.09923* .

Lin, Y., W. Li, F. Qiu, H. Xu. 2012. Research on optimization of vehicle routing problem for ride-sharing taxi. *Procedia-Social and Behavioral Sciences* **43** 494–502.

Lin, Y-K. 2001. Study on the multicommodity reliability of a capacitated-flow network. *Computers & Mathematics with Applications* **42**(1-2) 255–264.

Litman, Todd. 2020. Autonomous vehicle implementation predictions: Implications for transport planning. *Victoria Transport Policy Institute* .

Lotfi, M. M., R. Tavakkoli-Moghaddam. 2013. A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. *Applied Soft Computing* **13**(5) 2711–2726.

Lourenço, H. R., O. C. Martin, T. Stützle. 2019. Iterated local search: Framework and applications. M. Gendreau, J.-Y. Potvin, eds., *Handbook of Metaheuristics*. Springer, 129–168.

Lübbecke, M.E. 2010. Column generation. *Wiley encyclopedia of Operations Research and management science. Wiley, New York* 1–14.

Ma, Q., S. Li, H. Zhang, Y. Yuan, L. Yang. 2021. Robust optimal predictive control for real-time bus regulation strategy with passenger demand uncertainties in urban rapid transit. *Transportation Research Part C: Emerging Technologies* **127** 103086.

Maciejewski, M., J. Bischoff, S. Hörl, K. Nagel. 2017. Towards a testbed for dynamic vehicle routing algorithms. *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 69–79.

Maggioni, F., S. W. Wallace. 2012. Analyzing the quality of the expected value solution in stochastic programming. *Annals of Operations Research* **200** 37–54.

Malucelli, F., M. Nonato, S. Pallottino. 1999. Demand adaptive systems: some proposals on flexible transit. *Operational Research in Industry*. Springer, 157–182.

Malucelli, Federico, Maddalena Nonato, Teodor Gabriel Crainic, Francois Guertin. 2001. Adaptive memory programming for a class of demand responsive transit systems. S. Voß, J.R. Daduna, eds., *Computer-Aided Scheduling of Public Transport*. Springer, 253–273.

Merlin, L. A. 2019. Transportation sustainability follows from more people in fewer vehicles, not necessarily automation. *Journal of the American Planning Association* **85**(4) 501–510.

Moeckel, R., N. Kuehnel, C. Llorca, A. T. Moreno, H. Rayaprolu. 2019. Microscopic travel demand modeling: using the agility of agent-based modeling without the complexity of activity-based models. *Annual Meeting of the Transportation Research Board*.

Moshref-Javadi, M., S. Lee. 2016. The customer-centric, multi-commodity vehicle routing problem with split delivery. *Expert Systems with Applications* **56** 335–348.

MVG. 2022. Vehicle data MVG. `https://www.mvg.de/ueber/das-unternehmen/fahrzeuge.html`. Accessed: 2022-08-10.

Ostrovsky, M., M. Schwarz. 2019. Carpooling and the economics of self-driving cars. *Proceedings of the 2019 ACM Conference on Economics and Computation*. 581–582.

Ouorou, A., P. Mahey, J.-P. Vial. 2000. A survey of algorithms for convex multicommodity flow problems. *Management Science* **46**(1) 126–147.

Padberg, M. W., T. J. Van Roy, L. A. Wolsey. 1985. Valid linear inequalities for fixed charge problems. *Operations Research* **33**(4) 842–861.

Paraskevopoulos, D., S. Gürel, T. Bektaş. 2016. The congested multicommodity network design problem. *Transportation Research Part E: Logistics and Transportation Review* **85** 166–187.

Pavone, Marco, Amin Saberi, Maximilian Schiffer, Matthew Tsao. 2020. Online hypergraph matching with delays. *Operations Research* .

Pei, M., P. Lin, J. Du, X. Li. 2019a. Operational design for a real-time flexible transit system considering passenger demand and willingness to pay. *IEEE Access* **7** 180305–180315.

Pei, M., P. Lin, J. Ou. 2019b. Real-time optimal scheduling model for transit system with flexible bus line length. *Transportation Research Record* **2673**(4) 800–810.

Pillac, V., M. Gendreau, C. Guéret, A. Medaglia. 2013. A review of dynamic vehicle routing problems. *European Journal of Operational Research* **225**(1) 1–11.

Pisinger, D. 2024. A consensus fixing heuristic for the stochastic prize collecting TSP. *EURO-2024 Copenhagen: 33rd European Conference on Operational Research*.

Powell, W. 2011. *Approximate Dynamic Programming: Solving the curses of dimensionality*. 2nd ed. John Wiley & Sons.

Pratelli, A., F. Schoen. 2001. A mathematical programming model for the bus deviation route problem. *Journal of the Operational Research Society* **52** 494–502.

Psaraftis, H. N. 2011. A multi-commodity, capacitated pickup and delivery problem: The single and two-vehicle cases. *European Journal of Operational Research* **215**(3) 572–580.

Pyakurel, U., T.N. Dhamala. 2017. Continuous dynamic contraflow approach for evacuation planning. *Annals of Operations Research* **253**(1) 573–598.

Qiu, F., J. Shen, X. Zhang, C. An. 2015. Demi-flexible operating policies to promote the performance of public transit in low-demand areas. *Transportation Research Part A: Policy and Practice* **80** 215–230.

Qiu, Feng, Wenquan Li, Jian Zhang. 2014. A dynamic station strategy to improve the performance of flex-route transit services. *Transportation Research Part C: Emerging Technologies* **48** 229–240.

Quadrifoglio, L., M. Dessouky, K. Palmer. 2007. An insertion heuristic for scheduling mobility allowance shuttle transit (MAST) services. *Journal of Scheduling* **10**(1) 25–40.

Quadrifoglio, L., X. Li. 2009. A methodology to derive the critical demand density for designing and operating feeder transit services. *Transportation Research Part B: Methodological* **43**(10) 922–935.

Retvdri, G., J. Biro, T. Cinkler. 2004. A novel lagrangian-relaxation to the minimum cost multicommodity flow problem and its application to ospf traffic engineering. *Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No. 04TH8769)*, vol. 2. IEEE, 957–962.

Riley, C., P. Van Hentenryck, E. Yuan. 2020. Real-time dispatching of large-scale ridesharing systems: Integrating optimization, machine learning, and model predictive control. *arXiv preprint arXiv:2003.10942* .

Rockafellar, T., R. Wets. 1991. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research* **16**(1) 119–147.

Rodríguez-Martín, I., J. J. Salazar-González. 2010. A local branching heuristic for the capacitated fixed-charge network design problem. *Computers & Operations Research* **37**(3) 575–581.

Rossi, F., R. Zhang, Y. Hindy, M. Pavone. 2018. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. *Autonomous Robots* **42**(7) 1427–1442.

Rossi, F., R. Zhang, M. Pavone. 2016. Congestion-aware randomized routing in autonomous mobility-on-demand systems. *Transp. Res. Part B Methodol* **99** 1–29.

Rüschendorf, L. 1995. Convergence of the iterative proportional fitting procedure. *The Annals of Statistics* 1160–1174.

Sadykov, R., F. Vanderbeck, A. Pessoa, I. Tahiri, E. Uchoa. 2019. Primal heuristics for branch and price: The assets of diving methods. *INFORMS Journal on Computing* **31**(2) 251–267.

Salazar, M., N. Lanzetti, F. Rossi, M. Schiffer, M. Pavone. 2019. Intermodal autonomous mobility-on-demand. *IEEE Transactions on Intelligent Transportation Systems* **21**(9) 3946–3960.

Salazar, M., F. Rossi, M. Schiffer, C. H. Onder, M. Pavone. 2018. On the interaction between autonomous mobility-on-demand and public transportation systems. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2262–2269.

Salimifard, K., S. Bigharaz. 2022. The multicommodity network flow problem: state of the art classification, applications, and solution methods. *Operational Research* **22**(1) 1–47.

Schiffer, M., G. Hiermann, F. Rüdel, G. Walther. 2021. A polynomial-time algorithm for user-based relocation in free-floating car sharing systems. *Transportation Research Part B: Methodological* **143** 65–85.

Schilde, M., K. Doerner, R. Hartl. 2014. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research* **238**(1) 18–30.

Simoni, M, K Kockelman, K Gurumurthy, J Bischoff. 2019. Congestion pricing in a world of self-driving vehicles: An analysis of different strategies in alternative future scenarios. *Transportation Research Part C: Emerging Technologies* **98** 167–185.

Skutella, M. 2009. An introduction to network flows over time. *Research trends in combinatorial optimization*. Springer, 451–482.

Soeffker, N., M. Ulmer, D. Mattfeld. 2022. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research* **298**(3) 801–820.

Sun, M., J. E. Aronson, P. G. McKeown, D. Drinka. 1998. A tabu search heuristic procedure for the fixed charge transportation problem. *European Journal of Operational Research* **106**(2-3) 441–456.

Thapalia, B. K., T. G. Crainic, M. Kaut, S. W. Wallace. 2011. Single-commodity network design with stochastic demand and multiple sources and sinks. *INFOR: Information Systems and Operational Research* **49**(3) 193–211.

Thomas, B., C. White III. 2004. Anticipatory route selection. *Transportation Science* **38**(4) 473–487.

Tian, Q., Y. H. Lin, D. Wang. 2021. Autonomous and conventional bus fleet optimization for fixed-route operations considering demand uncertainty. *Transportation* **48** 2735–2763.

Toth, P., D. Vigo. 1996. Fast local search algorithms for the handicapped persons transportation problem. *Meta-Heuristics*. Springer, 677–690.

Tsao, M., R. Iglesias, M. Pavone. 2018. Stochastic model predictive control for autonomous mobility on demand. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 3941–3948.

Ulmer, M., J. Goodson, D. Mattfeld, B. Thomas. 2020. On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics* **9**(2) 100008.

UN, DESA. 2018. Revision of world urbanization prospects. *New York: United Nations Department of Economic and Social Affairs* .

Vansteenwegen, P., L. Melis, D. Aktaş, B.D.G Montenegro, F.S. Vieira, K. Sörensen. 2022. A survey on demand-responsive public bus systems. *Transportation Research Part C: Emerging Technologies* **137** 103573.

Varaiya, P. 2005. What we've learned about highway congestion. *Access Magazine* **1**(27) 2–9.

Viergutz, K., C. Schmidt. 2019. Demand responsive vs. conventional public transportation: A MATSim study about the rural town of Colditz, Germany. *Procedia Computer Science* **151** 69–76.

Wagner, D., G. R. Raidl, U. Pferschy, P. Mutzel, P. Bachhiesl. 2007. A multi-commodity flow approach for the design of the last mile in real-world fiber optic networks. *Operations Research Proceedings 2006*. Springer, 197–202.

Walker, W. E. 1976. A heuristic adjacent extreme point algorithm for the fixed charge problem. *Management Science* **22**(5) 587–596.

Wallace, S. W. 2000. Decision making under uncertainty: Is sensitivity analysis of any use? *Operations Research* **48**(1) 20–25.

Wallar, A., M. Van Der Zee, J. Alonso-Mora, D. Rus. 2018. Vehicle rebalancing for mobility-on-demand systems with ride-sharing. *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 4539–4546.

Wang, Y., W. Shi, Z. Chen. 2021. Impact of ride-hailing usage on vehicle ownership in the united states. *Transportation research part D: Transport and Environment* **101**.

Whitley, D. 2019. Next generation genetic algorithms: a user's guide and tutorial. M. Gendreau, J.-Y. Potvin, eds., *Handbook of Metaheuristics*. Springer, 245–274.

Wilson, N., J. Sussman, H. Wong, B. Higonnet. 1971. Scheduling algorithms for dial-a-ride systems. *Urban Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA* .

Wolsey, L. 2003. Strong formulations for mixed integer programs: valid inequalities and extended formulations. *Mathematical Programming* **97** 423–447.

Yaghini, M., M. Karimi, M. Rahbar, M. H. Sharifitabar. 2015. A cutting-plane neighborhood structure for fixed-charge capacitated multicommodity network design problem. *INFORMS Journal on Computing* **27**(1) 48–58.

Yaghini, M., M. Momeni, M. Sarmadi. 2012. A simplex-based simulated annealing algorithm for node-arc capacitated multicommodity network design. *Applied Soft Computing* **12**(9) 2997–3003.

Yaghini, M., M. Rahbar, M. Karimi. 2013. A hybrid simulated annealing and column generation approach for capacitated multicommodity network design. *Journal of the Operational Research Society* **64** 1010–1020.

Yang, S. Y., J. Kim. 2015. Bitcoin market return and volatility forecasting using transaction network flow properties. *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, 1778–1785.

Zgraggen, J., M. Tsao, M. Salazar, M. Schiffer, M. Pavone. 2019. A model predictive control scheme for intermodal autonomous mobility-on-demand. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 1953–1960.

Zhang, R., M. Pavone. 2016. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research* **35**(1-3) 186–203.

Zhang, Z., B. Cheang, C. Li, A. Lim. 2019. Multi-commodity demand fulfillment via simultaneous pickup and delivery for a fast fashion retailer. *Computers & Operations Research* **103** 81–96.

Ziemke, D., I. Kaddoura, K. Nagel. 2019. The MATSim open Berlin scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data. *Procedia computer science* **151** 870–877.