TUM School of Management
Technische Universität München

Contextual and Data-Driven Decision-Making

Recent Advancements for Supply Chain and Transport Optimization

Breno Serrano de Araujo

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. Axel Parmentier

Die Dissertation wurde am 02.09.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.11.2024 angenommen

Contents

List of Abbreviations

AIC	Akaike information criterion
AMoD	autonomous mobility-on-demand
BFS	Bilevel Feature Selection
BFS-CV	BFS with cross-validation
BHO	Bilevel Hyperparameter Optimization
BIC	Bayesian information criterion
BP&C	branch-price-and-cut
CCP	chance-constrained programming
CG	column generation
CO	combinatorial optimization
CSAA	conditional sample average approximation
CS-VRPTW	conditional stochastic VRPTW
CVRP	capacitated vehicle routing problem
DARP	dial-a-ride problem
DP	dynamic programming
ERM	Empirical Risk Minimization
ESPPRC	elementary shortest path problem with resource constraints
FPF	feature projection function
K-DSPP	K-disjoint shortest path problem
KKT	Karush-Kuhn-Tucker
LP	linear program
MILP	mixed integer linear programming
MIP	mixed integer programming
MIQP	mixed integer quadratic programming
ML	machine learning
MoD	mobility-on-demand
PDP	pick-up and delivery problem
PTO	*predict-then-optimize*
RCSP	resource-constrained shortest path
RL	reinforcement learning
RMP	restricted master problem
RO	robust optimization
RSAA	residual-based sample average approximation
SAA	sample average approximation
SPR	stochastic programming with recourse
TSPTW	traveling salesman problem with time windows
VRP	vehicle routing problem
VRPTW	vehicle routing problem with time windows

Acknowledgments

First and foremost, I would like to sincerely thank my supervisor, Prof. Dr. Maximilian Schiffer, for his excellent technical guidance and for continually motivating me to grow as a researcher. I am immensely grateful for all the opportunities that he provided me throughout my PhD, which greatly contributed to my professional and personal growth, and that would not have been possible without the trust that he put in me and in my work. I would like to thank my second supervisor, Prof. Dr. Stefan Minner, for always ensuring that our work is held to the highest standards and for providing me detailed feedback on my work. I would also like to thank Prof. Dr. Thibaut Vidal for sparking my interest in research during my master's studies and inspiring me to pursue a PhD. I am thankful to Dr. Alexandre M. Florio for his invaluable contributions and for our technical discussions. I am also thankful to Prof. Dr. Alexandre Jacquillat for hosting me during my research stay at MIT and for being so enthusiastic and dedicated to our collaboration. I would also like to express my gratitude to Prof. Dr. Rainer Kolisch and Prof. Dr. Axel Parmentier for being part of my defense committee.

I am thankful to all my colleagues and friends at the Professorship of Business Analytics & Intelligent Systems (BAIS) for making the everyday routine at the office more enjoyable and fun. Special thanks go to Antonio, Georgina, Gerhard, and Paul (in alphabetical order), who were involved at certain points in technical discussions that helped me shape this dissertation. Thanks also to Benedikt and Christina for providing prompt assistance with organizational and general matters in addition to technical discussions. Thanks also to Maxi, Ramin, Shidi, and Xidong, who shared the office with me, for being very pleasant office mates. I would also like to extend my thanks to the members of the research training group AdONE (Advanced Optimization in a Networked Economy), who were also a significant part of my social network at TUM, especially during the beginning of my PhD.

On a personal level, I want to thank my family, especially my parents Beatriz and Paulo, my brother Luiz Paulo, and my sister Giulia, for their unconditional love and support, and for making the effort to visit me in Europe as much as possible. I would also like to thank my friends in Munich, including those in my friends' group *Nito*, my long-time friend Christoph, and my very-long-time friend Thiago, for making life beyond academia more relaxed and fun. I am also thankful to my friends and other family members in Brazil for remaining close despite the distance.

Finally, I want to thank my wife Juliana for her love and support even during the most difficult times. Thank you for embarking with me on this journey with such a positive energy and open mind. I really admire your companionship, courage, and perseverance, in moving to Germany with me, learning a very difficult language, pursuing your own professional relocation, and even being supportive when I was away for months in the USA. You are an inspiration to me and I love you more than you can imagine.

Abstract

The use of real-time data for operational decision-making has already transformed several industries and continues to hold great potential for the future of transportation and supply chain management. However, these opportunities require the development of novel optimization methodologies, especially to leverage contextual information, i.e., feature variables that may provide partial information regarding the uncertain problem parameters. Against this background, this thesis proposes novel models and methods focusing on three key application areas: inventory management, logistics, and urban mobility.

This thesis includes an introduction, a literature review, three methodological chapters, and a conclusion. The introduction outlines current trends and operational challenges related to transportation and supply chain management, where contextual and data-driven optimization can make significant impact. The literature review describes state-of-the-art methodologies, highlighting recent publications in the aforementioned application areas. The three methodological chapters are the main contributions of this thesis.

The first methodological chapter considers the feature-based newsvendor problem, in which the decision-maker has access to historical data containing demand observations and contextual information. The chapter proposes a novel bilevel programming formulation and a mixed integer linear programming reformulation that directly incorporates feature selection into solving the feature-based newsvendor. Computational experiments show that the proposed method recovers ground-truth features with accuracy above 96% already for instances with 400 observations. In contrast, existing regularization-based techniques often fail at feature recovery or require thousands of observations to obtain similar accuracy.

The second methodological chapter focuses on a vehicle routing problem with time windows (VRPTW) and uncertain travel times, in which the decision-maker observes contextual information before making routing decisions. The chapter introduces a novel problem variant, the contextual stochastic VRPTW, which minimizes the total transportation cost and expected late arrival penalties conditioned on the observed features. The chapter presents multiple data-driven prescriptive methods for approximating the problem's objective function and develops specialized solution algorithms based on branch-price-and-cut. Computational experiments show that, surprisingly, a feature-dependent sample average approximation outperforms existing and novel methods in most settings.

The third methodological chapter studies a mixed-service ride-hailing platform that offers users the option of requesting a ride on demand or pre-booking a ride in advance. The chapter presents a novel optimization framework that allows us to study the trade-offs between higher planning certainty due to pre-booked requests and the rise of unfavorable rides due to shifts in the travel demand distribution. The chapter provides managerial insights regarding the performance of a mixed-service platform compared to a purely on-demand system, based on the New York City yellow taxi trip data. Results show that greedily accepting all pre-booking requests leads to a 14.5% reduction in the operator's profit compared to the purely on-demand baseline. In contrast, the proposed solutions lead to profit increases of up to 7.7% while satisfying customer pick-up time windows.

The thesis concludes with a summary of the main contributions and managerial insights. The chapter then discusses limitations and opportunities for future research building upon the proposed methodologies.

1 Introduction

1.1 Background

Over the past decades, many industries across the globe have transformed their operations by automating and digitalizing their processes to reduce costs and improve efficiency and productivity—a transition commonly referred to as digital transformation. Simultaneously, advancements in sensor technologies, the adoption of smartphones with precise global positioning systems, and accessible mobile Internet have enabled the collection of unprecedented amounts of data. Against this background, new business models have emerged that are transforming customer behavior and the dynamics of supply chain and transportation systems. For instance, the retail and logistics industries have experienced profound changes with the rise of e-commerce and same-day parcel delivery, while ride-hailing platforms have reshaped passenger transportation in urban areas (Cramer and Krueger, 2016).

In today's rapidly changing environments, companies face the task of efficiently optimizing large-scale operations involving several complex decisions. Moreover, uncertainty plays a critical role due to the dynamic nature of customer behavior and uncertain external factors affecting the supply chain and transportation systems. In this context, the growing availability of data creates opportunities for companies to make more informed decisions, but also poses challenges that require novel methodologies. Accordingly, data-driven decision-making has attracted the attention of both academia and industry. In particular, the research fields of contextual and data-driven optimization have focused on solving stochastic optimization problems where the decision-maker has access to historical data on the uncertain parameters and possibly to related contextual information, represented as feature variables. This thesis makes methodological contributions to these research fields, focusing on three application areas where contextual and data-driven decision-making can have significant impact: inventory management, logistics, and urban mobility.

First, a fundamental problem faced in the retail industry is matching supply with demand for perishable products with short life cycles. In this context, the newsvendor problem and its variants have served as fundamental building blocks for modeling inventory and supply chain management problems. In particular, the feature-based newsvendor problem (cf. Ban and Rudin, 2019; Beutel and Minner, 2012) considers a decision-maker

that optimizes the inventory of a product with uncertain demand, having access to related contextual information before making ordering decisions. However, in many practical settings, companies have large amounts of feature data, which may compromise explainability or cause the model to overfit to the training data. Although feature selection methods are known to address these challenges in the predictive machine learning setting (cf. Kuhn and Johnson, 2019), an open challenge remains regarding how to effectively integrate the tasks of feature selection, demand estimation, and inventory optimization within the newsvendor setting.

Second, day-ahead route planning is a complex challenge that is prevalent in the logistics industry. Vehicle routing is not only a computationally intractable problem in theory but is in practice also heavily affected by several uncertain factors, some of which are not directly observable by the decision-maker. As optimization theory and technologies have become more mature for deterministic vehicle routing problems (cf. Laporte, 2009), the research community has shifted its attention to the challenge of handling uncertainty (cf. Gendreau, Jabali, and Rei, 2014). However, despite the extensive literature in this field, current models cannot fully capture the complexities and nuances that influence the dynamics of real-world vehicle routing. Accordingly, this thesis aims to contribute to the theory and practice of vehicle routing by introducing novel optimization methods that leverage contextual information, e.g., regarding road closures, seasonal events, or the day of the week, to improve routing decisions.

Third, ride-hailing platforms are constantly exploring new concepts for urban mobility with the aim of improving customer experience. One recent innovation deployed by major platforms is a pre-booking service, which allows time-sensitive customers to reserve rides in advance (Bolt 2023; Lyft 2023; Uber 2023). Accordingly, customers have the option to either pre-book a ride in advance or request a ride on demand, giving rise to mixed-service ride-hailing systems. From the operator's perspective, pre-booked rides correspond to higher planning certainty, as the travel demand is known in advance. However, the operator must decide whether to accept pre-booking customers without knowledge of future on-demand requests, which may later force the operator to reject potentially more profitable on-demand customers if the driver availability is insufficient to meet overall demand. In addition, mixed-service systems may face induced pre-booking demand in areas traditionally experiencing low driver supply, e.g., low-demand suburban neighborhoods, resulting in a shift in the travel demand distribution. These challenges motivate the need for novel optimization frameworks to analyze and control mixed-service ride-hailing systems.

In conclusion, current supply chain and transportation systems face several methodological challenges regarding the effective use of data and contextual information for

making more informed decisions. Moreover, different industry sectors present specific nuances that require tailored methodologies. Against this background, this thesis provides a broad set of contributions to the fields of contextual and data-driven decision-making for the aforementioned application areas.

1.2 Aims and Scope

This thesis proposes novel models and methods combining concepts from machine learning and operations research for data-driven and contextual optimization, focusing on three different application domains, particularly inventory management, logistics, and urban mobility. Specifically, this thesis addresses the following challenges and opportunities:

Feature selection and contextual optimization: Companies nowadays have large data sets for training machine learning models, which are embedded into contextual optimization frameworks. However, it is often necessary to reduce the number of available feature variables to avoid overfitting to the training data and to improve explainability, which is crucial, e.g., when dealing with high-stakes decisions. This thesis investigates the challenge of integrating feature selection within a contextual optimization framework, focusing on a data-driven newsvendor problem setting.

Route planning with contextual information: In day-ahead route planning, incorporating contextual information—such as road closures, seasonal events, or the day of the week—creates opportunities for logistics service providers to plan more cost-effective routes. This thesis investigates novel optimization models and methods for integrating contextual information into day-ahead route planning, specifically focusing on a vehicle routing problem with time windows (VRPTW) with uncertain travel times.

Mixed-service ride-hailing platforms: Pre-booking services offer potential advantages for time-sensitive customers by providing service guarantees regarding driver availability and on-time pick-up. This thesis proposes a novel data-driven optimization framework for the analysis of mixed-service ride-hailing systems, allowing us to evaluate their profitability compared to purely on-demand systems.

1.3 Structure and Contribution

The remainder of this thesis is organized as follows:

Chapter 2 reviews the literature on contextual and data-driven optimization from a methodological perspective. Connecting to the application areas investigated in this thesis, the chapter reviews recent literature on the data-driven newsvendor problem, stochastic vehicle routing problems, and mobility-on-demand systems.

Chapter 3 addresses the feature-based newsvendor problem, in which a decision-maker has access to historical data consisting of demand observations and related exogenous features. This chapter investigates the task of feature selection, aiming to derive sparse, explainable models with improved out-of-sample performance. The chapter introduces a novel bilevel programming formulation for learning a decision function that prescribes ordering decisions directly from feature observations, utilizing only relevant features among a possibly large set of available features. Specifically, the upper-level problem selects a subset of features that minimizes an estimate of the out-of-sample cost based on a held-out validation set. The lower-level problem learns the optimal coefficients of the decision function on a training set, using only the features selected by the upper-level. The bilevel program is reformulated as a mixed integer linear program and solved to optimality with standard optimization solvers. The computational experiments showed that the proposed method recovers ground-truth features with accuracy above 96% already for instances with 400 observations. In contrast, existing regularization-based methods often fail at feature recovery and even the best-performing methods cannot consistently achieve accuracy values above 90%. Regarding out-of-sample generalization, the proposed method achieved improved or comparable cost performance.

Chapter 4 deals with the VRPTW under uncertain travel times, assuming that feature data is revealed to the decision-maker before routing decisions are made. For example, the decision-maker may use information about road closures, seasonal events, or simply knowledge of the day of the week, among other travel time predictors, to make better routing decisions. Accordingly, this chapter introduces the contextual stochastic VRPTW, which minimizes the total transportation costs and expected late arrival penalties conditional on observed features. Since the contextual stochastic VRPTW formulation relies on the unknown joint distribution of travel times and features, the chapter presents data-driven prescriptive models that leverage historical data to approximate the conditional expectation in the objective function. The prescriptive models are distinguished between three different paradigms: point-based approximation, sample average approximation (SAA), and penalty-based approximation, each taking a different perspective on dealing with uncertainty. A specialized branch-price-and-cut (BP&C) algorithm is developed for solving the proposed models. Computational experiments compare the out-of-sample cost per-

formance of different data-driven prescriptive models based on instances with up to one hundred customers. The proposed conditional SAA method provides up to a 13.2% reduction in test cost compared to the classical SAA baseline, and up to a 26.1% reduction compared to a point-based approximation method following the traditional predict-then-optimize approach.

Chapter 5 considers a mixed-service ride-hailing system that offers customers the option to request a ride on demand or to pre-book it in advance. This chapter introduces a novel two-stage stochastic optimization formulation in which the first-stage problem consists of deciding which pre-booking requests to accept, while the second-stage problem involves assigning vehicles to requests and planning routes with uncertain on-demand requests. The chapter then presents a SAA formulation and develops a scalable solution algorithm that solves approximations of the second-stage subproblems using a polynomial-time algorithm. Computational experiments based on the New York City yellow taxi data show that a greedy policy that accepts all pre-booking requests can lead to a profit decrease of up to 14.5%. In contrast, the proposed SAA solutions lead to profit increases ranging from 6.5% to 7.7%, in settings with weak and strong distributional shifts, respectively, while satisfying customer pick-up time windows.

Chapter 6 offers a comprehensive summary of the key managerial and methodological contributions, before discussing current limitations and highlighting future research perspectives.

2 State of the art

This chapter provides an overview on works related to contextual and data-driven optimization, focusing on the application areas studied in the thesis. Section 2.1 provides a brief overview of recent developments in the field of contextual and data-driven optimization. Section 2.2 discusses the literature on data-driven and feature-based newsvendor problems. Section 2.3 reviews works on stochastic and dynamic vehicle routing problems. Section 2.4 focuses on methodologies for the control and optimization of mobility-on-demand systems.

2.1 Contextual and Data-driven Optimization

The emerging field of contextual optimization addresses optimization problems in which the objective function and the feasible region depend on parameters that are uncertain and for which related contextual information is available to the decision-maker. Mišić and Perakis (2020) reviewed recent applications of contextual and data-driven optimization to operations management problems, including supply chain management, revenue management, and healthcare operations. From a methodological perspective, Sadana et al. (2024) surveyed the recent literature and categorized the proposed methods into three different paradigms, which we summarize in the following. Other loosely related research streams, not covered in our literature review, have investigated different ways of combining machine learning (ML) and combinatorial optimization (CO), e.g., using ML to improve solution algorithms for solving CO problems, or directly learning solutions to CO problems, among other topics. We refer to Bengio, Lodi, and Prouvost (2021) and Kotary et al. (2021) for a general overview of those research areas.

Decision rule optimization consists in employing a parameterized decision function that maps feature observations to decisions, and learning the parameter values by optimizing the empirical performance based on the available data. In the context of inventory optimization, Ban and Rudin (2019) and Beutel and Minner (2012) studied the data-driven newsvendor problem and proposed models for learning a decision function that prescribes order quantities directly from feature observations. Further works by Oroojlooyjadid, Snyder, and Takáč (2020) and Zhang and Gao (2017) employed non-linear

decision functions based on neural networks in a similar problem setting. Shlezinger, Eldar, and Boyd (2022) provide a general introduction to decision rule optimization based on deep neural networks.

Sequential learning and optimization is a two-stage procedure that first estimates the conditional distribution of the uncertain parameters given the features, and then solves the corresponding conditional stochastic optimization problem. Bertsimas and Kallus (2019) proposed a framework for contextual optimization based on weighted sample average approximation (SAA) and derived weight functions based on ML methods, such as k-nearest neighbors regression, local linear regression, and tree-based methods. The authors applied the framework to a two-stage shipment planning problem and to a multi-product inventory problem. Bertsimas and McCord (2019) and Bertsimas, McCord, and Sturt (2023) extended the framework of Bertsimas and Kallus (2019) to multi-stage and multi-period stochastic optimization problems, respectively. Ban, Gallien, and Mersereau (2019) and Rios, Wets, and Woodruff (2015) proposed a scenario generation method based on estimating the distribution of forecast errors, i.e., residuals. Ban, Gallien, and Mersereau (2019) focused on a multi-stage stochastic procurement problem under uncertain demand, where they learned a regression model that relates features to demands and then used the residuals to generate demand samples for a new product. At a subsequent stage, they solved an SAA model to determine the optimal procurement policy.

Integrated learning and optimization takes the structure of the downstream optimization problem into account while training the predictive component, e.g., by considering a modified loss function instead of minimizing a standard regression loss. Donti, Amos, and Kolter (2017) proposed a gradient-based method for training a neural network model that minimizes a loss function with respect to the downstream optimization problem. The authors focused on quadratic optimization problems, and evaluated their approach on applications to energy systems and inventory management. Focusing on problems with a linear objective, Elmachtoub and Grigas (2021) introduced the smart "predict, then optimize" framework, which trains the prediction model based on a modified regret loss function. Structured learning is an alternative approach for solving contextual optimization problems that consists of embedding a differentiable CO layer within an ML prediction model (see, e.g., Dalle et al. 2022; Nowozin and Lampert 2011; Parmentier 2021). Using structured learning, Jungel et al. (2023) proposed a framework for controlling a fleet of autonomous mobility-on-demand (AMoD) vehicles and Baty et al. (2024) solved a dynamic vehicle routing problem with time windows (VRPTW). Parmentier and T'kindt (2023) used a structured learning framework for solving a single-machine scheduling problem with release dates.

This thesis contributes to the literature in the following ways. Chapter 3 proposes a novel method based on bilevel optimization that directly integrates feature selection and decision rule optimization for the data-driven feature-based newsvendor introduced by Ban and Rudin (2019) and Beutel and Minner (2012). Chapter 4 shows how existing sequential learning and optimization approaches can be adapted to the stochastic VRPTW and proposes novel methods for this problem setting. Chapter 5 proposes a data-driven optimization framework for analyzing a mixed-service ride-hailing system, particularly focusing on the impact of distributional shifts on the system performance.

2.2 Data-driven Newsvendor Problem

The newsvendor problem, along with its many variants, has served as a foundational model for inventory and supply chain management. In its traditional form, the problem involves a risk-neutral decision-maker who sets the order quantity of a perishable product before observing its uncertain demand. At the end of the day, after demand realizes, the newsvendor incurs a shortage cost per unit of unmet demand and a holding cost per unit of unsold products discounted by their unit salvage value. Early works assumed the demand distribution to be known, allowing the optimal order quantity to be derived as a specific quantile of the demand distribution. There is extensive literature on extensions of the newsvendor problem, e.g., with different objectives or utility functions (Chen et al., 2007; Wang and Webster, 2009), pricing policies (Petruzzi and Dada, 1999), and multi-product or multi-period settings (Kogan and Lou, 2003; Lau and Lau, 1996). We focus only on closely related works in the following and refer to Khouja (1999), Qin et al. (2011), and Choi (2012) for general surveys on newsvendor models, extensions, and applications.

In practice, the decision-maker does not have access to the true underlying demand distribution, relying instead on a finite set of demand observations, which has led to research on the distribution-free newsvendor problem. In this context, the seminal work of Scarf (1958) derived the optimal order quantity that maximizes profit against the worst-case demand distribution, assuming that only the mean and variance of demand are known. For a review on the distribution-free newsvendor and extensions thereof, we refer to Gallego and Moon (1993), Moon and Gallego (1994), and Yue, Chen, and Wang (2006).

More recent works on the data-driven newsvendor proposed model formulations that directly leverage a sample of available data instead of focusing on estimating distributional parameters. In this context, SAA is a common solution approach (see, e.g., Kleywegt, Shapiro, and Mello, 2002; Shapiro, 2003). Levi, Roundy, and Shmoys (2007) applied SAA for the single-period featureless newsvendor problem and established upper bounds

on the number of samples required to achieve a specified relative error. In this course, Besbes and Mouchtaki (2023), Cheung and Simchi-Levi (2019), and Levi, Perakis, and Uichanco (2015) further improved upon previous SAA bounds. Bertsimas and Thiele (2005) proposed a data-driven model based on robust optimization that can be reformulated as a linear program (LP) and trades off higher profits for a decrease in the downside risk. Further robust optimization approaches were investigated by Bertsimas and Thiele (2006) and See and Sim (2010) for a multi-period inventory problem. Finally, several authors employed data-driven distributionally robust approaches in the context of multi-item newsvendor problems (see, e.g., Ben-Tal et al., 2013; Bertsimas, Gupta, and Kallus, 2018; Hanasusanto et al., 2015; Wang, Glynn, and Ye, 2016).

Beutel and Minner (2012) and Ban and Rudin (2019) introduced feature-based newsvendor models that integrate demand estimation and order quantity optimization. They proposed to learn a linear decision function that directly predicts ordering decisions based on input features, in contrast to the existing "estimate-then-optimize" paradigm, which first estimates the demand distribution and then optimizes the inventory level. Further works by Oroojlooyjadid, Snyder, and Takáč (2020) and Zhang and Gao (2017) applied neural networks to the newsvendor problem, incorporating tailored loss functions that take into account the inventory shortage and holding costs. A related study by Huber et al. (2019) empirically compared various data-driven approaches to the feature-based newsvendor, evaluating their performance against model-based approaches that rely on distributional assumptions. Their experiments on real-world data showed the superior performance of data-driven approaches in most cases.

Regarding feature selection, Ban and Rudin (2019) penalized the complexity of the solution by including a regularization term in the objective function, thereby favoring the selection of fewer features. However, specifying the regularization parameter value remains an open challenge, for which heuristics are often employed. To address this challenge, Chapter 3 formalizes the combined problem of feature selection and decision rule optimization as a bilevel optimization problem and provides a tractable single-level reformulation, which avoids regularization altogether.

2.3 Vehicle Routing

Vehicle routing is a fundamental problem arising in the transportation of goods across supply chain networks. Due to its practical importance, researchers have studied the VRP and its several variants for more than six decades (Laporte, 2009). The VRP variants that received the most attention in research are the capacitated vehicle routing problem (CVRP), in which vehicles have limited capacity; the vehicle routing problem with time

windows (VRPTW), in which vehicles can only serve customers during the respective customer time windows; and the pick-up and delivery problem (PDP), in which vehicles transport goods from a pick-up location to a drop-off location. For general surveys on classic and emerging VRP variants, we refer to Golden, Raghavan, and Wasil (2008), Toth and Vigo (2014), and Vidal, Laporte, and Matl (2020).

Solution methods for VRPs fall into the category of exact algorithms or (meta)heuristics. The most efficient exact methods typically adopt a set-partitioning formulation and rely on branch-and-price (Barnhart et al., 1998), which combines branch-and-bound with column generation (CG). The additional use of cutting planes (Desaulniers, Desrosiers, and Spoorendonk, 2011) leads to so-called branch-price-and-cut (BP&C) algorithms. Feillet (2010) provides a tutorial on the fundamental concepts behind BP&C for VRPs, while Costa, Contardo, and Desaulniers (2019) review recent advancements. Pessoa et al. (2020) developed a generic VRP solver based on BP&C that includes most state-of-the-art components introduced for the most classical problem variants. Regarding (meta)heuristics, a wide variety of tailored algorithms have been proposed to address the several VRP variants from the literature. We refer to Cordeau et al. (2002) and Vidal et al. (2013) for comprehensive surveys. A more detailed discussion on modern solution algorithms for VRPs remains out of scope of this thesis and we focus instead on the main modeling assumptions of problem variants that are closely related to our work.

A VRP variant that is related to this thesis is the time-dependent VRP, which considers travel times that vary as a continuous (deterministic) function of time. Consequently, the time of the day implicitly provides contextual information that relates to the travel times. These travel time functions may be obtained, e.g., by directly taking into account historical patterns that capture traffic congestion throughout the day. In this setting, Dabia et al. (2013) proposed a branch-and-price algorithm and Gmira et al. (2021) developed a heuristic based on tabu search for the time-dependent VRPTW. For an introduction to time-dependent routing problems, we refer the reader to the review paper by Gendreau, Ghiani, and Guerriero (2015).

In practice, uncertainty plays a key role in routing problems, e.g., due to uncertain customers, demands, travel times, or service times. In response to this practical challenge, several problem variants have emerged aiming to address different aspects related to uncertainty. Pillac et al. (2013) proposed a classification of VRPs into two dimensions regarding information *quality* (i.e., stochasticity) and information *evolution* (i.e., dynamism). Accordingly, a problem may be *deterministic* or *stochastic* regarding information quality, and may be *static* or *dynamic* regarding information evolution. In the following, we review recent works on stochastic VRPs (Section 2.3.1) and dynamic VRPs (Section 2.3.2).

2.3.1 Stochastic Vehicle Routing

Stochastic VRPs traditionally refer to day-ahead (static) planning problems having uncertain or probabilistic parameters, which can be represented as random variables. Routes are designed before their execution and uncertainty realizes during their execution. After uncertainty realization, there is limited flexibility for reoptimization, and typically minor adjustments are allowed. The most common variants of this problem type focus on stochastic demands, travel times, and customers. For example, demand may be static, i.e., time-invariant, but uncertain, such that demand is unknown to the decision-maker during route planning and only realizes when a vehicle visits the respective customer. In the case of stochastic customers, the set of possible customers is known a priori but each customer has a probability of actually requiring service. For an introduction to stochastic VRPs, we refer to the surveys of Oyola, Arntzen, and Woodruff (2017, 2018) and to the respective book chapter in Gendreau, Jabali, and Rei (2014). The three most common modeling approaches from the literature are based on stochastic programming with recourse (SPR), chance-constrained programming (CCP), and robust optimization (RO). Given the extensive body of literature, we provide only a concise overview of recent publications on stochastic VRPs for day-ahead planning.

Stochastic programming with recourse: The decision-maker defines a recourse policy describing what actions to take in case of a route failure. The objective typically minimizes the deterministic transportation cost associated with the planned routes plus the expected cost associated with the recourse policy. In the case of uncertain demand, a simple recourse policy determines that the vehicle returns to the depot to restock when its capacity is exceeded (Rei, Gendreau, and Soriano, 2010). After restocking, the vehicle resumes service at the customer where the route failure occurred and then continues the planned route. Other common recourse policies consist in returning to the depot for preventive restocking before a failure occurs (Marinakis, Iordanidou, and Marinaki, 2013), or re-optimizing the vehicle route after each customer visit (Secomandi and Margot, 2009). Florio, Hartl, and Minner (2020) developed an exact algorithm for the VRP with stochastic demands assuming optimal restocking. Exact BP&C algorithms were also developed by Gauvin, Desaulniers, and Gendreau (2014), among others. Stochastic VRPTW formulations often assume soft time windows, such that early or late service are penalized (see, e.g., Taş et al. 2014a; Taş et al. 2014b). In particular, Taş et al. (2014b) studied a stochastic version of the time-dependent VRPTW with soft time windows, in which travel times are represented by random variables whose probability distribution functions vary with time.

Chance-constrained programming: A chance-constrained model formulation con-

tains probabilistic constraints which bound the level of risk experienced by the decision-maker. These constraints ensure that the probability of route failure is below a certain threshold, while typically ignoring the cost of failures (Tan, Cheong, and Goh, 2007). In particular, Mendoza, Rousseau, and Villegas (2016) proposed a model formulation for the VRP with stochastic demand that enforces that the probability of the total route duration exceeding a maximum duration must be lower than a given threshold. Miranda and Conceição (2016) and Errico et al. (2018) adopted a CCP approach for the VRPTW with stochastic travel and service times, respectively. Further model formulations based on CCP were investigated by Beraldi et al. (2015) and Zhang, Lam, and Chen (2013, 2016), among others. Dinh, Fukasawa, and Luedtke (2018) investigated exact algorithms for solving chance-constrained VRPs.

Robust optimization: A robust solution corresponds to routes that are feasible for all realizations of the uncertain parameters within a predetermined uncertainty set. Lee, Lee, and Park (2012) studied the stochastic VRP with deadlines, a special case of the VRPTW, and considered stochastic demands and travel times. Adulyasak and Jaillet (2016) proposed a distribution-free RO approach that minimized the risk of violating the deadlines, i.e., the lateness probability. Zhang et al. (2021) addressed the VRPTW with stochastic travel times and proposed a data-driven distributionally robust optimization model. Gounaris, Wiesemann, and Floudas (2013) proposed an RO formulation for the CVRP with stochastic demand. Further works by Ghosal, Ho, and Wiesemann (2024) and Ghosal and Wiesemann (2020) proposed distributionally robust chance-constrained optimization models.

Chapter 4 contributes to the literature by introducing a novel problem formulation that incorporates contextual information into the well-studied VRPTW variant. In the novel contextual stochastic VRPTW, travel times are uncertain and the decision-maker observes related contextual information before planning the routes. Given the complexity of the stochastic VRPTW, several tailored solution techniques have been developed over the last decades. Chapter 4 introduces customized BP&C algorithms for solving the contextual stochastic VRPTW, describing how state-of-the-art solution algorithms designed for the VRPTW can be adapted to incorporate contextual information.

2.3.2 Dynamic Vehicle Routing

In dynamic VRPs, requests enter the system throughout the operating horizon and the decision-maker must redesign vehicle routes in an ongoing fashion in response to revealed information. For example, food delivery or ride-hailing platforms receive customer requests throughout the day and, in response, must dynamically plan vehicle routes to

serve those requests. In contrast to stochastic VRPs, all request-specific parameters become known to the decision-maker once the request enters the system, i.e., even before the customer is visited by a vehicle. We note that, in practice, VRPs may be both dynamic and stochastic, where information becomes available during the operating horizon and the input parameters are represented as random variables, whose probability distribution function may also change over time. For surveys on dynamic VRPs, we refer the reader to Bektaş, Repoussis, and Tarantilis (2014), Pillac et al. (2013), Psaraftis, Wen, and Kontovas (2016), Rios et al. (2021), Ritzinger, Puchinger, and Hartl (2016), and Soeffker, Ulmer, and Mattfeld (2022).

The earliest approaches for solving dynamic VRPs often adapted solution methods developed for static versions of the problem (Gendreau et al., 1999). In particular, myopic approaches repeatedly solve a static version of the optimization problem in a rolling horizon fashion, as soon as new information is revealed. Bent and Van Hentenryck (2004) proposed a look-ahead scenario-based approach, which samples possible realizations of future requests, individually solves the VRP corresponding to each scenario, and then selects a consensus solution. Further works in the literature investigated approaches based on approximate dynamic programming (see, e.g., Novoa and Storer 2009; Ulmer, Soeffker, and Mattfeld 2018; Ulmer 2017) and reinforcement learning (RL; see, e.g., Joe and Lau 2020; Nazari et al. 2018; Zhao et al. 2021). In particular, the survey papers by Hildebrandt, Thomas, and Ulmer (2023) and Raza, Sajid, and Singh (2022) discuss recent advancements and opportunities for solving dynamic VRPs using RL. Recently, Baty et al. (2024) proposed a novel ML pipeline that incorporates a CO layer for solving a dynamic VRPTW.

Lastly, related works on partially dynamic VRPs consider that part of the requests are known to the decision-maker before the start of operations, while the remaining requests are revealed in an online fashion. Lund, Madsen, and Rygaard (1996) introduced the concept of degree of dynamism, defined as the number of dynamic requests divided by the total number of requests that enter the system, i.e., before and during the operating horizon. They proposed heuristic methods adapted from the insertion heuristic by Solomon (1987). Larsen, Madsen, and Solomon (2002) proposed the effective degree of dynamism, which extends the degree of dynamism by considering the requests' pick-up times.

One particular real-world application that involves dynamic vehicle routing is the operation of mobility-on-demand systems. Chapter 5 specifically studies a mixed-service ride-hailing platform that serves both on-demand and pre-booked customers. Since pre-booked requests are known to the system operator before the start of the operating horizon, this system can be viewed as a partially dynamic VRP. The following section reviews concepts related to mobility-on-demand systems, discusses methodologies for their control

and optimization, and outlines our contributions to this area of research.

2.4 Mobility-on-Demand Systems

Mobility-on-demand (MoD) typically refers to systems in which a fleet of vehicles provides flexible and individualized point-to-point passenger transportation. Taxi services were among the earliest MoD systems, offering fast transportation within cities. Over the past decades, cities worldwide have experienced the rise of ride-hailing platforms (Cramer and Krueger, 2016) in which customers request rides via a mobile app, and a central operator dispatches incoming ride requests to nearby available drivers. More recently, the emergence of autonomous vehicles has inspired new business models for passenger transportation, such as autonomous mobility-on-demand (AMoD) systems. Lastly and particularly related to this thesis, mixed-service ride-hailing systems offer time-sensitive customers the option of pre-booking a ride in advance. In the following, we review the literature on MoD systems that are most related to this thesis.

From an optimization and control perspective, MoD systems face several operational challenges, including travel demand estimation and forecasting (Moreira-Matias et al., 2013a,b; Osorio, 2019), pricing (Banerjee, Freund, and Lykouris, 2021; Garg and Nazerzadeh, 2022; Ma, Fang, and Parkes, 2021), driver compensation and incentive schemes (Bai et al., 2019; Hu and Zhou, 2020), trip-vehicle assignment (Xu et al., 2018), pooling, routing, and repositioning (Braverman et al., 2019). In addition, MoD platforms have multiple objectives that may conflict with each other (Lyu et al., 2019), including short-term goals, such as maximizing immediate earnings or minimizing customer wait times, and long-term goals related to passenger and driver satisfaction, platform profit and reputation. We refer the reader to Wang and Yang (2019) for a review of methodologies for the operation of different MoD systems.

Ride-hailing systems can be modeled as dial-a-ride problems (DARPs), a variant of PDPs, in which vehicles transport customers instead of products, and customers may be associated with pick-up time windows. For a review on dynamic DARPs, we refer to the survey paper by Cordeau and Laporte (2007). In particular, Berbeglia, Cordeau, and Laporte (2012) introduced a hybrid tabu search algorithm for the dynamic DARP. A large body of literature focused on trip-vehicle assignment—often denoted as order or vehicle dispatching—and alternatively modeled the problem as a dynamic bipartite matching problem (see, e.g., Dickerson et al. 2021; Özkan and Ward 2020; Xu et al. 2018). For a survey on dynamic pricing and matching for ride-hailing, we refer to Yan et al. (2020).

The earliest approaches for dynamic matching involved batching incoming ride requests during a given time interval and then determining trip-vehicle assignments by solving a

(static) maximum weighted bipartite matching, where the weights correspond, e.g., to the profits associated with each trip-vehicle pair (Ashlagi et al., 2022). Bertsimas, Jaillet, and Martin (2019) proposed an online re-optimization method that iteratively assigns incoming ride requests to vehicles by solving an offline optimization model in a rolling-horizon fashion. Additional constraints may be considered to prevent drivers from being assigned to customers at distant pick-up locations (Feng, Kong, and Wang, 2020). Further works incorporated predictive components in order to anticipate future demand (see, e.g., Jungel et al., 2023).

Algorithms for online bipartite matching are often more scalable than algorithms for VRPs, which favored their deployment in real-world systems. In particular, Zhang et al. (2017) provided a detailed description of the matching algorithm based on request batching that was deployed at DiDi, serving tens of millions of users daily. Zhou et al. (2019) introduced a multi-agent RL method that extends a deep Q-learning network with Kullback-Leibler divergence optimization. Tang et al. (2019) proposed a novel RL method for vehicle dispatching and conducted offline simulations using data from DiDi. Sadeghi Eshkevari et al. (2022) and Xu et al. (2018) described RL methods deployed by DiDi that dynamically solves a maximum weighted bipartite matching problem using a state value function. Jiao et al. (2021) studied the problem of rebalancing idle vehicles using an RL framework.

Ride-sharing, or ride pooling, refers to a mobility concept that allows multiple passengers to share a vehicle when traveling along similar routes. Ride-sharing systems can potentially bring significant societal and environmental benefits by improving vehicle utilization and reducing the amount of privately-owned cars, but introduce challenges that require novel optimization methodologies. For a review on dynamic ride-sharing, we refer to Agatz et al. (2012), Furuhata et al. (2013), and Lokhandwala and Cai (2018). In particular, Alonso-Mora et al. (2017) proposed an algorithm based on model-predictive control for optimizing a ride-sharing system. Liu and Samaranayake (2020) extended the framework of Alonso-Mora et al. (2017) focusing on techniques for proactive rebalancing. Ashlagi et al. (2022) proposed a batching algorithm for online matching in a ride-sharing application where at most 2 passengers can share a ride. Pavone et al. (2022) proposed a polynomial-time randomized batching algorithm and generalized the work of Ashlagi et al. (2022) to the setting of high capacity ride-sharing, i.e., where potentially more than 2 passengers share a ride. Soza-Parra, Kucharski, and Cats (2024) investigated how different travel demand patterns affect the shareability of a ride-pooling system, a metric that measures the extent to which different customers can share rides, taking into account the rides' compatibility in time and space, and customer preferences. Qin, Zhu, and Ye (2021, 2022) surveyed papers that employ RL for the control of ride-sharing systems.

Autonomous mobility-on-demand is an emerging concept wherein a centrally controlled fleet of robotic, self-driving vehicles transports passengers on demand. AMoD systems hold the potential of transforming personal urban mobility as they relieve passengers from the task of driving and allow for the rebalancing of idle vehicles to areas with potentially more customers. For a general overview of methods for the analysis and control of AMoD systems, we refer to Pavone (2015) and Zardini et al. (2022). Spieser et al. (2014) performed a case study in the city of Singapore, in which they provided guidelines regarding fleet sizing and assessed the financial feasibility of implementing an AMoD system. Common approaches for the control of AMoD systems include methods based on network flow optimization (Rossi et al., 2018), queuing-theoretical models (Zhang and Pavone, 2016), stochastic model predictive control (Tsao, Iglesias, and Pavone, 2018), structured learning (Jungel et al., 2023), and reinforcement learning (Qin, Zhu, and Ye, 2022). In particular, Enders et al. (2023) and Hoppe et al. (2024) proposed a hybrid RL algorithm that uses multi-agent soft actor-critic to parameterize a weighted bipartite matching problem. Other works focused explicitly on the rebalancing problem (see, e.g., Gammelli et al., 2021; Iglesias et al., 2018; Liang et al., 2022; Pavone et al., 2012; Skordilis et al., 2022).

Mixed-service ride-hailing systems—also denoted as reservation-based systems—offer customers the option to either pre-book a ride in advance or request a ride on demand. Although some of the most popular ride-hailing platforms recently started to offer pre-booking services (Bolt 2023; Lyft 2023; Uber 2023), research on this type of system is still scarce. In particular, Engelhardt, Dandl, and Bogenberger (2022) studied a mixed-service ride-pooling system, and proposed a solution algorithm that first provides an offline solution for pre-booked requests and then accommodates on-demand customers in an online fashion by using a framework based on Alonso-Mora et al. (2017). Engelhardt, Dandl, and Bogenberger (2022) showed that on-demand customers also benefit from decreased waiting and detour times when pre-booking is enabled. Bilali et al. (2019) proposed an analytical model to study the influence of ride requests' reservation times in a ride-sharing system, considering short reservation times between 2 to 15 minutes. In contrast, this thesis focuses on systems with longer reservation times, where pre-booking requests enter the system at least one day before the operating horizon.

According to the review paper by Narayanan, Chaniotakis, and Antoniou (2020), pre-booking has been studied in the context of shared AMoD systems by Lamotte, De Palma, and Geroliminis (2017), Levin (2017), Ma et al. (2017), and Pimenta et al. (2017). However, most works assumed that only pre-booking is possible, instead of considering a mix of pre-booking and on-demand requests. Duan et al. (2020) considered a mixed-service AMoD system and proposed a framework consisting of a centralized dispatcher that as-

signs short-term requests to vehicles, while each vehicle manages its own long-term route, using a heuristic method to respond to incoming long-term requests. Abkarian, Mahmassani, and Hyland (2022) studied a mixed-service system that combined on-demand passenger transportation and carsharing using autonomous vehicles, such that customers can either rent a vehicle for a certain time slot or pre-book a ride specifying their origin and destination locations. The authors developed a dynamic simulation framework and proposed re-optimization methods for assigning vehicles to requests. In contrast to our work, they assumed that the operator cannot reject ride requests and that customers are willing to wait indefinitely for their ride. Lastly, Elting and Ehmke (2021) considered the option of pre-booking a ride in a dynamic ride-sharing system, modeled as a variant of a DARP. The authors studied how the degree of dynamism (cf. Lund, Madsen, and Rygaard 1996) affects the system performance, e.g., in terms of rejection and occupancy rates.

Literature on mixed-service platforms is limited and often assumes that on-demand and pre-booking requests follow the same spatio-temporal distribution, although pre-booking requests may arrive before the operating horizon. Chapter 5 studies a mixed-service ride-hailing system under distributional shift, i.e., considering that the travel demand distribution of pre-booking and on-demand customers may diverge. The chapter analyzes the system performance under varying levels of distributional shift and provides managerial insights for ride-hailing operators.

2.5 Conclusion

This chapter presented a concise review of recent literature related to the research areas explored in this thesis. These research fields have been subject to vivid activity in recent years, emphasizing their relevance in academia and practice. This thesis contributes to the existing literature by closing the research gaps outlined in the preceding sections, which are summarized in the following.

Feature selection in the data-driven newsvendor: Recent literature on the feature-based newsvendor problem has proposed methods that integrate demand estimation and inventory optimization. In practical settings, it is often necessary to reduce the number of available feature variables in order to avoid overfitting to the training data and to improve explainability. This thesis addresses this challenge by proposing an integrated approach to feature selection based on bilevel optimization.

Contextual optimization for vehicle routing: Despite extensive literature on day-ahead planning for stochastic vehicle routing, integrating contextual information with

data-driven optimization has received limited attention in this setting. This thesis proposes novel models and methods that enable logistics service providers to harness historical data on travel times and related contextual information to plan context-dependent delivery routes.

Data-driven optimization for mixed-service ride-hailing: There is limited academic literature on mixed-service ride-hailing systems, particularly regarding shifts in the travel demand distribution. This thesis proposes a novel data-driven optimization framework and provides managerial insights to support ride-hailing platforms in the operation of such mixed-service systems.

3 Bilevel Optimization for Feature Selection in the Data-Driven Newsvendor Problem

Abstract

We study the feature-based newsvendor problem, in which a decision-maker has access to historical data consisting of demand observations and exogenous features. In this setting, we investigate feature selection, aiming to derive sparse, explainable models with improved out-of-sample performance. Up to now, state-of-the-art methods utilize regularization, which penalizes the number of selected features or the norm of the solution vector. As an alternative, we introduce a novel bilevel programming formulation. The upper-level problem selects a subset of features that minimizes an estimate of the out-of-sample cost of ordering decisions based on a held-out validation set. The lower-level problem learns the optimal coefficients of the decision function on a training set, using only the features selected by the upper-level. We present a mixed integer linear program reformulation for the bilevel program, which can be solved to optimality with standard optimization solvers. Our computational experiments show that the method recovers ground-truth features with accuracy above 96% already for instances with 400 observations. In contrast, regularization-based techniques often fail at feature recovery and even the best-performing methods cannot consistently achieve accuracy values above 90%. Regarding out-of-sample generalization, we achieve improved or comparable cost performance.

This chapter is based on an article published as:

Serrano B., Minner S., Schiffer M., Vidal T. (2024). Bilevel optimization for feature selection in the data-driven newsvendor problem. *European Journal of Operational Research* 315(2):703-714. https://doi.org/10.1016/j.ejor.2024.01.025

3.1 Introduction

The newsvendor problem and its variants have served as fundamental building blocks for models in inventory and supply chain management. In the classical newsvendor problem, a decision-maker optimizes the inventory of a perishable product that has a stochastic demand with a known distribution. However, having complete knowledge of the demand distribution is a strong assumption that does not hold in practice: often, the only information available is a limited set of historical data. Against this background, data-driven approaches became popular and strive to use past demand data to inform the newsvendor's ordering decisions.

In this context, we study the feature-based newsvendor problem (cf. Ban and Rudin 2019; Beutel and Minner 2012) in which the decision-maker has access not only to historical demand observations but also to a set of feature variables—often referred to as contextual information or covariates—that may provide partial information about future realizations of the uncertain demand. For example, consider a retail company that sells products in stores at different locations and with different assortments, e.g., newspaper stands, a chain of restaurants, or fashion retail stores. Then, features could include information about store location, product characteristics, weather forecasts, e.g., temperature and expected precipitation, day of the week, seasonal trends, social media events, competitor activity, epidemic outbreaks, supplier promotions, regulatory changes, and information about holidays or special events close to the store location. In a broader sense, feature variables can include any information that is available to the decision-maker before or at the time instant when ordering decisions are made, e.g., past sales of related products to capture correlated demands, or forecasts based on human expert knowledge. In the context of a car retailer, Tian and Zhang (2023) applied a feature-based newsvendor model where features included historical sales data, textual online reviews, search traffic data, and macroeconomic indicator data. Moreover, the newsvendor model has applications beyond inventory management, such as finding the optimal staffing level of nurses for hospital emergency rooms (Ban and Rudin, 2019), where the uncertain demand corresponds to the number of arriving patients and features include the day of the week and time of the day, among others.

Companies nowadays have large amounts of data that are used to train machine learning models with the aim of improving operational decisions. In practice, such models often suffer from overfitting to the training data, or lack explainability, which is crucial, e.g., when dealing with high-stakes decisions. In this setting, selecting a subset of the available features can lead to sparser, more explainable models with improved out-of-sample performance. Against this background, we investigate the challenge of feature selection (cf. Kuhn and Johnson, 2019; Molina, Belanche, and Nebot, 2002): given a data

set with a possibly large set of feature variables, we aim to learn a linear decision function for the feature-based newsvendor that can generalize to out-of-sample data, utilizing only relevant features.

The goal of this paper is to propose an approach to feature selection based on bilevel optimization. Accordingly, we learn a linear decision function for the feature-based newsvendor based on a training data set, where the set of features available for training is restricted to a given subset of all features. The problem of learning a linear decision function corresponds to the lower-level problem of our bilevel optimization formulation, while the problem of selecting a subset of available features corresponds to the upper-level problem. In the upper-level problem, we search for a subset of features that minimizes the out-of-sample cost measured on a held-out data set, which we denote as the validation data set. In the remainder of this section, we first review related literature before we detail our contribution and describe the organization of this paper.

3.1.1 Related Works

Our work relates to the fields of data-driven optimization for the newsvendor problem, and more broadly to prescriptive analytics, machine learning, and bilevel programming. We briefly review the most related papers in the following.

Newsvendor problem. Research on the newsvendor problem often assumed a decision-maker with full knowledge about the demand distribution, and considered various settings, e.g., with different objectives or utility functions (Chen et al., 2007; Wang and Webster, 2009), pricing policies (Petruzzi and Dada, 1999), and multi-product or multi-period settings (Kogan and Lou, 2003; Lau and Lau, 1996). For general surveys on newsvendor models and extensions, we refer the interested reader to Khouja (1999), Qin et al. (2011) and Choi (2012). In practice, the decision-maker often has only a finite set of demand observations and cannot estimate the true underlying distribution, which motivated works on the distribution-free newsvendor problem. In this context, the seminal work of Scarf (1958) derived the optimal order quantity that maximizes profit against the worst-case demand distribution, assuming that only the mean and variance of demand are known. For a review on the distribution-free newsvendor and extensions thereof, we refer to Gallego and Moon (1993) and Moon and Gallego (1994), and Yue, Chen, and Wang (2006). Later works on this problem variant assumed additional information about the demand distribution, such as percentiles (Gallego, Ryan, and Simchi-Levi, 2001), symmetry, and unimodality (Perakis and Roels, 2008).

In contrast to working with moments or distributional parameters, data-driven approaches build directly upon a sample of available data that reflects realizations of the

underlying uncertainty. In this context, a common solution approach is sample average approximation (SAA) (cf. Kleywegt, Shapiro, and Mello 2002; Shapiro 2003). Levi, Roundy, and Shmoys (2007) applied SAA for the single-period featureless newsvendor problem and established upper bounds on the number of samples required to achieve a specified relative error. In this course, Levi, Perakis, and Uichanco (2015), Cheung and Simchi-Levi (2019), and Besbes and Mouchtaki (2023) further improved upon previous SAA bounds. Ban (2020), Besbes and Muharremoglu (2013), and Sachs and Minner (2014) studied the impact of demand censoring, i.e., a problem variant in which only *sales* observations are available but excess demand is not recorded. They derived upper and lower bounds on the difference between the cost achieved by a policy and the optimal cost with knowledge of the demand distribution. Adopting a robust optimization perspective, Bertsimas and Thiele (2005) proposed a data-driven approach that can be reformulated as a linear program (LP) and trades off higher profits for a decrease in the downside risk. Robust optimization approaches were also investigated by Bertsimas and Thiele (2006) and See and Sim (2010) for a multi-period inventory problem. Finally, many authors applied data-driven distributionally robust approaches for dealing with uncertainty in the context of multi-item newsvendor problems (see, e.g., Ben-Tal et al. 2013; Hanasusanto et al. 2015; Wang, Glynn, and Ye 2016, and Bertsimas, Gupta, and Kallus 2018).

Despite numerous extensions to the newsvendor problem, most data-driven approaches consider only demand data but no feature variables to be available. However, ignoring the presence of features can lead to inconsistent decisions as shown in Ban and Rudin (2019). In the following, we review papers that also consider the presence of features in the context of data-driven optimization.

Data-driven optimization. Beyond the newsvendor problem, some recent works have studied the integration of estimation and optimization. In particular, Bertsimas and Kallus (2019) proposed a framework for feature-based stochastic optimization problems based on a weighted SAA approach, in which the weights are generated by machine learning methods, such as k-nearest neighbors regression, local linear regression, classification and regression trees, or random forests. Elmachtoub and Grigas (2021) focused on problems with a linear objective and used features to learn a prediction model for the stochastic cost vector. They proposed a modified loss function that directly leverages the structure of the optimization problem instead of minimizing a standard prediction error, such as the least squares loss. Despite this modification, their approach still handles prediction and optimization as separate tasks and does not integrate them into a one-step process. Mandi et al. (2020) further adapted the approach from Elmachtoub and Grigas (2021) to solve some hard combinatorial problems, e.g., by proposing tailored warm-starting techniques.

In the context of the feature-based newsvendor, Beutel and Minner (2012) integrated es-

timation and optimization by learning a decision function that predicts ordering decisions directly from features, opposed to first estimating the demand and then optimizing the inventory level. The proposed model formulation is an LP that solves an Empirical Risk Minimization (ERM) problem over a training data set. Oroojlooyjadid, Snyder, and Takáč (2020) and Zhang and Gao (2017) applied neural networks to the newsvendor problem, proposing specific loss functions that consider the impact of inventory shortage and holding costs. Huber et al. (2019) provided an empirical evaluation of different data-driven approaches for the feature-based newsvendor and compared their performance against *model-based* approaches, which model the uncertainty through a demand distribution assumption. Their experiments on real-world data showed that data-driven approaches outperform their model-based counterparts in most cases. Further, Mandl and Minner (2023) studied a multi-period commodity procurement problem under price uncertainty and proposed a data-driven model to derive optimal purchase policies based on economic indicators.

Regarding feature selection, Ban and Rudin (2019) extended the model of Beutel and Minner (2012) by including a regularization term to the objective function, which penalizes the complexity of the solution, thereby favoring the selection of fewer features. However, feature selection is not the main focus of Ban and Rudin (2019), and an open challenge remains regarding the specification of the regularization parameter, for which heuristics are often employed. In this work, we avoid regularization by formalizing the task of feature selection as a bilevel optimization problem for which we provide a tractable single-level reformulation.

Bilevel optimization in machine learning. Bilevel optimization has been applied in the field of machine learning for hyperparameter optimization (Bennett et al., 2006, 2008; Franceschi et al., 2018; Mackay et al., 2019) and feature selection (Agor and Özaltın, 2019). In particular, Bennett et al. (2006, 2008) proposed a bilevel program for optimizing the hyperparameters of a support vector regression model. They reformulated the model into a single-level nonlinear program and employed off-the-shelf solvers based on Sequential Quadratic Programming (Fletcher and Leyffer, 2002). Franceschi et al. (2018) also proposed a bilevel programming approach for hyperparameter optimization, highlighting connections to meta-learning, and solved it with a gradient-based method.

Only Agor and Özaltın (2019) addressed feature selection as a bilevel optimization problem in the context of classification models, e.g., Lasso-based logistic regression and support vector machines. However, their bilevel formulations do not apply to our problem setting, since the feature-based newsvendor combines aspects from supervised learning, i.e., regression, and data-driven optimization. Moreover, the solution method of Agor and Özaltın (2019) consists of a tailored genetic algorithm, which does not provide solution-quality

guarantees. In contrast, our methodology is based on mixed integer linear programming (MILP) and allows to optimally solve the proposed bilevel programming formulations.

Mixed integer optimization for feature selection. The problem of feature selection, also referred to as the best subset selection problem, has been extensively studied in machine learning research. For an introduction to different methods, we refer to Guyon and Elisseeff (2003), Kuhn and Johnson (2019), and Molina, Belanche, and Nebot (2002). Recent works proposed mixed integer programming (MIP) formulations for feature selection in the context of multiple linear regression (Takano and Miyashiro, 2020), classification with support vector machines (Maldonado et al., 2014), and cluster analysis (Benati and García, 2014), among others. In particular, Maldonado et al. (2014) proposed a MILP formulation that simultaneously learns the classifier and selects relevant features by limiting the number of selected features using a budget constraint.

For regression models, Miyashiro and Takano (2015) proposed a mixed integer second-order cone programming formulation for feature selection with respect to various statistical criteria. Gómez and Prokopyev (2021) showed that a formulation based on mixed integer fractional programming has a stronger convex relaxation than Miyashiro and Takano (2015). Bertsimas, King, and Mazumder (2016) considered a MIP formulation for linear regression under a cardinality constraint on the subset of selected features, and proposed discrete first-order algorithms and tailored warm-starting techniques. Park and Klabjan (2020) solved a mixed integer quadratically constrained program for feature selection with respect to criteria such as the mean squared error and the mean absolute error. Kimura and Waki (2018) proposed a mixed integer nonlinear program and a tailored branch-and-bound algorithm for minimizing the Akaike information criterion (AIC) criterion. However, many statistical criteria designed for linear regression are not meaningful in the context of feature-based stochastic optimization, as we are interested in the impact of feature selection in the downstream newsvendor problem. Therefore, we adopt a cross-validation criterion that evaluates the out-of-sample cost of a selection of features on a held-out validation data set, which requires only mild assumptions in contrast to other information criteria (cf. Takano and Miyashiro 2020).

Related work by Takano and Miyashiro (2020) proposed a bilevel optimization formulation for feature selection based on a cross-validation criterion. They focused on multiple linear regression and proposed a single-level reformulation based on mixed integer quadratic programming (MIQP). Due to its computational complexity, they cannot optimally solve the MIQP in most cases. A key difference to our work is that they rely on regularization in the lower-level problem and require a regularization parameter value to be given a priori. In the experiments, they employ a grid search method to tune this parameter and leave it for future research to devise a formulation that simultaneously

selects features and the optimal regularization value. As we discuss later, the problem of hyperparameter tuning itself can be cast as a bilevel optimization problem for which currently no tractable reformulation exists. In contrast, we propose a bilevel program for feature selection which does not require regularization and has a MILP single-level reformulation.

3.1.2 Contribution

We close the research gaps outlined above by proposing a novel bilevel optimization model that directly incorporates feature selection into solving the data-driven newsvendor problem. Specifically, our contribution is fourfold. First, we introduce a bilevel program designed for feature selection, which we denote the Bilevel Feature Selection (BFS) model. In contrast to regularization-based methods, which penalize the norm of the solution vector, BFS captures the more intuitive notion of selecting a subset of features that minimizes an estimate of the out-of-sample cost, measured on a held-out validation set. We reformulate the bilevel program into a single-level optimization problem, which we solve to optimality with off-the-shelf optimization solvers. Second, we extend the BFS model to accommodate cross-validation strategies, which further improves its solution quality. Third, to illustrate the drawback of regularization-based methods for feature selection, we present a bilevel program, which we refer to as Bilevel Hyperparameter Optimization (BHO), that searches for the optimal hyperparameter for the regularized ERM model (cf. Ban and Rudin 2019). BHO formally describes the optimization model that established hyperparameter optimization methods implicitly solve by means of heuristics, such as grid search, random search, or Bayesian optimization. Fourth, we conduct extensive numerical experiments, using synthetic instances with correlated features. We compare the proposed BFS models against regularization-based methods in terms of out-of-sample performance and ground-truth feature recovery. We further compare the methods' behavior under demand misspecification, assuming a nonlinear demand model. Our results show that the proposed BFS approach consistently achieves higher accuracy in feature recovery. In most cases, we also observe an improvement in out-of-sample cost performance, i.e., a decrease in test cost.

3.1.3 Organization

The remainder of this paper is structured as follows. In Section 3.2, we review the model formulations for the classical newsvendor and the feature-based newsvendor problem. Section 3.3 presents the BHO and the BFS models, and consecutively extends the BFS to cross-validation. Section 3.4 describes our experimental design, and Section 3.5 presents

the results comparing the proposed method against state-of-the-art techniques based on regularization. Section 3.6 concludes this paper and gives an outlook on future research.

3.2 Fundamentals

In the classical newsvendor problem, a risk-neutral decision-maker sets the order quantity of a product before observing its uncertain demand. Here, the objective is to minimize the expected cost:

$$\min_{q \geq 0} \mathbb{E}\left[C(q; d)\right], \tag{3.1}$$

where q is the order quantity, $d \sim \mathcal{D}$ is the random variable representing the uncertain demand,

$$C(q; d) := b(d - q)^+ + h(q - d)^+ \tag{3.2}$$

is the cost of ordering q units and observing demand d, based on the per unit shortage cost b for lost profits and unit holding cost h, corresponding to the procurement cost of unsold products discounted by their unit salvage value. If the demand distribution is known, then the optimal decision q^* is given at the $b/(b + h)$ quantile of its cumulative distribution function.

In practice, the demand distribution is often not known. We consider the feature-based newsvendor problem, in which the decision-maker has access to historical demand data and contextual information given by a set of feature variables $\mathbf{x} \in \mathbb{R}^{m+1}$ (cf. Beutel and Minner 2012). Here, the uncertain demand d and feature variables $\mathbf{x}$ follow an (unknown) joint probability distribution $(\mathbf{x}, d) \sim \mathcal{X} \times \mathcal{D} = \mathcal{Z}$. The decision-maker's objective is to minimize the expected cost conditioned on the observed features:

$$\min_{q \geq 0} \mathbb{E}\left[C(q(\mathbf{x}); d(\mathbf{x}))|\mathbf{x}\right]. \tag{3.3}$$

One approach to solve the feature-based newsvendor is to separate the estimation and optimization problems, i.e., one first estimates the conditional demand distribution from historical data and then optimizes the order quantity based on new feature observations. One drawback of this approach is that the first step's estimation problem does not account for the asymmetry in the newsvendor cost function, related to under- and over-predicting demand. To address this issue, Beutel and Minner (2012) proposed to integrate estimation and optimization into a one-step process, by introducing a linear decision function that maps feature observations directly to ordering decisions. To learn the optimal coefficients of the decision function, one minimizes the empirical cost over a data set with demand and feature observations.

Let $\{(\mathbf{x}_i, d_i)\}_{i \in S}$ be a data set indexed by $S = \{1, \ldots, n\}$, where $\mathbf{x}_i$ is an $(m+1)$-dimensional feature vector and d_i is a scalar demand observation. Let $J = \{0, \ldots, m\}$ denote the set of feature indices. We assume that $x_i^0 = 1$ represents the feature-independent intercept term, for all $i \in S$. In this setting, Beutel and Minner (2012) consider a linear decision function of the form:

$$q(\mathbf{x}) = \beta^0 + \sum_{j=1}^{m} \beta^j x^j = \boldsymbol{\beta}^\top \mathbf{x}, \tag{3.4}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{m+1}$ is the parameter vector, whose values are learned by minimizing the empirical cost on data set S. Upon observing new feature values, the decision-maker can then directly decide upon the order quantity instead of first estimating the uncertain demand.

Since the learned decision function may overfit to the in-sample data set S, it is common practice in machine learning to evaluate the out-of-sample generalization on a separate test data set S_{test}. To avoid overfitting and improve the out-of-sample generalization, Ban and Rudin (2019) proposed an extension of Beutel and Minner (2012) by integrating a regularization term into the loss function. Accordingly, the objective comprises a trade-off between minimizing the empirical in-sample cost and the regularization term, with a constant hyperparameter balancing these two terms:

$$\text{(ERM-}\ell_p\text{)} \qquad \min_{\boldsymbol{\beta}} \quad \frac{1}{|S|} \sum_{i \in S} C(q_i \,; d_i) + \lambda \, ||\boldsymbol{\beta}||_p \tag{3.5}$$

$$\text{s.t.} \quad q_i = \boldsymbol{\beta}^\top \mathbf{x}_i \qquad \forall i \in S, \tag{3.6}$$

where $\lambda \geq 0$ is the regularization hyperparameter and $||\boldsymbol{\beta}||_p$ is the ℓ_p-norm of the vector $\boldsymbol{\beta}$. Depending on the choice of p in the regularization, the resulting model may be a MILP, an LP, or a second-order cone program, for ℓ_0, ℓ_1, and ℓ_2-norm regularization, respectively. Effectively, regularization enables feature selection by penalizing the complexity of the solution, thereby favoring sparse solution vectors.

3.3 Methodology

We start this section presenting the BHO model, which incorporates hyperparameter fitting in (ERM-ℓ_p). Then, we introduce the BFS model as an alternative bilevel program that avoids regularization.

3.3.1 Bilevel Hyperparameter Optimization (BHO)

In Section 2, we assumed the hyperparameter λ as introduced in (ERM-ℓ_p) to be given. However, identifying λ constitutes a challenge in itself as a respective misspecification can significantly reduce cost performance. To parametrize λ correctly, one may utilize existing techniques for hyperparameter optimization, which partition the original data set S into a training set T and a validation set V. On the training set, one learns the model parameters for a fixed hyperparameter value. Using the validation set, one can then assess the cost of the trained model for a variety of hyperparameter values, to finally choose the value λ^* that leads to a minimum cost on the validation set. Next, we present the BHO formulation, which models the search for the optimal hyperparameter λ^* as a bilevel optimization problem.

We introduce variables u_i to model the inventory shortage and variables o_i to model the surplus inventory at the end of period $i \in T \cup V$. In the following bilevel programming formulation, the upper-level (UL) problem searches for an optimal regularization value $\lambda^* \geq 0$ that minimizes cost on the validation set V. In turn, the lower-level (LL) problem solves the feature-based newsvendor, as stated in (ERM-ℓ_p), on the training set T:

$$\text{(BHO-}\ell_p\text{ UL)} \qquad C^*_{\text{BHO}} = \min \quad \frac{1}{|V|}\sum_{i\in V}(bu_i + ho_i) \tag{3.7}$$

$$\text{s.t.} \quad u_i \geq d_i - \boldsymbol{\beta}^\top \mathbf{x}_i \qquad \forall i \in V \tag{3.8}$$

$$o_i \geq \boldsymbol{\beta}^\top \mathbf{x}_i - d_i \qquad \forall i \in V \tag{3.9}$$

$$u_i \geq 0,\, o_i \geq 0 \qquad \forall i \in V \tag{3.10}$$

$$\lambda \geq 0 \tag{3.11}$$

$$\boldsymbol{\beta} \in \Omega_p(\lambda), \tag{3.12}$$

where $\Omega_p(\lambda)$ is the set of optimal solutions $\boldsymbol{\beta}$ to the lower-level problem, parameterized by λ:

$$\text{(BHO-}\ell_p\text{ LL)} \qquad \Omega_p(\lambda) \coloneqq \arg\min \quad \frac{1}{|T|}\sum_{i\in T}(bu_i + ho_i) + \lambda\,||\boldsymbol{\beta}||_p^2 \tag{3.13}$$

$$\text{s.t.} \quad u_i \geq d_i - \boldsymbol{\beta}^\top \mathbf{x}_i \qquad \forall i \in T \tag{3.14}$$

$$o_i \geq \boldsymbol{\beta}^\top \mathbf{x}_i - d_i \qquad \forall i \in T \tag{3.15}$$

$$\boldsymbol{\beta} \in \mathbb{R}^{m+1} \tag{3.16}$$

$$u_i \geq 0,\, o_i \geq 0 \qquad \forall i \in T \tag{3.17}$$

The upper-level objective (3.7) minimizes the newsvendor cost on the validation set V and the lower-level objective (3.13) minimizes the regularized newsvendor cost on the training

set T. Constraints (3.8) and (3.14) define the inventory shortage for period $i \in V$ and $i \in T$, respectively, given the decision function parametrized by $\boldsymbol{\beta}$. Constraints (3.9) and (3.15) define the surplus inventory at period $i \in V$ and $i \in T$. Constraints (3.10), (3.11), (3.12), (3.16), and (3.17) define the variable domains.

So far, we define the BHO formulation in (3.7)–(3.17) for a general ℓ_p-norm, which leads to a different model for different p. In the following, we illustrate some properties of BHO under the special case of the ℓ_0-norm regularization, which minimizes the number of non-zero elements in the $\boldsymbol{\beta}$ vector. In this case, we introduce the binary variable z^j to indicate whether coefficient β^j is non-zero. The lower-level problem can then be formulated as a MIP:

$$\text{(BHO-}\ell_0\text{ LL)} \qquad \Omega_0(\lambda) \coloneqq \arg\min \quad \frac{1}{|T|}\sum_{i\in T}(bu_i + ho_i) + \lambda \sum_{j\in J} z^j \tag{3.18}$$

$$\text{s.t.} \quad (3.14)\text{–}(3.17)$$

$$\beta^j = 0 \text{ if } z^j = 0 \qquad \forall j \in J \tag{3.19}$$

$$z^j \in \{0,1\} \qquad \forall j \in J, \tag{3.20}$$

where Constraints (3.14)–(3.17) define the shortage and surplus inventory and Constraints (3.19) enforce that $\beta^j = 0$ if the corresponding feature is not selected.

The BHO formulation (3.7)–(3.17) generalizes many common methods for hyperparameter optimization. To avoid the high computational effort of solving the BHO model to optimality, existing methods relax the assumption that λ can take any value in $\mathbb{R}_{\geq 0}$, and consider a finite support set $\Lambda \subseteq \mathbb{R}_{\geq 0}$ instead (Bergstra and Bengio 2012; Bergstra, Yamins, and Cox 2013). For example, suppose the values in Λ are equally spaced along a grid, i.e., a line segment, then the resulting model corresponds to the well-known *grid search* method. If the values in Λ are randomly selected in a closed region, then the formulation describes the *random search* method. Other approaches, e.g., based on Bayesian optimization, would perform an adaptive search, iteratively selecting a value λ for the upper-level variable and then optimizing the lower-level problem. The iterative selection of new values for λ depends on the validation performance of previously selected points. In essence, current methods for hyperparameter optimization, such as the examples described above, are heuristics that avoid solving the BHO model to optimality.

3.3.2 Bilevel Feature Selection (BFS)

To remedy the drawback of BHO, we introduce a bilevel programming formulation specifically designed for feature selection. Instead of penalizing the number of selected features, we propose a more intuitive model, in which the upper-level problem selects a subset of

features that minimize the empirical cost on a validation set. We then reformulate the resulting model into a single-level problem, which is computationally more tractable, and finally compare the proposed BFS and BHO models.

Consider our original data set S, which we partition into a training set T and a validation set V. We introduce binary variables z^j, for $j \in J$, to indicate whether feature j is marked as relevant ($z^j = 1$) or not ($z^j = 0$). In the upper-level, BFS selects a subset of features that minimizes the empirical cost on the validation set V. The lower-level problem then learns the optimal coefficients of the decision function in the training set T by solving the ERM model using only the features selected in the upper-level. We formulate the resulting upper-level problem as follows:

$$\text{(BFS UL)} \qquad C^*_{\text{BFS}} = \min \quad \frac{1}{|V|}\sum_{i\in V}(bu_i + ho_i) \tag{3.21}$$

$$\text{s.t.} \quad (3.8)\text{–}(3.10)$$

$$z^j \in \{0,1\} \qquad \forall j \in J \tag{3.22}$$

$$\boldsymbol{\beta} \in \Pi_0(\mathbf{z}), \tag{3.23}$$

where $\Pi_0(\mathbf{z})$ is the set of optimal solutions $\boldsymbol{\beta}$ to the lower-level problem:

$$\text{(BFS LL)} \qquad \Pi_0(\mathbf{z}) \coloneqq \arg\min \quad \frac{1}{|T|}\sum_{i\in T}(bu_i + ho_i) \tag{3.24}$$

$$\text{s.t.} \quad (3.14)\text{–}(3.17)$$

$$\beta^j = 0 \text{ if } z^j = 0 \qquad \forall j \in J \tag{3.25}$$

The upper and lower-level objectives (3.21) and (3.24) minimize the newsvendor cost on the validation set V and training set T, respectively. Constraints (3.8)–(3.10) and (3.14)–(3.17) define the shortage and surplus inventory. Constraints (3.22)–(3.23) define the variable domains and Constraints (3.25) ensure that $\beta^j = 0$ if feature j is not selected.

We reformulate Model (3.21)–(3.25) by substituting the lower-level problem by its Karush-Kuhn-Tucker (KKT) conditions (cf. Cao and Chen 2006; Fontaine and Minner 2014). We introduce the dual variables μ_i, and γ_i corresponding to constraints (3.14) and (3.15) of the lower-level problem. The equivalent single-level (SL) optimization problem

can then be expressed by using indicator constraints:

$$\text{(BFS SL)} \qquad \min \quad \frac{1}{|V|}\sum_{i\in V}(bu_i + ho_i) \tag{3.26}$$

$$\text{s.t.} \quad (3.8)\text{–}(3.10),\ (3.14)\text{–}(3.17),\ (3.22),\ (3.25)$$

$$\frac{1}{|T|}\sum_{i\in T}(bu_i + ho_i) \leq \sum_{i\in T}(\gamma_i - \mu_i)d_i \tag{3.27}$$

$$\mu_i + \frac{b}{|T|} \geq 0 \qquad \forall i \in T \tag{3.28}$$

$$\gamma_i + \frac{h}{|T|} \geq 0 \qquad \forall i \in T \tag{3.29}$$

$$\sum_{i\in T}(\mu_i - \gamma_i)x_i^j = 0 \text{ if } z^j = 1 \qquad \forall j \in J \tag{3.30}$$

$$\mu_i \leq 0,\ \gamma_i \leq 0 \qquad \forall i \in T \tag{3.31}$$

As before, Constraints (3.8)–(3.10) and (3.14)–(3.17) define the shortage and surplus inventory. Constraints (3.22) and (3.25) model the selection of features. Constraint (3.27) represents the optimality condition of the lower-level problem, by comparing its primal objective value with the corresponding dual objective value. Constraints (3.28) and (3.29) are the dual constraints of the lower-level problem associated with primal variables u_i and o_i for $i \in T$. Constraints (3.30) are the dual constraints related to the primal variables β^j for $j \in J$, and Constraints (3.31) define the domain of the dual variables. The single-level reformulation has $2n + 2|T|+2|J|$ variables and $2n + 2|T|+2|J|+1$ constraints.

The BFS model shares some similarities with the BHO model. Both models have the same upper-level objective and the lower-level objectives differ only in the presence of the regularization term. We provide an overview of the main properties of both models in Table 3.1. The main advantage of the BFS model regarding tractability is due to the existence of binary variables being limited to the upper-level problem. Consequently, we can reformulate the BFS model into a MILP and leverage existing exact methods to find optimal solutions at a limited scale or heuristic methods to efficiently solve larger instances.

Moreover, the following results show that the optimal cost of the BFS model is a lower bound to the optimal cost of the BHO model when adopting ℓ_0-norm regularization.

Lemma 3.1 *Given a fixed selection of features* $\mathbf{z}$ *for both BHO and BFS, i.e.,* $\mathbf{z}_{BHO} = \mathbf{z}_{BFS} = \mathbf{z}'$ *(assuming that* $\mathbf{z}'$ *is feasible for both problems), solving the remaining problems for the rest of the decision variables yields optimal solutions* $\boldsymbol{\beta}_{BHO}|_{\mathbf{z}=\mathbf{z}'} = \boldsymbol{\beta}_{BFS}|_{\mathbf{z}=\mathbf{z}'}$ *with costs* $C_{BHO}|_{\mathbf{z}=\mathbf{z}'} = C_{BFS}|_{\mathbf{z}=\mathbf{z}'}$.

Formulation		BHO (ℓ_0-norm reg.)	BFS
Upper-level	Objective	minimize validation cost	minimize validation cost
	Variables	$\lambda \in \mathbb{R}_{\geq 0}$	$\mathbf{z} \in \{0,1\}^{m+1}$
Lower-level	Objective	minimize training cost + regularization	minimize training cost
	Variables	$\boldsymbol{\beta} \in \mathbb{R}^{m+1}$, $\mathbf{z} \in \{0,1\}^{m+1}$	$\boldsymbol{\beta} \in \mathbb{R}^{m+1}$

Table 3.1: Comparison between BFS and BHO with ℓ_0-norm regularization

Proof. By fixing $\mathbf{z}_{\text{BHO}} = \mathbf{z}'$, the regularization term in the lower-level objective becomes constant and λ does not influence the optimal solution. Therefore, we can ignore regularization and the lower-level problem of the BHO becomes equal to the lower-level problem of the BFS, leading to $\boldsymbol{\beta}_{\text{BHO}}|_{\mathbf{z}=\mathbf{z}'} = \boldsymbol{\beta}_{\text{BFS}}|_{\mathbf{z}=\mathbf{z}'}$ as the optimal solution. Since the upper-level objectives are equal in both models, the optimal costs will be equal: $C_{\text{BHO}}|_{\mathbf{z}=\mathbf{z}'} = C_{\text{BFS}}|_{\mathbf{z}=\mathbf{z}'}$. □

Proposition 3.1 *The optimal cost of the BFS model is a lower bound for the optimal cost of the BHO model with ℓ_0-norm regularization: $C^*_{BHO} \geq C^*_{BFS}$.*

Proof. Let $\mathbf{z}_{\text{BHO}} = \mathbf{z}^*$ be the optimal selection of features according to BHO with cost C^*_{BHO}. Suppose that $\mathbf{z}_{\text{BFS}} = \mathbf{z}'$ is a solution to BFS, such that $C^*_{\text{BHO}} < C_{\text{BFS}}|_{\mathbf{z}=\mathbf{z}'}$. We can always improve the cost of BFS by setting $\mathbf{z}_{\text{BFS}} = \mathbf{z}^*$ in the upper-level problem. Because of Lemma 3.1, this will result in a new solution with cost $C_{\text{BFS}}|_{\mathbf{z}=\mathbf{z}^*} = C^*_{\text{BHO}} \geq C^*_{\text{BFS}}$. □

3.3.3 Bilevel Feature Selection with Cross-Validation (BFS-CV)

Cross-validation strategies often improve the generalization ability of machine learning models and prevent overfitting by using data re-sampling methods. Accordingly, we extend the BFS model to cross-validation instead of simple hold-out validation. We consider K training-validation splits of the data and search for the set of features that minimize the average cost over all K validation sets. For each $k \in [K] = \{1, \ldots, K\}$, we consider a subset of observations $S_k \subseteq S$ sampled from the original data set S. Analogously to BFS, we partition the set S_k into a training set T_k and a validation set V_k. We introduce variables u_{ik} and o_{ik} to model the inventory shortage and surplus, respectively, at the end of period $i \in T_k \cup V_k$ for each training-validation split $k \in [K]$. We then learn the model parameters $\boldsymbol{\beta}_k \in \mathbb{R}^{m+1}$ using the corresponding training set T_k, and select features by minimizing the average validation cost over all validation sets V_k for $k \in [K]$. The

resulting problem is a bilevel program with K lower-level problems:

$$\text{(BFS-CV UL)} \quad \min \quad \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|V_k|}\sum_{i\in V_k}(bu_{ik}+ho_{ik}) \tag{3.32}$$

$$\text{s.t.} \quad u_{ik} \geq d_i - \boldsymbol{\beta}_k^\top \mathbf{x}_i \qquad \forall\, k\in[K], \forall i\in V_k \tag{3.33}$$

$$o_{ik} \geq \boldsymbol{\beta}_k^\top \mathbf{x}_i - d_i \qquad \forall\, k\in[K], \forall i\in V_k \tag{3.34}$$

$$u_{ik}\geq 0,\, o_{ik}\geq 0 \qquad \forall\, k\in[K], \forall i\in V_k \tag{3.35}$$

$$z^j \in \{0,1\} \qquad \forall j\in J \tag{3.36}$$

$$\boldsymbol{\beta}_k \in \Pi_k(\mathbf{z}) \qquad \forall\, k\in[K], \tag{3.37}$$

where $\Pi_k(\mathbf{z})$ is the set of optimal solutions corresponding to the k^{th} lower-level problem:

$$\text{(BFS-CV LL)} \quad \Pi_k(\mathbf{z}) \coloneqq \arg\min \quad \frac{1}{|T_k|}\sum_{i\in T_k}(bu_{ik}+ho_{ik}) \tag{3.38}$$

$$\text{s.t.} \quad u_{ik} \geq d_i - \boldsymbol{\beta}_k^\top \mathbf{x}_i \qquad \forall i\in T_k \tag{3.39}$$

$$o_{ik} \geq \boldsymbol{\beta}_k^\top \mathbf{x}_i - d_i \qquad \forall i\in T_k \tag{3.40}$$

$$u_{ik}\geq 0,\, o_{ik}\geq 0 \qquad \forall i\in T_k \tag{3.41}$$

$$\beta_k^j = 0 \text{ if } z^j = 0 \qquad \forall j\in J \tag{3.42}$$

$$\boldsymbol{\beta}_k \in \mathbb{R}^{m+1} \tag{3.43}$$

Constraints (3.33)–(3.35) and (3.39)–(3.41) define the shortage and surplus inventory for period $i\in V_k$ and $i\in T_k$, respectively for each split $k\in[K]$. Constraints (3.36), (3.37), and (3.43) define the variable domains and Constraints (3.42) ensure that $\beta_k^j = 0$ if feature j is not selected for the training-validation split k.

The above model can accommodate different cross-validation strategies, such as K-fold, random permutations (Shuffle & Split), or Leave-P-Out cross-validation (see, e.g., Arlot and Celisse 2010; Hastie, Tibshirani, and Friedman 2009). Each particular choice of cross-validation strategy corresponds to a different approach for constructing the subsets S_k and partitioning the data into T_k and V_k. Moreover, the special case with $K=1$ corresponds to the previously introduced BFS model.

Analogously to BFS, BFS with cross-validation (BFS-CV) can be reformulated into a

single-level MILP:

$$\text{(BFS-CV SL)} \quad \min \quad \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|V_k|}\sum_{i\in V_k}(bu_{ik}+ho_{ik}) \tag{3.44}$$

$$\text{s.t.} \quad (3.33)\text{–}(3.36),\ (3.39)\text{–}(3.43)$$

$$\frac{1}{|T_k|}\sum_{i\in T_k}(bu_{ik}+ho_{ik}) \le \sum_{i\in T_k}(\gamma_{ik}-\mu_{ik})d_i \qquad \forall\, k\in[K] \tag{3.45}$$

$$\mu_{ik}+\frac{b}{|T_k|}\ge 0 \qquad \forall k\in[K],\ \forall i\in T_k \tag{3.46}$$

$$\gamma_{ik}+\frac{h}{|T_k|}\ge 0 \qquad \forall k\in[K],\ \forall i\in T_k \tag{3.47}$$

$$\sum_{i\in T_k}(\mu_{ik}-\gamma_{ik})x_i^j = 0 \text{ if } z^j=1 \qquad \forall k\in[K],\ \forall j\in J \tag{3.48}$$

$$\mu_{ik}\le 0,\ \gamma_{ik}\le 0 \qquad \forall k\in[K],\ \forall i\in T_k \tag{3.49}$$

Constraints (3.45) represents the optimality condition of the lower-level problem, for each training-validation split $k \in [K]$. Constraints (3.46) and (3.47) are the dual constraints of the lower-level problem associated with primal variables u_{ik} and o_{ik} for $i \in T_k$ for $k \in [K]$. Constraints (3.48) are the dual constraints related to the primal variables $\boldsymbol{\beta}^k$ for $k \in [K]$, and Constraints (3.49) define the domain of the dual variables. The single-level reformulation has $K(4|T|+2|V|+|J|)+|J|$ variables and $K(4|T|+2|V|+2|J|+1)$ constraints.

3.4 Experimental Design

To benchmark our approaches, we perform extensive computational experiments on synthetic instances. The goals of our computational experiments are fourfold.

(*i*) We evaluate the performance of a MILP approach for the proposed BFS and BFS-CV models in terms of feature selection and generalization to out-of-sample data;

(*ii*) We compare the performance of our approach against existing regularization-based methods;

(*iii*) For each method, we compare the performance of hold-out validation against cross-validation;

(*iv*) We analyze the effect of different instance parameters on each method's performance.

We implemented all methods in `C++`, using CPLEX 20.1 to solve the respective MILP formulations. All experiments were performed on machines with Intel Core i7-6700 CPU

at 3.40 GHz, with 16 GB of RAM, and Ubuntu 16.04.6 LTS operating system, under a time limit of 900 seconds. We provide the source code and data at `https://github.com/tumBAIS/Feature-Selection-Newsvendor`.

3.4.1 Instances

We adapt the experimental setup from Zhu, Huang, and Li (2012) and consider a linear demand model with additive noise:

$$d_{\text{linear}}(\mathbf{x}) = 5 + \boldsymbol{\beta}^\top \mathbf{x} + \epsilon, \tag{3.50}$$

where $\boldsymbol{\beta} = (0, 2, -2, -1, 1, 0, \ldots, 0)^\top / \sqrt{10}$ is an $(m+1)$-dimensional vector representing the ground-truth coefficients. The feature variables $\mathbf{x} \in \mathbb{R}^{m+1}$ are drawn from a multivariate Gaussian distribution with mean zero and covariance matrix with entries $\sigma_{ij} = 0.5^{|i-j|}$, for $(i, j) \in J^2$. The noise term follows a Gaussian distribution $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. We set negative demand values to zero.

We generate instances varying the number of samples n from 40 to 2000 and the number of features m from 8 to 14. We deliberately limit the instance sizes in our experiments so that we can find globally optimal or near-optimal solutions within reasonable computation time. For each configuration, we generate 20 instances to account for variability in the distributions. Additionally, we generate a separate test set with 1000 observations associated with each instance, following the same distributions.

Furthermore, we analyze the impact of demand misspecification, i.e., when the demand is not linearly related to the features. We investigate the following nonlinear demand model (cf. Zhu, Huang, and Li 2012):

$$d_{\text{nonlinear}}(\mathbf{x}) = 10 + \sin\left(2(\boldsymbol{\beta}^\top \mathbf{x})\right) + 2\exp\left(-16(\boldsymbol{\beta}^\top \mathbf{x})^2\right) + \varphi(\boldsymbol{\beta}^\top \mathbf{x})\epsilon, \tag{3.51}$$

where $\varphi(\boldsymbol{\beta}^\top \mathbf{x}) = 1$ for a homoscedastic case and $\varphi(\boldsymbol{\beta}^\top \mathbf{x}) = \exp(\boldsymbol{\beta}^\top \mathbf{x})$ for a heteroscedastic case.

We solve the proposed BFS and BFS-CV models and compare the results against regularization-based methods from the literature. Since our motivation for feature selection is to provide more explainable decisions, we focus on methods based on linear decision functions, which are intrinsically more explainable. We consider the ERM model of Ban and Rudin (2019) with ℓ_0 and ℓ_1-norm regularization and use grid search with 50 break-points to calibrate the regularization parameter. We run each considered method once using a hold-out validation set and once with Shuffle & Split cross-validation (CV). For hold-out validation, we use half of the samples in each instance as a training set and

the other half as a validation set, following the setting in Zhu, Huang, and Li (2012). For cross-validation, we perform $K = 50$ re-sampling iterations, where we sample a subset of size $|S_k| = \min\{200, n\}$, for each $k \in [K]$.

3.4.2 Performance metrics

We assess the ability of each method to recover the ground-truth feature vector, adopting the accuracy measure:

$$\alpha = \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}(\hat{z}_j = z_j^*), \tag{3.52}$$

where $\mathbb{1}(\cdot)$ is the indicator function, $\hat{\mathbf{z}} = [\hat{z}_1, \ldots, \hat{z}_m]$ is the estimated binary feature vector, and $\mathbf{z}^* = [z_1^*, \ldots, z_m^*]$ is the ground-truth vector, defined as $z_j^* = 0$ if $\beta_j^* = 0$, i.e., feature j is non-informative, otherwise $z_j^* = 1$. Our definition of accuracy is analogous to the one commonly adopted for binary classification, where $z_j^* \in \{0, 1\}$ represents the class assigned to feature j.

Additionally, we analyze the cost performance of applying the learned decision functions to out-of-sample data. We therefore evaluate the out-of-sample cost on a separate test data set with 1000 observations. We report the test cost values of each method $\mathcal{M}$ in terms of its percentage deviation relative to the test cost achieved by BFS-CV on the same configuration:

$$\delta(\mathcal{M}) = 100 \times \frac{C_{\mathcal{M}}^{test} - C_{\text{BFS-CV}}^{test}}{C_{\text{BFS-CV}}^{test}}, \tag{3.53}$$

where $C_{\mathcal{M}}^{test}$ is the average newsvendor cost of method $\mathcal{M}$ calculated on the test set. Deviation values greater than zero indicate that BFS-CV improves upon method $\mathcal{M}$ regarding test cost performance, while values below zero indicate that method $\mathcal{M}$ achieves lower test cost than BFS-CV.

3.5 Results

First, we present results concerning instances generated by the linear demand model, and then discuss results on instances with nonlinear demand.

3.5.1 Linear demand model

We analyze how instance size, number of features, noise level, shortage cost, and holding cost affect the performance of each method. Unless otherwise stated, we use a setting with $n = 1000$ samples, $m = 10$ features, a shortage cost of $b = 2$, and a holding cost of $h = 1$ as reference configuration. For the noise term, we consider $\sigma_\epsilon = 1$ as reference

configuration for the Gaussian distribution. We provide results regarding computation times in Appendix 3.A.

Instance size. Figure 3.1 reports the feature recovery accuracy of the different methods for a varying number of samples $n \in [40, 2000]$, averaged over 20 randomly generated instances. In general, BFS-CV achieves the highest accuracy among all methods and faster convergence for increasing n, with accuracy values above 0.9 already for instances with 200 samples. In contrast, existing methods often yield accuracy values below 0.9 and fail to recover the ground-truth features accurately, even for instances with a larger number of samples. We confirm these results at 5% significance level by pairwise Wilcoxon signed-rank tests, and refer to Appendix 3.B for details on the respective p-values.

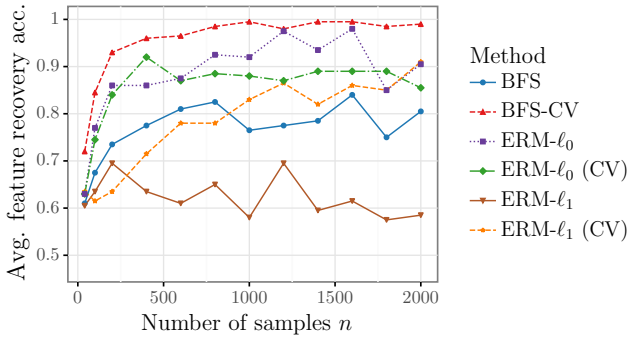


Figure 3.1: Impact of instance size on the accuracy of feature recovery

Besides feature recovery, we evaluate the out-of-sample cost performance of each method. Figure 3.2 shows the distribution of percentage deviations, where a positive deviation indicates that BFS-CV is superior to the respective other method. We split the results in three different plots based on the sample size n, classifying each instance as small, medium, or large. BFS-CV outperforms the other methods in most cases, as the lower quartiles are always above or close to zero. For smaller instances, test cost deviations can be as high as 30% in the best case. As we increase n, all methods present improving test cost performance and the variance in the test cost distribution decreases. Yet, BFS-CV is still superior to the other methods in the wide majority of cases. For large instances, all methods present mostly positive test cost deviations, with values ranging from -1% to 8%.

Number of features. Figure 3.3 shows average feature recovery accuracy values, where we now fix the number of samples to $n = 1000$ and vary the number of features $m \in \{8, 10, 12, 14\}$. For all methods, the number of features does not strongly affect the accuracy performance. Notably, BFS-CV achieves average accuracy values consistently above 0.95, being superior to the other methods, as confirmed by pairwise Wilcoxon tests.

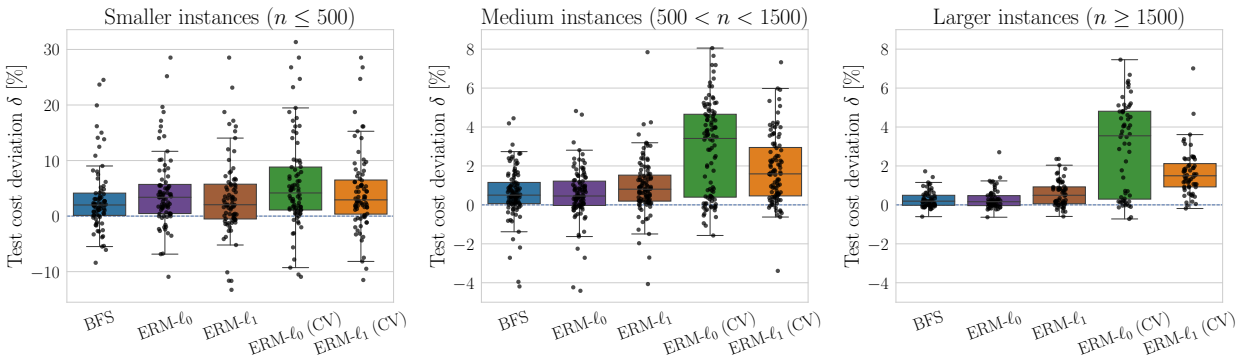


Figure 3.2: Impact of instance size on the test cost performance of different methods, relative to BFS-CV

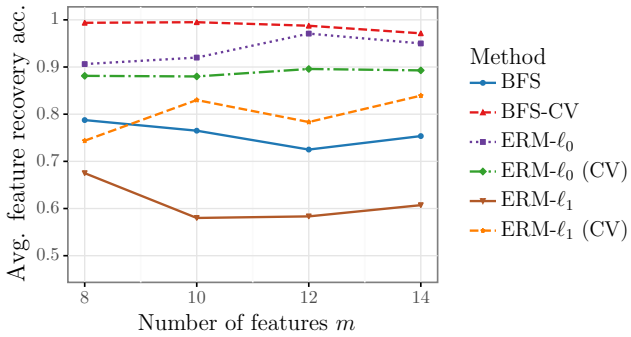


Figure 3.3: Impact of number of features m on the accuracy of feature recovery

Regarding test cost performance, Figure 3.4 shows the distribution of test cost deviations. We focus on large instances ($n \geq 1500$) in this analysis, which have lower variance, so that we can isolate the impact of m on the test cost. For BFS, ERM-ℓ_0, and ERM-ℓ_1, the number of features has no strong influence on the test cost deviations. In contrast, the performance of ERM-ℓ_0 (CV) and ERM-ℓ_1 (CV) shows increasing deviation values for increasing m. In the majority of cases, BFS-CV outperforms the other considered methods.

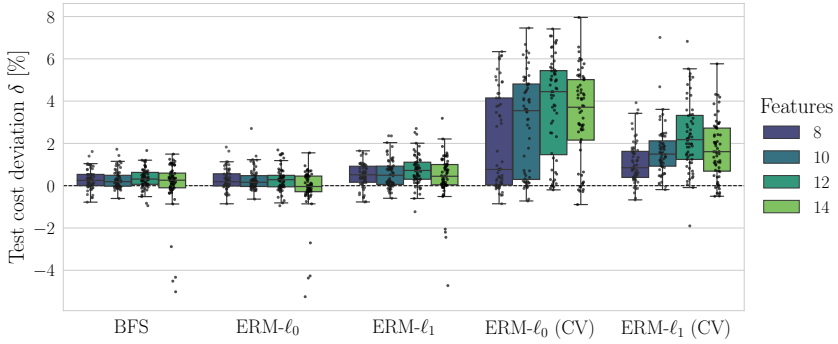


Figure 3.4: Impact of number of features m on the test cost performance of different methods, relative to BFS-CV

Noise level. We vary the coefficient of variation $c_v = \sigma_\epsilon/\mu \in [0.2, 1]$ and report the

average feature recovery accuracy for each method in Figure 3.5. As we increase the level of noise, it becomes harder to recover the informative features and the accuracy of all methods deteriorates. For $c_v \in [0.2, 0.6]$, BFS-CV achieves the highest accuracy among the considered methods. Outside this range, BFS-CV is outperformed by ERM-ℓ_0 (CV) for $c_v = 0.1$ and by ERM-ℓ_0 for $c_v \geq 0.8$, respectively. Still, BFS-CV generally attains comparatively high accuracy, being superior to most other methods.

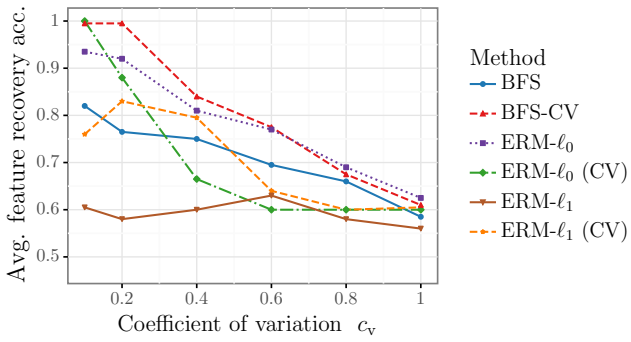


Figure 3.5: Impact of different noise levels on the accuracy of feature recovery

Figure 3.6 illustrates the impact of different noise levels on test cost performance, considering large instances ($n \geq 1500$). Methods BFS, ERM-ℓ_0, and ERM-ℓ_1 mostly outperform BFS-CV for $c_v \geq 0.4$. In such cases, test cost deviations range from -4% to 4%, indicating that BFS-CV achieves comparable results even when its performance is inferior to other methods. Methods ERM-ℓ_0 (CV) and ERM-ℓ_1 (CV) perform comparatively worse, with mostly positive deviations and larger variance in the distribution.

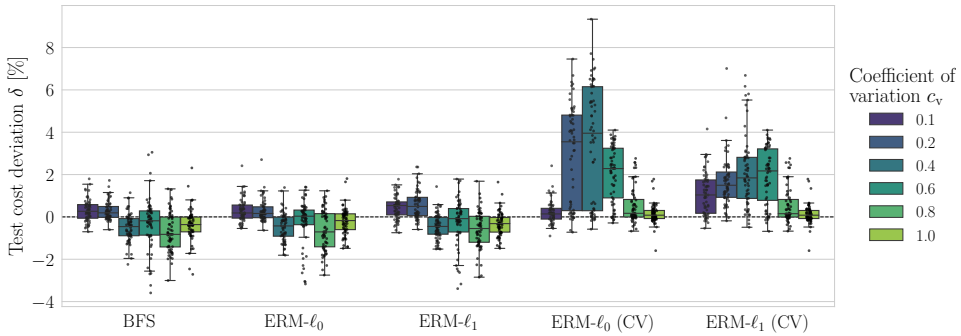


Figure 3.6: Impact of noise level on the test cost performance of different methods, relative to BFS-CV

Shortage and holding costs. Figure 3.7 displays results on the accuracy performance as a function of the newsvendor ratio $b/(b+h)$, by varying $(b, h) \in [1, 10]^2$. In general, BFS-CV has accuracy values consistently above 0.9 and outperforms the other methods.

Figure 3.8 shows results on test cost performance of each method as a function of b, for large instances ($n \geq 1500$), where we fixed $h = 1$, corresponding to newsvendor ratios $b/(b+h) \geq 0.5$. ERM-ℓ_0 (CV) and ERM-ℓ_1 (CV) generally perform worse than BFS-CV,

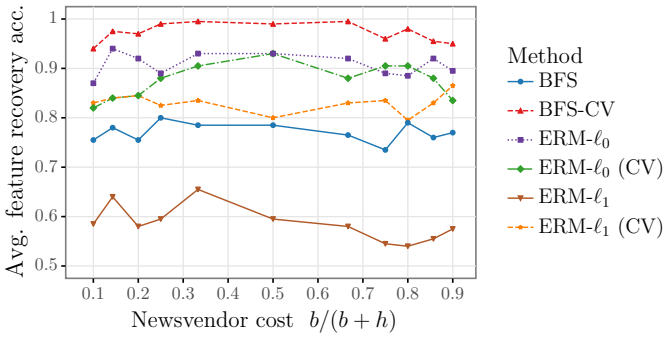


Figure 3.7: Impact of shortage cost b and holding cost h on the accuracy of feature recovery

since the deviation values are often positive. For all considered methods, we observe that the test cost deviations range from -6% to 10% and the variance increases with increasing b. We observed similar results for cases with $b/(b+h) < 0.5$.

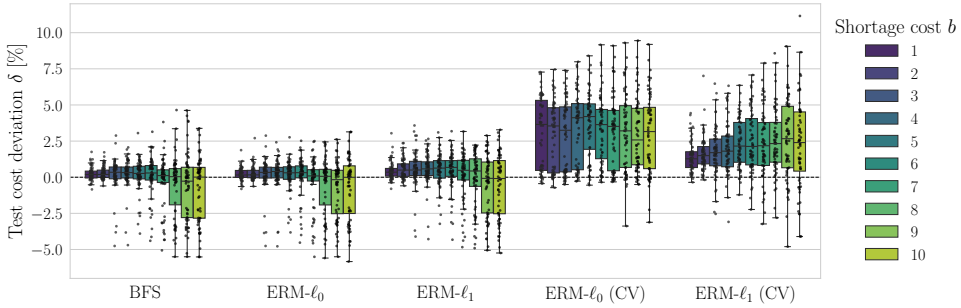


Figure 3.8: Impact of shortage cost b on the test cost performance of different methods, relative to BFS-CV

3.5.2 Nonlinear demand model

Demand may not be linearly related to the features. Therefore, we investigate how each method performs under nonlinear demand models, considering homoscedastic and heteroscedastic settings. In the following, we only present results regarding accuracy performance. Results on test cost performance did not provide new insights, since we observed similar patterns as in the case of linear instances (see Appendix 3.C). Unless otherwise stated, we use the same reference configuration as in the previous section.

Instance size. Due to the nonlinear structure of the demand models, Figure 3.9 shows considerably lower accuracy values compared to instances with linear demand. For heteroscedastic instances, BFS-CV outperforms existing methods, with accuracy values above 0.9 already for $n = 500$ samples. For the homoscedastic case, all methods present inferior accuracy compared to the heteroscedastic case. In this setting, BFS-CV and ERM-ℓ_0 present comparable performance, superior to the other considered methods.

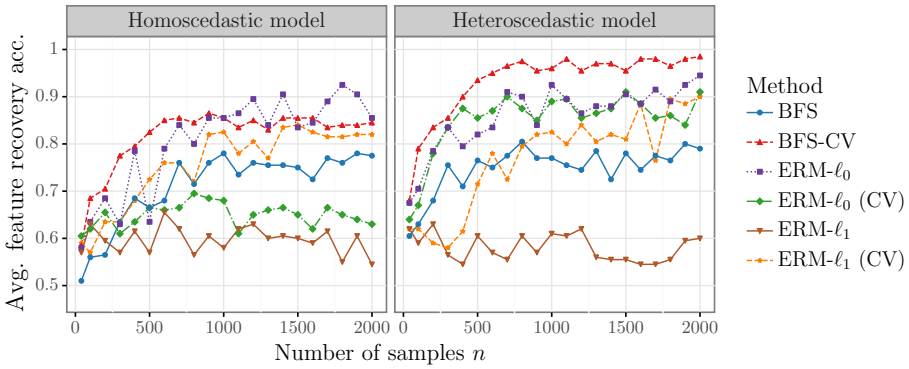


Figure 3.9: Impact of instance size on the accuracy of feature recovery

Number of features. Figure 3.10 shows the average feature recovery accuracy for varying $m \in \{8, 10, 12, 14\}$, for both homoscedastic and heteroscedastic demand. Similarly as for instances with linear demand, the number of features does not strongly influence the accuracy performance for nonlinear instances.

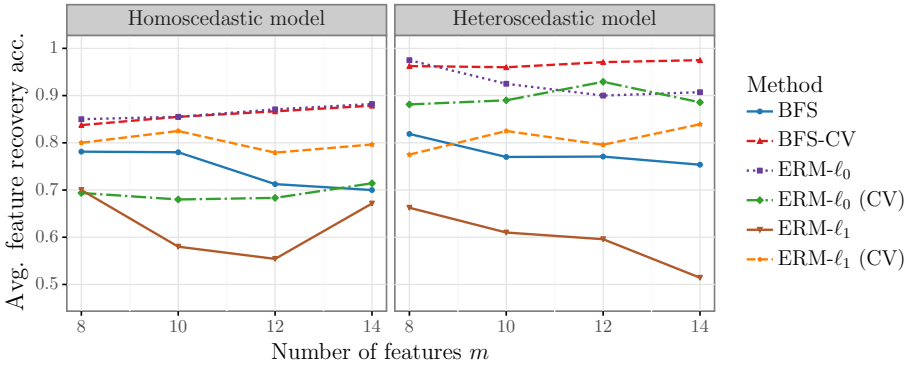


Figure 3.10: Impact of number of features m on the accuracy of feature recovery

Noise level. For the heteroscedastic case, BFS-CV outperforms existing methods (Figure 3.11). In this setting, varying the standard deviation σ_ϵ does not strongly affect the accuracy performance. For homoscedastic instances, all methods present inferior accuracy compared to heteroscedastic instances. Here, ERM-ℓ_0 is superior to other methods for most values of σ_ϵ, but BFS-CV often presents comparable accuracy.

Shortage and holding costs. In Figure 3.12, for heteroscedastic instances, BFS-CV has superior accuracy than other methods when $b/(b+h) \geq 0.3$. We observe that all methods perform poorly when the newsvendor ratio is below 0.3. In particular, the considered methods often do not recover any features when $b/(b+h) \in [0.1, 0.2]$, leading to accuracy values between 0.5 and 0.65. For the homoscedastic case, the accuracy of all methods decreases with increasing newsvendor ratios, with values often below 0.8. In this setting, BFS-CV is superior to most methods, but is outperformed by ERM-ℓ_0 when $b/(b+h) \geq 0.8$

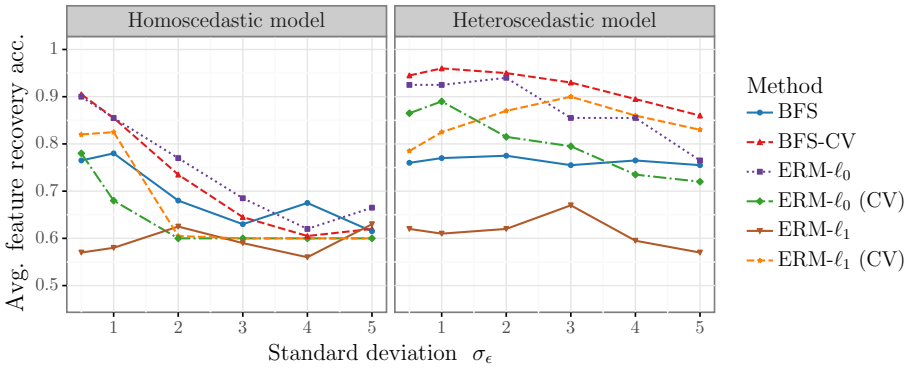


Figure 3.11: Impact of different noise levels on the accuracy of feature recovery

and by ERM-ℓ_1 (CV) when $b/(b+h) \in [0.2, 0.35]$.

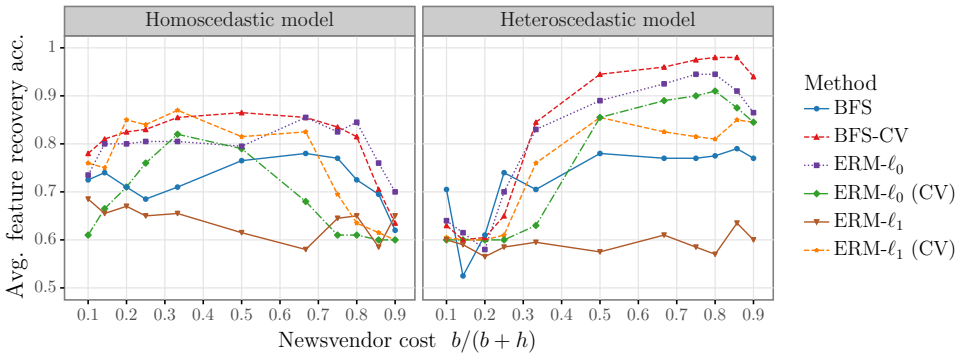


Figure 3.12: Impact of shortage cost b and holding cost h on the accuracy of feature recovery

One comment on these results is in order. Applying a linear decision function to nonlinear data may naturally lead to poor results, due to the inconsistent dependency structure with respect to the features. In some cases, we observed that all methods fail to achieve reasonable accuracy in feature recovery. However, for the vast majority of settings that we considered, BFS-CV presented good performance, outperforming the other methods. One possibility for dealing with such nonlinear instances would be to include additional features by considering nonlinear transformations of the original features, which we leave for future research.

3.6 Conclusion

We have presented a novel formulation based on bilevel optimization for incorporating feature selection in the feature-based newsvendor problem. The proposed BFS models provide an intuitive approach specifically designed for the task of feature selection, in which the upper-level problem directly selects the subset of relevant features. Our experimental results on synthetic data show that the proposed methods can accurately

recover ground-truth informative features, leading to more explainable inventory decisions in comparison to previous methods.

There are many possibilities for follow-up works. First, research on tailored solution methods for BFS and BFS-CV, e.g., based on decomposition strategies for mixed integer programming, may allow to improve the scalability of the proposed methods when dealing with a large number of features. Second, tailoring heuristic methods to efficiently solve very large instances remains an interesting avenue for future research. In this context, our proposed solution methods provide useful benchmarks for evaluating the performance of heuristics with respect to solution quality. Third, other classes of data-driven optimization problems may benefit from an extension of the proposed methodology. As an example, the newsvendor problem also captures the fundamental trade-offs emerging in decisions related to capacity planning. Accordingly, an extension of the proposed BFS models may be applied to select features in stochastic, data-driven variations of such problems. Finally, from a broader perspective, incorporating concepts from machine learning into data-driven optimization problems may lead to crucial advances for both fields. As exemplified in this work, feature selection is one among possibly many machine learning tasks that can be integrated into the decision-making process, especially given the growing need for more explainable decisions.

Acknowledgments

The authors are grateful for the support received through the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the research group Advanced Optimization in a Networked Economy (AdONE, GRK2201/277991500).

Appendix 3.A Computation times

In this section, we report the computation times of the different methods considered in the experiments. Figures 3.13, 3.14, 3.15, and 3.16 show how the solution times scale with the number of samples n, respectively for $m \in \{8, 10, 12, 14\}$ features. Here, we consider instances with linear demand model and we fix shortage cost to $b = 2$, the holding cost to $h = 1$, and the noise level to $\sigma_\epsilon = 1$.

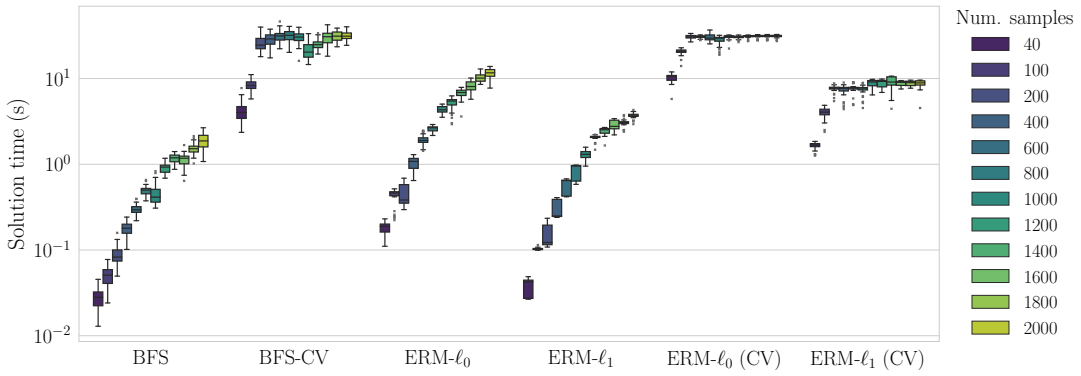


Figure 3.13: Solution times (in seconds) for instances with $m = 8$ features

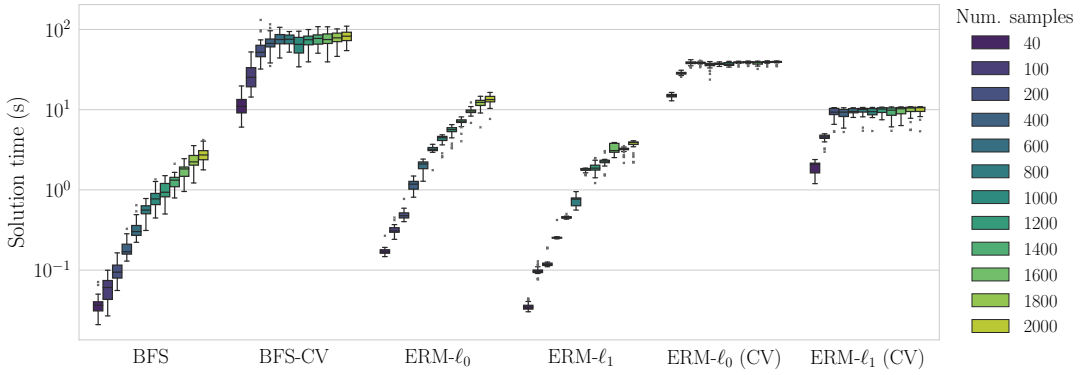


Figure 3.14: Solution times (in seconds) for instances with $m = 10$ features

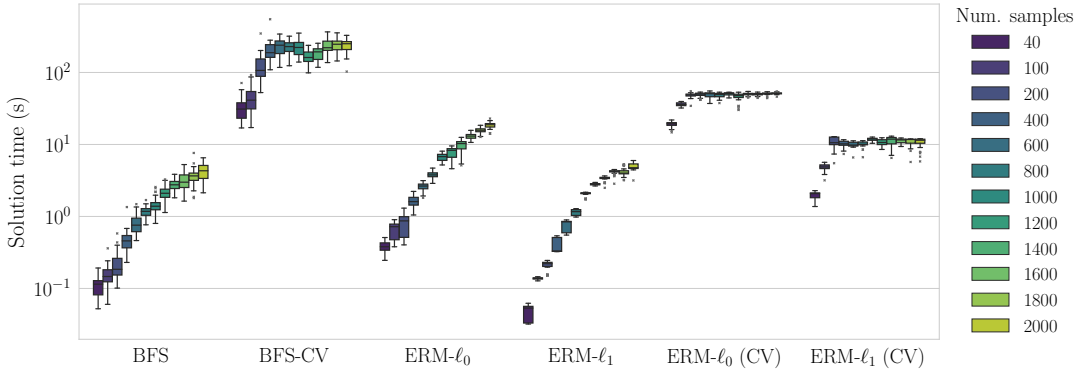


Figure 3.15: Solution times (in seconds) for instances with $m = 12$ features

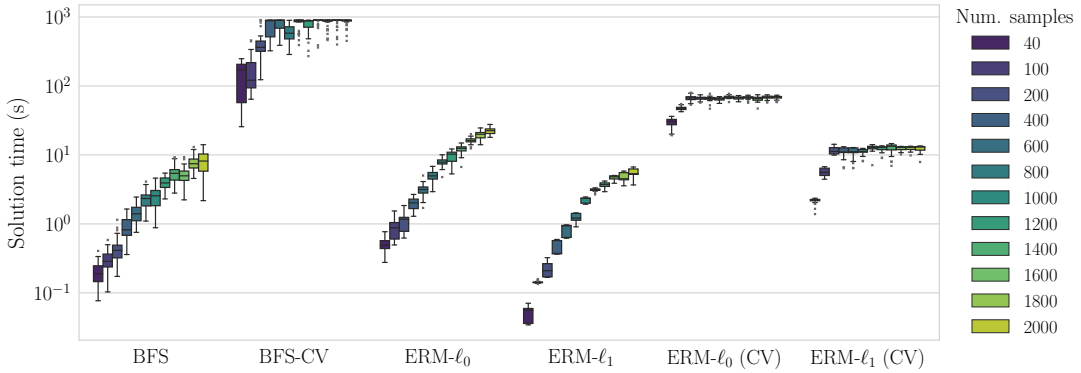


Figure 3.16: Solution times (in seconds) for instances with $m = 14$ features

Considering instances with a larger number of features ($m = 14$), BFS-CV cannot solve all instances to optimality within the specified time limit. Accordingly, Figure 3.17 illustrates the distribution of the optimality gaps (in percentage values) for instances with $m = 14$ features. In this case, optimality gaps are often below 5%.

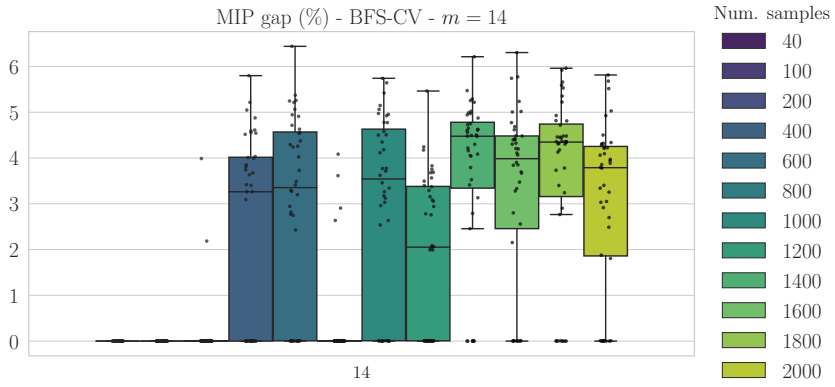


Figure 3.17: MIP gaps for $m = 14$

Appendix 3.B Complementary results

In the following, we report the p-values of the non-parametric one-sided Wilcoxon signed-rank test, with null hypothesis that the median difference in accuracy between the BFS-CV and other methods is negative. Tables 3.2 and 3.3 report p-values for instances with linear and nonlinear demand, respectively.

Figure	BFS	ERM-ℓ_0	ERM-ℓ_0 (CV)	ERM-ℓ_1	ERM-ℓ_1 (CV)
Figure 3.1	2.31E-35	8.73E-12	6.88E-23	2.25E-38	1.23E-32
Figure 3.3	2.53E-14	7.76E-04	4.14E-10	2.32E-14	1.95E-13
Figure 3.5	3.45E-07	**8.89E-02**	1.08E-11	1.06E-15	4.47E-13
Figure 3.7	2.00E-32	6.56E-08	3.96E-23	1.77E-37	5.02E-27

Table 3.2: P-values for linear instances.

Figure	Noise	BFS	ERM-ℓ_0	ERM-ℓ_0 (CV)	ERM-ℓ_1	ERM-ℓ_1 (CV)
Figure 3.9	Homoscedastic	8.20E-33	**3.15E-01**	1.77E-53	4.49E-55	2.81E-19
	Heteroscedastic	8.92E-56	1.92E-12	1.02E-29	7.49E-65	9.00E-49
Figure 3.10	Homoscedastic	4.45E-08	**7.87E-01**	2.66E-13	2.01E-12	2.37E-05
	Heteroscedastic	6.90E-13	3.32E-02	8.34E-08	9.35E-15	2.25E-11
Figure 3.11	Homoscedastic	2.47E-02	**9.86E-01**	5.60E-11	7.19E-10	6.27E-07
	Heteroscedastic	3.05E-16	1.72E-03	9.18E-15	5.51E-20	6.52E-08
Figure 3.12	Homoscedastic	2.58E-13	**3.98E-01**	6.28E-25	2.51E-22	2.21E-09
	Heteroscedastic	4.86E-13	**6.03E-02**	4.44E-17	1.72E-26	1.59E-16

Table 3.3: P-values for nonlinear instances.

Appendix 3.C Test cost results for nonlinear instances

We present results regarding test cost performance for instances with nonlinear demand model. This section follows the same structure as Section 3.5.2 of the main paper.

3.C.1 Instance size

Regarding the impact of instance size n, Figures 3.18 and 3.19 show test cost results for instances with heteroscedastic and homoscedastic demand, respectively.

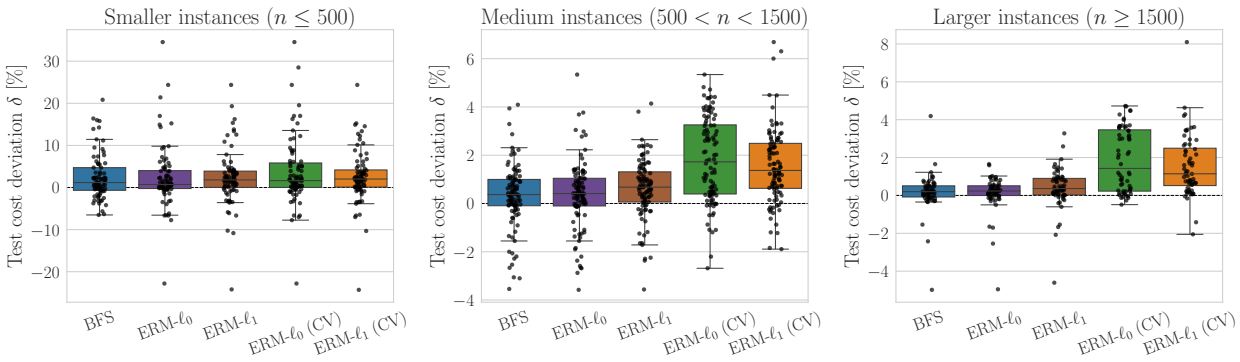


Figure 3.18: Impact of instance size on the test cost performance (heteroscedastic demand with Gaussian noise)

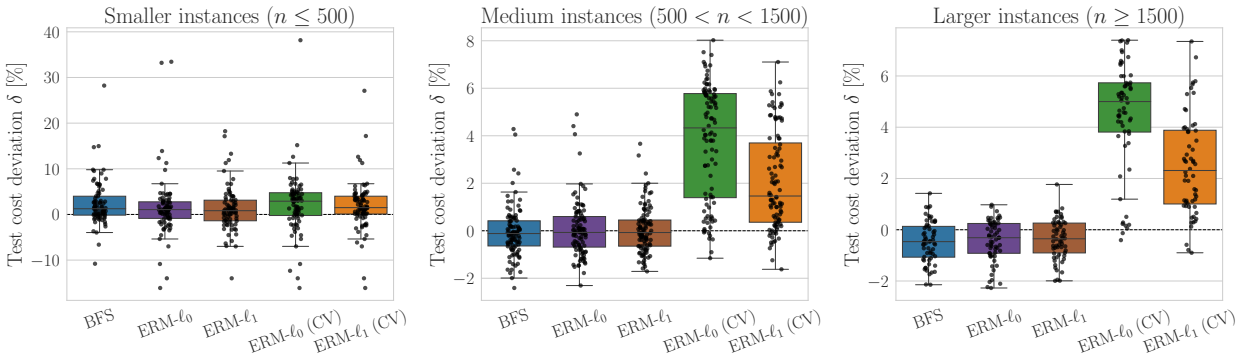


Figure 3.19: Impact of instance size on test cost performance (homoscedastic demand with Gaussian noise)

3.C.2 Number of features

Figures 3.20 and 3.21 show the impact of the number of features m on test cost results for large instances ($n \geq 1500$) with heteroscedastic and homoscedastic demand, respectively.

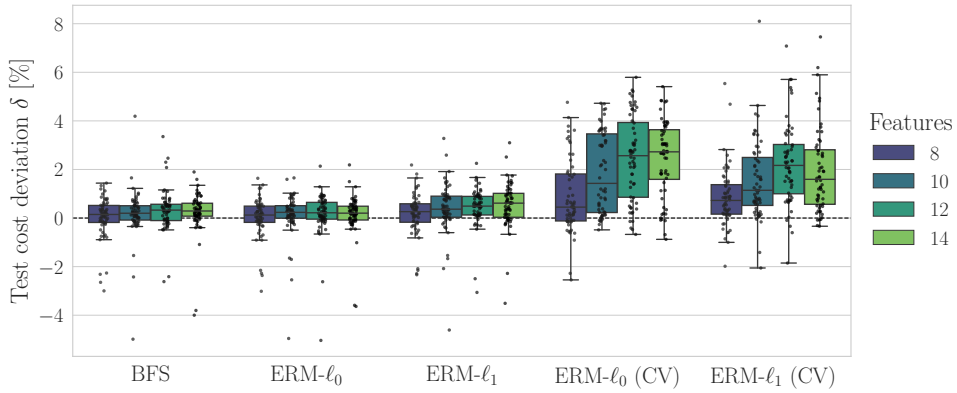


Figure 3.20: Impact of number of features m on the test cost (heteroscedastic demand with Gaussian noise)

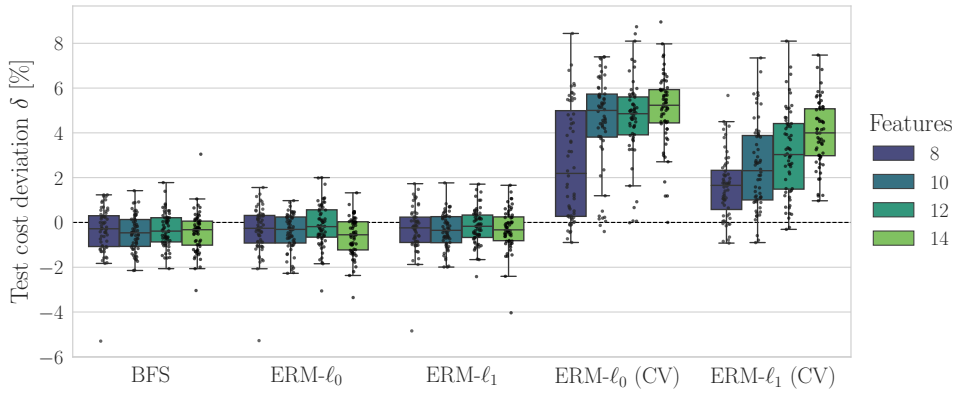


Figure 3.21: Impact of number of features m on test cost (homoscedastic demand with Gaussian noise)

3.C.3 Noise level

Figures 3.22 and 3.23 show the impact of noise level σ_ϵ on test cost results for large instances ($n \geq 1500$) with heteroscedastic and homoscedastic demand, respectively.

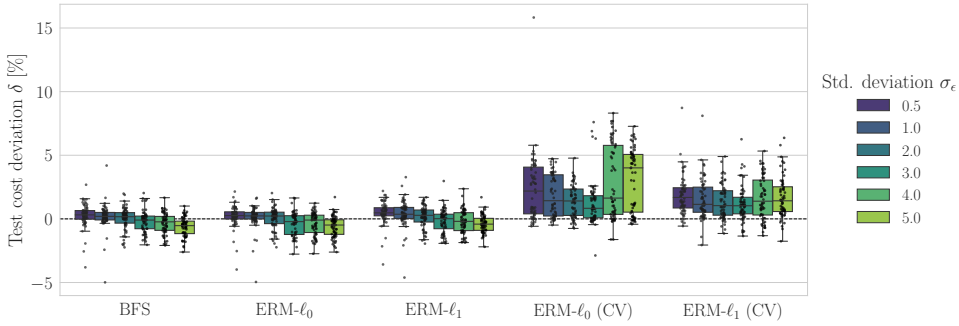


Figure 3.22: Impact of noise level σ_ϵ on test cost (heteroscedastic demand with Gaussian noise)

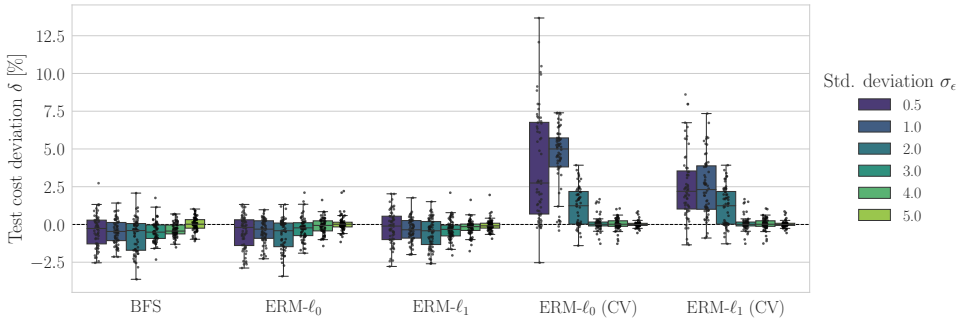


Figure 3.23: Impact of noise level σ_ϵ on test cost (homoscedastic demand with Gaussian noise)

3.C.4 Shortage cost

Figures 3.24 and 3.25 show the impact of shortage cost b on test cost results for large instances ($n \geq 1500$) with heteroscedastic and homoscedastic demand, respectively.

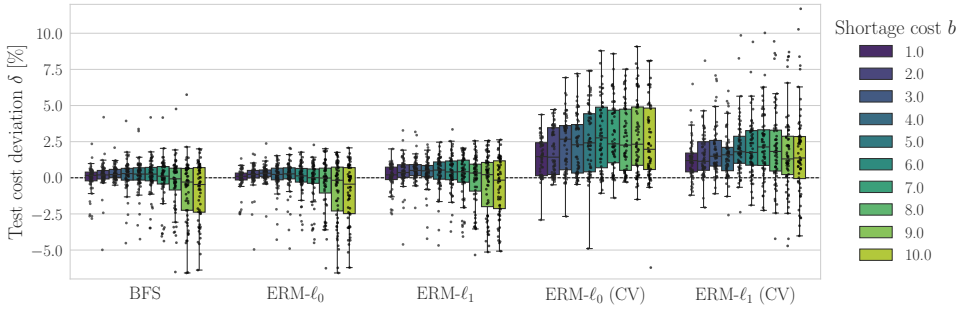


Figure 3.24: Impact of shortage cost b on the test cost (heteroscedastic demand with Gaussian noise)

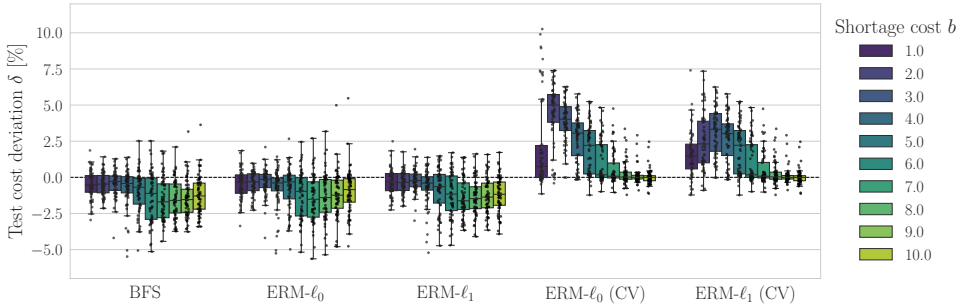


Figure 3.25: Impact of shortage cost b on test cost (homoscedastic demand with Gaussian noise)

3.C.5 Holding cost

Figures 3.26 and 3.27 show the impact of holding cost h on test cost results for large instances ($n \geq 1500$) with heteroscedastic and homoscedastic demand, respectively.

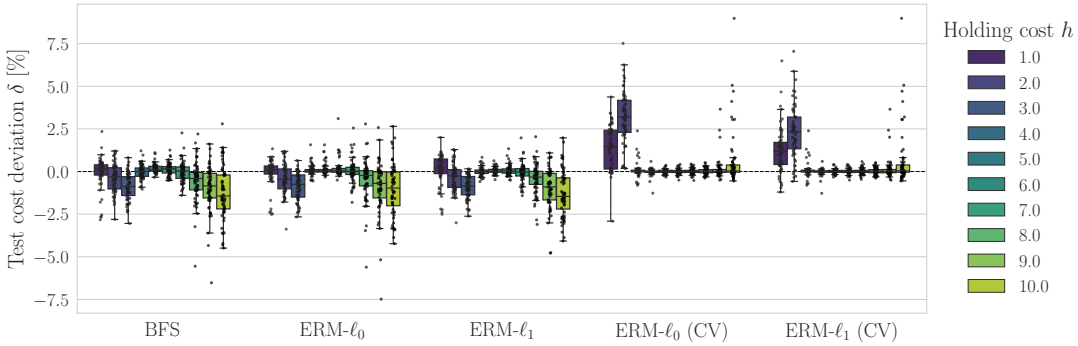


Figure 3.26: Impact of holding cost h on test cost (heteroscedastic demand with Gaussian noise)

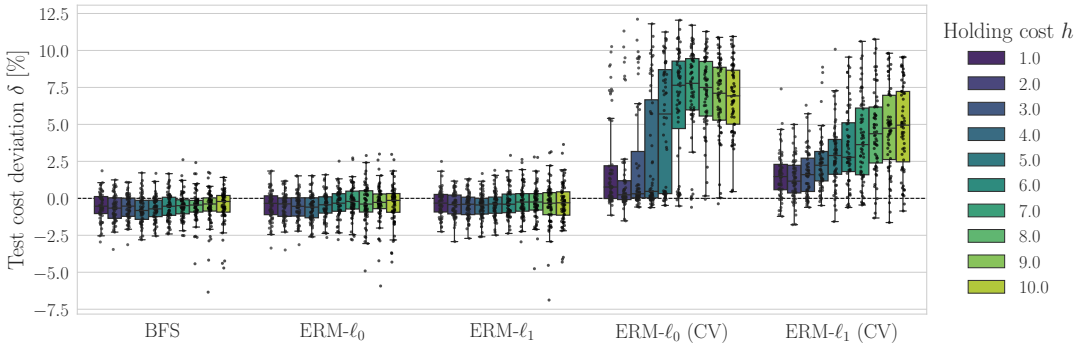


Figure 3.27: Impact of holding cost h on test cost (homoscedastic demand with Gaussian noise)

4 Contextual Stochastic Vehicle Routing with Time Windows

Abstract

We study the vehicle routing problem with time windows (VRPTW) and stochastic travel times, in which the decision-maker observes related contextual information, represented as feature variables, before making routing decisions. Despite the extensive literature on stochastic VRPs, the integration of feature variables has received limited attention in this context. We introduce the contextual stochastic VRPTW, which minimizes the total transportation cost and expected late arrival penalties conditioned on the observed features. Since the joint distribution of travel times and features is unknown, we present novel data-driven prescriptive models that use historical data to approximate the conditional expectation in the objective function. We distinguish the prescriptive models between point-based approximation, sample average approximation (SAA), and penalty-based approximation, each taking a different perspective on dealing with stochastic travel times and features. We develop specialized branch-price-and-cut algorithms to solve these data-driven prescriptive models. Our computational experiments compare the out-of-sample cost performance of different methods on instances with up to one hundred customers. Our results show that, surprisingly, a feature-dependent SAA outperforms existing and novel methods in most settings. The proposed method provides up to a 13.2% reduction in test cost compared to the classical SAA baseline and up to a 26.1% reduction compared to a point-based approximation method following the traditional predict-then-optimize approach.

This chapter is based on an article available as:

Serrano B., Florio A. M., Minner S., Schiffer M., Vidal T. (2024). Contextual Stochastic Vehicle Routing with Time Windows. `https://doi.org/10.48550/arXiv.2402.06968` (Available as pre-print).

4.1 Introduction

The vehicle routing problem (VRP) is one of the most studied problems in operations research, with the majority of papers focusing on deterministic variants, in which the decision-maker has complete information regarding the model parameters. In practice, VRPs have many sources of uncertainty, e.g., related to travel times, demands, or service times, which stimulated research interest in stochastic optimization models for VRPs. However, many studies on stochastic VRPs consider stylized problems and assume that the uncertain parameters of the optimization problem have known probability distributions.

With the growing availability of data, decision-makers can harness historical data on the uncertain parameters in addition to correlated contextual information, represented as feature variables, to improve uncertainty representation in VRP models. The benefits of contextual optimization for decision-making problems under uncertainty are evident in works from many fields (see, e.g., Sadana et al. 2024). However, within the VRP literature, integrating feature variables related to uncertain parameters has received limited attention. We address this research gap and directly incorporate contextual information into the classical VRP framework.

Specifically, we study the vehicle routing problem with time windows (VRPTW) and uncertain travel times, in which the decision-maker observes related features before making routing decisions. For example, the decision-maker may use contextual information about road closures, seasonal events, or the day of the week to improve routing decisions. To capture the dependence of the optimization problem on external features, we introduce the *conditional stochastic* VRPTW, which minimizes the total transportation cost and expected late arrival penalties conditioned on a set of observed features. We present prescriptive models that use historical data to provide an approximate solution to the problem. We distinguish the data-driven prescriptive models between point-based approximation, sample average approximation (SAA), and penalty-based approximation, each taking a different perspective in dealing with uncertain travel times and features. In particular, point-based approximation models correspond to the surrogate problem in which travel times are fixed to some estimated values given, e.g., by using a predictive model. In contrast, SAA models capture travel time variability by considering a set of travel time scenarios. In this setting, the decision-maker can take the observed features into account either by associating a feature-dependent weight function to each training observation or by directly constructing feature-dependent travel time scenarios. Finally, we present a penalty-based approximation model, which consists of a predictive model that is trained to directly predict late arrival penalties from features. We develop specialized branch-price-and-cut (BP&C) algorithms and compare the prescriptive models in computational experiments.

4.1.1 Related Work

We identified two main research streams in the related literature. Specifically, we start with a review of related works on stochastic VRPs and then discuss data-driven stochastic optimization problems that consider the presence of features for different problem settings beyond VRPs.

Stochastic VRPs. Numerous works have investigated stochastic variants of the VRP, e.g., with stochastic demands, travel times, or service times, among other sources of uncertainty. The three most common modeling approaches from the literature are based on robust optimization (RO), chance-constrained programming (CCP), and two-stage stochastic programming with recourse (SPR). Similarly to our paper, many SPR formulations assume a simple recourse policy consisting of a penalty, e.g., for late or early arrivals when time windows are considered. For an introduction to stochastic VRPs, we refer the reader to Gendreau, Jabali, and Rei (2014) and Oyola, Arntzen, and Woodruff (2017, 2018).

The seminal work of Laporte, Louveaux, and Mercure (1992) introduced an uncapacited VRP with stochastic service and travel times. Kenyon and Morton (2003) extended the work of Laporte, Louveaux, and Mercure (1992) by focusing on minimizing the completion time instead of the total transportation cost. Li, Tian, and Leung (2010) considered time window constraints, extending the formulations of Laporte, Louveaux, and Mercure (1992) to the stochastic VRPTW, for which they proposed a heuristic based on tabu search. Taş et al. (2014a) later solved a VRPTW under independently gamma-distributed travel times, in which early and late service are permitted but penalized. Their objective minimized the sum of transportation costs and the expected early and late arrival penalties. In contrast to Taş et al. (2014a), we make no assumptions about the distribution of travel times. In addition, we assume that early service is forbidden, which increases the problem difficulty because the distributions of service start times at customers are truncated (cf. Zhang et al. 2019).

From an RO perspective, Lee, Lee, and Park (2012) and Adulyasak and Jaillet (2016) studied the stochastic VRP with deadlines, a special case of the VRPTW. Further works by Miranda and Conceição (2016) and Errico et al. (2018) adopted a CCP approach for the VRPTW with stochastic travel and service times, respectively. Zhang et al. (2021) addressed the VRPTW with stochastic travel times and proposed a data-driven distributionally robust optimization model.

Focusing on the capacitated vehicle routing problem (CVRP) under demand uncertainty, Gounaris, Wiesemann, and Floudas (2013) proposed an RO formulation, and Ghosal and Wiesemann (2020) and Ghosal, Ho, and Wiesemann (2024) proposed distribu-

tionally robust chance-constrained optimization models. Recently, Rostami et al. (2021) considered a CVRP in which travel times are stochastic and statistically correlated. Their bi-objective model minimized the total expected travel time and total variance, assuming that the mean and covariance matrix of the travel times are known.

In terms of different optimization criteria, Jaillet, Qi, and Sim (2016) proposed the *requirements violation* (RV) index, which considers both the probability and magnitude of time window violations. They model a traveling salesman problem with time windows (TSPTW) where early service is permitted, and the total RV measure is minimized. Inspired by the RV index, Zhang et al. (2019) proposed the *essential riskiness* index, which enables a multi-commodity flow formulation of the TSPTW where early service is forbidden.

A number of works implicitly considered the impact of external features by modeling time-dependent travel times. For a review on time-dependent routing problems, we refer the reader to Gendreau, Ghiani, and Guerriero (2015). Dabia et al. (2013) proposed an arc-based formulation for the VRPTW under deterministic time-dependent travel times and Taş et al. (2014b) proposed metaheuristics for the VRPTW with time-dependent and stochastic gamma-distributed travel times. In contrast to time-dependent VRPs, we assume that travel times are time-independent within a service period, and we explicitly model the dependence of travel times on a set of external features.

Despite the extensive literature on stochastic VRPs, previous works considered the presence of contextual information only implicitly, e.g., with time-dependent travel times. Incorporating contextual information into the model can significantly reduce routing costs. Accordingly, we contribute to the current state of the art by proposing novel models and algorithms for the conditional stochastic VRPTW. In addition, we show how to adapt state-of-the-art BP&C techniques when contextual information is available.

Contextual stochastic optimization. Many recent works have investigated the use of contextual information to improve decisions under uncertainty, with applications to transportation, inventory, and operations management, among others. The recent survey by Sadana et al. (2024) reviewed papers on contextual optimization with a focus on models and methods, while the review paper by Mišić and Perakis (2020) discussed recent applications in the operations management literature.

Bertsimas and Kallus (2019) proposed a framework for conditional stochastic optimization based on weighted SAA and derived weight functions based on machine learning methods, such as k-nearest neighbors regression, local linear regression, and tree-based methods. The authors applied the framework to a two-stage shipment planning problem and to a multi-product inventory problem. Nearly simultaneously, Bertsimas and McCord (2018) adapted the framework for problems with decision-dependent uncertain-

ties. Bertsimas and McCord (2019) and Bertsimas, McCord, and Sturt (2023) extended the framework of Bertsimas and Kallus (2019) to multi-stage and multi-period stochastic optimization problems, respectively. Van Parys and Bennouna (2022) proposed robust formulations based on regularized SAA for conditional two-stage optimization problems.

Elmachtoub and Grigas (2021) proposed a framework tailored to optimization problems consisting of a linear objective with uncertain coefficients. They used feature data to learn a prediction model for the objective coefficients and adopted a modified loss function that directly leverages the structure of the optimization problem instead of minimizing a standard prediction error, such as the least squares loss. Mandi et al. (2020) extended this framework to solve some discrete combinatorial problems and illustrated their algorithmic improvements on a weighted knapsack problem and a scheduling problem.

In the context of inventory optimization, Beutel and Minner (2012) and Ban and Rudin (2019) studied the data-driven newsvendor problem and proposed models for learning a decision function that predicts order quantities directly from feature observations. Serrano et al. (2024) extended their framework, introducing an integrated approach based on bilevel programming that performs feature selection. Mandl and Minner (2023) studied a multi-period commodity procurement problem under price uncertainty with forward and spot purchase options. They proposed a data-driven model to derive optimal purchase policies based on features, such as economic indicators.

In the context of last-mile delivery operations, Liu, He, and Shen (2021) assigned orders to drivers under uncertain service times and considered that drivers' routes may deviate from planned routes. The authors used features, e.g., based on the customers' locations, to learn a prediction model for the drivers' travel times, which is then integrated into the order assignment optimization.

Motivated by the problem of forecasting electricity demands given a set of features, Rios, Wets, and Woodruff (2015) proposed a scenario generation method based on estimating the distribution of forecast errors, i.e., residuals. Ban, Gallien, and Mersereau (2019) investigated a similar methodology for a multi-stage stochastic procurement problem under uncertain demand, where they learned a regression model that relates features to demands and used the residuals to generate demand samples for a new product. They subsequently solved an SAA model to determine the optimal procurement policy.

Many works in the contextual optimization literature deal with optimization problems that can be efficiently solved, e.g., that can be modeled as linear programs. In contrast, the VRPTW is a challenging optimization problem, for which many researchers have proposed problem-tailored solution techniques over the last decades. On the one hand, we show how existing contextual optimization methods can be applied to this challenging application. On the other hand, we investigate novel methods that can be generally

applied to other contextual optimization problems.

4.1.2 Contributions

Our work contributes to the research streams outlined in the previous section. First, we introduce a novel formulation for the conditional stochastic VRPTW, in which realizations of travel times are conditioned on a set of related features. Second, we show how existing contextual optimization methods can be applied to the conditional stochastic VRPTW and propose novel data-driven methods leveraging historical data. In particular, we propose a conditional sample average approximation (CSAA) method, a general technique based on SAA for solving conditional stochastic optimization problems. Third, we propose a problem-tailored penalty-based prescriptive method that uses machine learning to predict late arrival penalties at customers instead of relying on travel time estimates. Fourth, we describe a specialized BP&C algorithm that is generally applicable to stochastic VRP variants in which the travel times can capture the problem uncertainty, e.g., when uncertain service times or costs can be modeled in terms of uncertain travel times. One key component of our BP&C implementation is the use of completion bounds instead of common dominance rules for discarding labels in the pricing algorithm. Fifth, we conduct computational experiments to compare the proposed prescriptive models and conclude that SAA methods equipped with feature-dependent scenarios show the best average performance, providing solutions that are closest to the full-information benchmark.

The remainder of this paper is organized as follows. Section 4.2 formally introduces the conditional stochastic VRPTW. Section 4.3 presents our data-driven prescriptive models. Section 4.4 describes the corresponding solution methods based on BP&C. Section 4.5 presents our experimental design and Section 4.6 discusses computational results. Finally, Section 4.7 concludes the paper.

4.2 The Conditional Stochastic VRPTW

We define the conditional stochastic VRPTW on a complete digraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, \ldots, N\}$ is the set of nodes and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}\}$ is the set of arcs. Set $\mathcal{V} \setminus \{0\}$ represents customers, and node 0 denotes the depot. A homogeneous fleet of K vehicles, each with a capacity of Q, is stationed at the depot. Each node $i \in \mathcal{V}$ has known demand q_i and a time window $[e_i, \ell_i]$, $0 \leq e_i \leq \ell_i$. Each arc $(i, j) \in \mathcal{A}$ has a deterministic cost c_{ij} and an uncertain travel time $\tilde{t}_{ij}$. Throughout the paper, we use a tilde to indicate random variables and follow standard boldfaced notation for vectors and vector-valued functions.

We focus on a setting in which the joint distribution of travel times $\tilde{\mathbf{t}} = [\tilde{t}_{ij}]_{(i,j)\in\mathcal{A}}$ is unknown. The decision-maker observes a set of features, represented by a p-dimensional vector $\mathbf{x} = [x_1, \ldots, x_p]$, before the travel times are revealed. Features are exogenous variables that may be travel time predictors, e.g., day of the week, season, events, meteorological data, and road works or closures.

A route is a non-empty sequence of customers $\theta = (v_1, \ldots, v_L)$ such that $\sum_{i\in\theta} q_i \leq Q$ and $v_i \neq v_j$ if $i \neq j$. We further define:

$$C_\theta = c_{0v_1} + \sum_{k=2}^{L} c_{v_{k-1}v_k} + c_{v_L 0} \tag{4.1}$$

as the transportation cost of route θ. We assume that transportation costs are deterministic and independent of travel times and features, e.g., if drivers have fixed salaries, and if fuel consumption costs can be well approximated considering only the distance traveled. Note that this assumption is not restrictive since all proposed methods can be adapted to a setting with uncertain costs by using (simplified variants of) the methodology presented in this work.

All routes start at the depot. We assume that service times equal to 0, without loss of generality, as a positive service time at a customer can be added to the travel times of its outgoing arcs. Early service is forbidden, such that in case of early arrival at customer i, the vehicle must wait until e_i before service can start. Given a route $\theta = (v_1, \ldots, v_L)$ and realized travel times $\mathbf{t} = [t_{ij}]_{(i,j)\in\mathcal{A}}$, we define the *arrival time* $a_\theta(v_k; \mathbf{t})$ at customer $v_k \in \theta$ as:

$$a_\theta(v_k; \mathbf{t}) = \begin{cases} t_{0v_1}, & \text{if } k = 1, \\ \max\{e_{v_{k-1}}, a_\theta(v_{k-1}; \mathbf{t})\} + t_{v_{k-1}v_k}, & \text{if } k \in \{2, \ldots, L\}. \end{cases} \tag{4.2}$$

Since early arrivals are not permitted, the instant when customer $v_k \in \theta$ is served, i.e., the *service start time* at customer v_k, is given by:

$$s_\theta(v_k; \mathbf{t}) = \max\{e_{v_k}, a_\theta(v_k; \mathbf{t})\}. \tag{4.3}$$

Equivalently, $s_\theta(v_k; \mathbf{t}) + t_{v_k v_{k+1}} = a_\theta(v_{k+1}; \mathbf{t})$. Moreover, a penalty of $\pi(a_\theta(v_k; \mathbf{t}) - \ell_{v_k})$ occurs in case of late arrivals at customer v_k, where $\pi(\cdot)$ is a nondecreasing penalty function such that $\pi(0) = 0$. For notational convenience, we define $\pi(u) = 0$ if $u < 0$.

With Θ being the set of all routes, we represent a solution to the conditional stochastic VRPTW by a vector $\mathbf{z} = [z_\theta]_{\theta\in\Theta} \in \{0, 1\}^{|\Theta|}$, where $z_\theta = 1$ if and only if route θ belongs

to the solution. Accordingly, the feasible region is determined by the set:

$$\mathcal{Z}_\Theta = \left\{ \mathbf{z} \in \{0,1\}^{|\Theta|} \;\middle|\; \begin{array}{ll} \displaystyle\sum_{\theta\in\Theta} \mathbb{1}(i \in \theta) z_\theta = 1, \quad i \in \mathcal{V} \setminus \{0\}, & \text{(a)} \\ \displaystyle\sum_{\theta\in\Theta} z_\theta \leq K & \text{(b)} \end{array} \right\}, \tag{4.4}$$

where $\mathbb{1}(\cdot)$ is the indicator function. The partitioning constraints (4.4a) ensure that all customers are visited once, and constraint (4.4b) enforces that no more than K vehicles are used.

We formulate the conditional stochastic VRPTW as a two-stage stochastic program:

$$\mathbf{z}^*(\mathbf{x}) = \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} \; C(\mathbf{z}) + \mathbb{E}\left[Q(\mathbf{z}, \tilde{\mathbf{t}}) \,\middle|\, \tilde{\mathbf{x}} = \mathbf{x} \right], \tag{CS-VRPTW}$$

where the first-stage objective corresponds to the total transportation costs $C(\mathbf{z}) = \sum_{\theta\in\Theta} C_\theta z_\theta$, and the second-stage recourse policy consists of penalizing late arrivals, leading to the second-stage value function:

$$Q(\mathbf{z}, \mathbf{t}) = \sum_{\theta\in\Theta} \Big(\sum_{i\in\theta} \pi(a_\theta(i;\mathbf{t}) - \ell_i) \Big) z_\theta. \tag{4.5}$$

We further denote by $f_\Theta(\mathbf{z}, \mathbf{t})$ the value of a solution $\mathbf{z}$ under realized travel times $\mathbf{t}$:

$$f_\Theta(\mathbf{z}, \mathbf{t}) = C(\mathbf{z}) + Q(\mathbf{z}, \mathbf{t}). \tag{4.6}$$

Thus, we can equivalently express the CS-VRPTW as:

$$\mathbf{z}^*(\mathbf{x}) = \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} \; \mathbb{E}\left[f_\Theta(\mathbf{z}, \tilde{\mathbf{t}}) \,\middle|\, \tilde{\mathbf{x}} = \mathbf{x} \right]. \tag{4.7}$$

Since the joint distribution of travel times and features is unknown, we approximate the conditional expectation in the CS-VRPTW formulation using data-driven prescriptive models that leverage historical travel times and features.

4.3 Data-driven Prescriptive Models

As the conditional stochastic VRPTW formulation cannot be solved directly, we discuss approximate and computationally tractable reformulations in the following. We assume that the decision-maker has access to historical travel times for the last n periods, represented by the $n \times |\mathcal{A}|$ matrix $\mathbf{T} = [\mathbf{t}^1, \ldots, \mathbf{t}^n]^\top$, where $\mathbf{t}^k = [t_{ij}^k]_{(i,j)\in\mathcal{A}}$ are the travel times observed at period k. Further, the decision-maker has access to feature data for the last

n periods, represented by the $n \times p$ matrix $\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^n]^\top$, where $\mathbf{x}^k = [x_1^k, \ldots, x_p^k]$ contains the realizations of p features at period k. We assume w.l.o.g. that $x_1^k = 1$ for all $k \in \{1, \ldots, n\}$. We further assume that the data is low-dimensional, i.e., $p < n$, and that $\mathbf{X}$ has full rank. The decision-maker solves the CS-VRPTW for period $n+1$, having observed the feature vector $\mathbf{x}^{n+1}$ but without knowing the realization of travel times.

4.3.1 Point-based Approximation

We simplify the CS-VRPTW by fixing the travel times to point estimates $\hat{\mathbf{t}} = [\hat{t}_{ij}]_{(i,j)\in\mathcal{A}}$ and considering the deterministic optimization problem:

$$\hat{\mathbf{z}}_{\text{D}}^*(\mathbf{x}) \;=\; \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} \; f_\Theta\left(\mathbf{z}, \hat{\mathbf{t}}\right) \;=\; \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} \; C(\mathbf{z}) + Q(\mathbf{z}, \hat{\mathbf{t}}). \tag{P}$$

In this setting, different options to compute $\hat{\mathbf{t}}$ from historical travel times $\mathbf{T}$ and feature data $\mathbf{X}$ exist. If $\mathbf{X}$ is a poor predictor of $\mathbf{T}$, then a reasonable option is to set $\hat{t}_{ij}$ to a statistic of $\{t_{ij}^k\}_{k\in\{1,\ldots,n\}}$, e.g., the mean, median, or a higher percentile in case more protection against late arrivals is desired. Otherwise, if $\mathbf{X}$ is a good predictor of $\mathbf{T}$, we may fix travel times to:

$$\hat{\mathbf{t}} = \mathbf{g}(\mathbf{x}^{n+1}; \boldsymbol{\varphi}), \tag{4.8}$$

where $\mathbf{g} : \mathbb{R}^p \mapsto \mathbb{R}^{|\mathcal{A}|}$ is a travel time prediction model, e.g., a multivariate regression model or a neural network, with parameters $\boldsymbol{\varphi}$ trained from historical travel times and features. The latter approach corresponds to the *predict-then-optimize* (PTO) paradigm (Elmachtoub and Grigas, 2021). Since point estimates are used, Model (P) does not capture the impact of travel time variability on the decision cost and thus may yield poor decisions because the effects of under- and over-predicting travel times are not symmetric.

Note that a special case of Model (P) arises when the penalty function is such that $\pi(s) = \infty$ if $s > 0$. In this case, the model reduces to the classical VRPTW with hard time windows, for which efficient exact algorithms exist (cf. Pessoa et al. 2020). For general non-decreasing penalty functions, we solve Model (P) with a tailored BP&C algorithm as described in Section 4.4.

4.3.2 Sample Average Approximation

Travel time variability plays a key role in the CS-VRPTW since late arrival penalties grow with the amount of delay. We represent travel time variability in the stochastic model with a set of scenarios $\{\mathbf{t}^\omega\}_{\omega\in\Omega}$, where $\mathbf{t}^\omega = [t_{ij}^\omega]_{(i,j)\in\mathcal{A}}$ is a possible realization of travel times at period $n+1$, corresponding to scenario $\omega \in \Omega$. We further associate a feature-dependent weight function $\alpha^\omega(\mathbf{x}) : \mathbb{R}^p \mapsto \mathbb{R}$ to each scenario $\omega \in \Omega$ and approximate the

conditional expected penalty using weighted SAA:

$$\hat{\mathbf{z}}_{\mathrm{s}}^{*}(\mathbf{x}) \;=\; \arg\min_{\mathbf{z}\in\mathcal{Z}_{\Theta}} \sum_{\omega\in\Omega} \alpha^{\omega} f_{\Theta}\left(\mathbf{z}, \mathbf{t}^{\omega}\right) \;=\; \arg\min_{\mathbf{z}\in\mathcal{Z}_{\Theta}} \; C(\mathbf{z}) + \sum_{\omega\in\Omega} \alpha^{\omega} Q(\mathbf{z}, \mathbf{t}^{\omega}), \tag{S}$$

where we assume, w.l.o.g., that $\sum_{\omega\in\Omega} \alpha^{\omega} = 1$. Model (S) computes the second-stage cost by SAA (Shapiro, Dentcheva, and Ruszczyński, 2014). Note that Model (P) is a special case of Model (S) with $|\Omega| = 1$.

Before solving Model (S), the decision-maker must define the set of scenarios Ω. If the feature data $\mathbf{X}$ are poor predictors of the travel times $\mathbf{T}$, then a sensible option is to define the travel time scenarios as the historical travel times $\{\mathbf{t}^{\omega}\}_{\omega\in\Omega} = \{\mathbf{t}^{k}\}_{k\in\{1,\ldots,n\}}$, with $\alpha^{\omega} = 1/n$ for $\omega \in \Omega$.

4.3.3 Conditional SAA

To generate feature-dependent scenarios, we specify a probabilistic model for travel times and features. Let $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_p]$ be a random vector representing features before they are revealed at the current period $n+1$. Then, we assume that travel times (conditioned on $\tilde{\mathbf{x}}$) are given by:

$$\tilde{\mathbf{t}} \mid \tilde{\mathbf{x}} = \mathbf{B}^{\top}\tilde{\mathbf{x}} + \tilde{\boldsymbol{\varepsilon}}, \tag{4.9}$$

where $\mathbf{B}$ is a $p \times |\mathcal{A}|$ matrix of constant parameters and $\tilde{\boldsymbol{\varepsilon}} = [\tilde{\varepsilon}_{ij}]_{(i,j)\in\mathcal{A}}$ is a random noise vector following a multivariate distribution with zero mean and unknown covariance matrix $\boldsymbol{\Sigma}$.

The conditional travel times model from Equation (4.9) generalizes the correlated travel times model by Jaillet, Qi, and Sim (2016) in that $\tilde{x}_1, \ldots, \tilde{x}_p$ are not assumed to be independent and the noise terms $\tilde{\varepsilon}_{ij}$ allow for a stronger or weaker dependence between $\tilde{t}_{ij}$ and $\tilde{\mathbf{x}}$ for each arc. Therefore, the feature space may include interaction terms between travel time covariates, which may improve travel time prediction accuracy as non-linearities can be captured. Moreover, the error terms $\tilde{\varepsilon}_{ij}$ may be correlated, so the model allows for correlated travel time variability caused by unanticipated events, e.g., emergency work or traffic incidents.

From historical travel times $\mathbf{T}$ and feature data $\mathbf{X}$, we estimate matrix $\mathbf{B}$ by least squares:

$$\hat{\mathbf{B}} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{T}. \tag{4.10}$$

Following multivariate analysis theory (Rencher and Christensen, 2012), an estimate of

the covariance matrix $\boldsymbol{\Sigma}$ is given by:

$$\hat{\boldsymbol{\Sigma}} = \frac{\mathbf{T}^\top \mathbf{T} - \hat{\mathbf{B}}^\top \mathbf{X}^\top \mathbf{T}}{n - p}. \tag{4.11}$$

Accordingly, if the feature data are good travel time predictors, we obtain a feature-dependent representation of travel time uncertainty at period $n+1$ by sampling travel times from a multivariate distribution with mean and covariance given by $\hat{\mathbf{B}}^\top \mathbf{x}^{n+1}$ and $\hat{\boldsymbol{\Sigma}}$, respectively.

4.3.4 Residual-based SAA

The conditional SAA model from Section 4.3.3 requires the decision-maker to make distributional assumptions on travel times to generate feature-dependent scenarios. We present a distribution-free residual-based sample average approximation (RSAA) model based on the residual tree method of Ban, Gallien, and Mersereau (2019). Consider a travel time prediction model $\mathbf{g}(\cdot)$ as introduced in Equation (4.8) with parameters learned from historical travel times $\mathbf{T}$ and features $\mathbf{X}$. We compute the prediction error, which we denote as the *residual*, associated with training observation k as:

$$\boldsymbol{\epsilon}^k = \mathbf{g}(\mathbf{x}^k) - \mathbf{t}^k. \tag{4.12}$$

Given a feature vector $\mathbf{x}^{n+1}$ at period $n+1$, we generate a set of feature-dependent scenarios where scenario k is given by the sum of the model prediction for the observed feature vector and the residual associated with training observation k:

$$\Omega = \left\{ \mathbf{g}(\mathbf{x}^{n+1}; \boldsymbol{\varphi}) + \boldsymbol{\epsilon}^k \right\}_{k=1,\dots,n}. \tag{4.13}$$

We then solve Model (S) with uniformly weighted residual-based scenarios Ω.

4.3.5 Late Arrival Penalty Approximation

Instead of leveraging feature data to estimate the travel times distribution, and then using such estimates to approximate the expected penalty of a solution, we can use data to predict penalties directly. We define the penalty-based approximation model:

$$\hat{\mathbf{z}}^*_{\text{P}}(\mathbf{x}) \;=\; \arg\min_{\mathbf{z} \in \mathcal{Z}_\Theta} \hat{f}_\Theta(\mathbf{z}, \mathbf{x}) \;=\; \arg\min_{\mathbf{z} \in \mathcal{Z}_\Theta} C(\mathbf{z}) + \hat{Q}(\mathbf{z}, \mathbf{x}), \tag{L}$$

where $\hat{Q}(\mathbf{z}, \mathbf{x})$ approximates the conditional expected penalty $\mathbb{E}\left[Q(\mathbf{z}, \tilde{\mathbf{t}}) \,\middle|\, \tilde{\mathbf{x}} = \mathbf{x} \right]$.

By definition, the conditional expected penalty of a solution $\mathbf{z}$ in Equation (4.5) can be

decomposed into the sum of the late arrival penalties at the customers along the routes in $\mathbf{z}$. Clearly, the predicted penalty at a given customer i along a given route $\theta \in \Theta$ depends on customer and route characteristics, e.g., the expected penalty is correlated with the position of i in the route or the customer time window, in addition to the features $\mathbf{x}$ related to the travel times. Therefore, we define a feature projection function (FPF), $\mathbf{f} : \mathbb{R}^p \times \Theta \times \mathcal{V} \setminus \{0\} \mapsto \mathbb{R}^{\bar{p}}$, that combines travel time covariates with node and route characteristics to obtain a vector of penalty predictors $\mathbf{y} \in \mathbb{R}^{\bar{p}}$:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \theta, i). \tag{4.14}$$

Essentially, $\mathbf{f}(\mathbf{x}, \theta, i)$ projects $\mathbf{x}$ onto a $\bar{p}$-dimensional feature space with potential predictors of the penalty at customer $i \in \theta$. We describe the projected features in detail in Appendix 4.A.

A prediction model $h(\cdot)$ outputs late arrival penalty predictions based on the projected features $\mathbf{y}$ given by the FPF. We approximate the expected value $\hat{Q}$ of the second-stage penalty as:

$$\hat{Q}(\mathbf{z}, \mathbf{x}) = \sum_{\theta \in \Theta} \sum_{i \in \theta} h(\mathbf{y}; \mathbf{H})\, z_\theta \tag{4.15}$$

where $\mathbf{H}$ denotes the trained parameters of the prediction model $h(\cdot)$.

We adopt a supervised learning setting and build a training dataset with $\bar{n}$ observations of the projected features $\mathbf{y}^k$ and penalties π^k, for $k \in \{1, \ldots, \bar{n}\}$. Specifically, we generate the projected features, represented by $\mathbf{Y} = [\mathbf{y}^1, \ldots, \mathbf{y}^{\bar{n}}]$, from the original feature data $\mathbf{X}$, historical travel times $\mathbf{T}$, and a set of routes Θ. In our experiments, we construct a set Θ of routes by solving the other prescriptive models, i.e., point-based and SAA models, using the BP&C algorithm described in Section 4.4, and record all routes evaluated during the solution process. The corresponding penalties, represented by $\boldsymbol{\pi} = [\pi^1, \ldots, \pi^{\bar{n}}]$, can be computed by propagating the arrival times at each node along the route based on the respective travel times. The learning problem then corresponds to finding the model parameters that minimize some empirical risk of the form:

$$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} \sum_{k=1}^{\bar{n}} \mathcal{L}(h(\mathbf{y}^k; \mathbf{H}), \pi^k) \tag{4.16}$$

where $\mathcal{L}$ defines a loss function, e.g., a squared error, and the prediction model can be any supervised regression model, e.g., based on linear regression or neural networks.

4.4 Solution Methods for the Data-driven Prescriptive Models

We adopt a set-partitioning formulation, where the set of decision variables representing the feasible routes Θ can be exponentially large. To solve the proposed models, we rely on branch-and-price, which is an effective method for solving integer linear programs with a very large number of variables (Barnhart et al., 1998). In vehicle routing, the technique is applied to solve many problem variants that admit a set-partitioning formulation (Costa, Contardo, and Desaulniers, 2019). We start by introducing the solution method for solving Model (S) in Sections 4.4.1 and 4.4.2 before we explain how to adapt the method to solve Model (L) in Section 4.4.3.

4.4.1 Column Generation

In branch-and-price, we solve the continuous relaxation of the set-partitioning formulation by column generation. We define a restricted master problem (RMP) considering only a subset $\Theta' \subset \Theta$ of the set of feasible routes, which yields the following linear program:

$$\begin{aligned}
\min \quad & \sum_{\omega \in \Omega} \alpha^{\omega} f_{\Theta'}\left(\mathbf{z}, \mathbf{t}^{\omega}\right) && && \text{(RMP)} \\
\text{s.t.} \quad & \sum_{\theta \in \Theta'} \mathbb{1}(i \in \theta) z_{\theta} = 1, && i \in \mathcal{V} \setminus \{0\} \\
& \sum_{\theta \in \Theta'} z_{\theta} \leq K \\
& 0 \leq z_{\theta} \leq 1, && \theta \in \Theta'.
\end{aligned}$$

The column generation procedure repeatedly solves (RMP), identifies routes that correspond to negative reduced cost variables, and adds those routes to Θ'. To this end, we solve the corresponding pricing problem:

$$\min_{\theta \in \Theta} \quad \overline{C}_{\theta} \triangleq C_{\theta} + \sum_{\omega \in \Omega} \alpha^{\omega} \sum_{i \in \theta} \pi(a_{\theta}(i; \mathbf{t}^{\omega}) - \ell_i) - \sum_{i \in \theta} \gamma_i - \mu, \qquad \text{(PP-S)}$$

where, given a solution to (RMP), $\boldsymbol{\gamma} = [\gamma_i]_{i \in \mathcal{V} \setminus \{0\}}$ and μ are the dual values associated with Constraints (4.4a) and (4.4b), respectively. The pricing problem is an elementary shortest path problem with resource constraints (ESPPRC), which is $\mathcal{NP}$-hard.

4.4.2 Pricing Algorithm

We use an extend-and-bound labeling algorithm, where a label represents a partial path from the depot to a customer. The labeling procedure creates new labels by extending partial paths to all feasible customers. For each new label extension, we compute a lower bound on the reduced cost of all routes that can be generated from the label extension, i.e., a completion bound. Then, we discard the label extension if this bound is non-negative.

A label L_θ representing a path θ is a tuple $L_\theta = (i, \overline{C}_\theta, q_\theta, \tau_\theta)$, where $i \in \mathcal{V} \setminus \{0\}$ is the last customer in θ, $\overline{C}_\theta$ is the reduced cost, $q_\theta = \sum_{v \in \theta} q_v$ is the cumulated load along θ, and τ_θ is the earliest time at which service can start at customer i if θ is used to reach i:

$$\tau_\theta = \min_{\omega \in \Omega} \left\{ s_\theta(i; \mathbf{t}^\omega) \right\}, \tag{4.17}$$

considering all scenarios $\omega \in \Omega$.

If we extend label L_θ along an arc (i, j), we generate a new path $\theta' = \theta \oplus (i, j)$ with a corresponding label $L_{\theta'} = (j, \overline{C}_{\theta'}, q_{\theta'}, \tau_{\theta'})$, such that:

$$\overline{C}_{\theta'} = \overline{C}_\theta - c_{i0} + c_{ij} + c_{j0} + \sum_{\omega \in \Omega} \alpha^\omega \cdot \pi(a_{\theta'}(j; \mathbf{t}^\omega) - \ell_j) - \gamma_j \tag{4.18}$$

$$\tau_{\theta'} = \min_{\omega \in \Omega} \{ s_{\theta'}(j; \mathbf{t}^\omega) \}. \tag{4.19}$$

Path θ' is feasible if $q_{\theta'} \leq Q$. Otherwise, it is infeasible, and we discard the new label.

A completion bound $\widehat{T}(i, q)$ is a lower bound on the value of the best path starting at customer i and ending at the depot with a total load less than or equal to q. If the following condition holds, then we can discard label L_θ without losing any negative reduced cost route:

$$\overline{C}_\theta + \widehat{T}(i, Q - q_\theta) \geq 0. \tag{4.20}$$

Since we solve the pricing problem multiple times during the execution of the BP&C method, it is important to generate completion bounds quickly. Hence, we employ heuristic algorithms based on a resource-constrained shortest path (RCSP) problem and based on a knapsack formulation (cf. Florio, Hartl, and Minner 2020). Unlike many labeling algorithms from the literature, we do not discard labels based on dominance rules. In order to use dominance rules in our setting, we would need labels to store a resource corresponding to the service start time under each travel time scenario. We leave the investigation of dominance rules and the assessment of their effectiveness for future research.

Resource-Constrained Shortest Path (RCSP) Bound. We derive RCSP bounds by relaxing the elementary requirement of the ESPPRC and enforcing 2-cycle elimination, i.e., we allow paths with cycles only if cycles have a length of at least three customers.

Resource constraints correspond to the capacity constraints on the vehicles. We compute a lower bound on the late arrival penalties along a path θ based on the earliest service start time τ_θ. We provide a dynamic programming (DP) algorithm for computing the RCSP bounds in Appendix 4.B, where we apply a time discretization with a time step Δt such that we can efficiently store the optimal value of each subproblem. The DP algorithm solves the following recursive formulation:

$$\widehat{T}_{\text{RCSP}}(i, Q-q_\theta) = -c_{i0} + \min_{\substack{j \in \mathcal{V}\setminus\{0\}:\ \theta' = \theta \oplus (i,j),\\ j \neq i,\ \rho(i) \neq j,\\ q_\theta + q_j \leq Q}} \left\{ c_{ij} + \pi(\delta_{\theta'} - \ell_j) - \gamma_j + c_{j0} + \widehat{T}_{\text{RCSP}}(j, Q - q_{\theta'}) \right\}, \tag{4.21}$$

where $\theta' = \theta \oplus (i,j)$ denotes the extension of path θ along the arc (i,j) and δ_θ is a multiple of Δt representing a lower bound on the service start time of customer $i \in \theta$, such that $\tau_\theta \in [\delta_\theta, \delta_\theta + \Delta t]$. The resulting completion bound expresses the maximum reduced cost decrease from node i given that departure from node i is not before δ_θ.

Proposition 4.1 *Let θ be a route starting at the depot and ending at customer i with reduced cost $\overline{C}_\theta$ and cumulative demand q_θ. Let $L_{\theta'}$ be a label extension associated with the route $\theta' = \theta \oplus \mathcal{E}$, where $\mathcal{E} = (u_1, u_2, \ldots, u_L)$ is a path such that $u_j \notin \theta$, $\forall u_j \in \mathcal{E}$, and $q_{\theta'} \leq Q$. If $\overline{C}_\theta + \widehat{T}_{\text{RCSP}}(i, Q - q_\theta) \geq 0$, then the label extension has non-negative reduced cost, i.e., $\overline{C}_{\theta'} \geq 0$.*

Proof. We provide proof for the RCSP bound in Appendix 4.C. □

Knapsack Bound. Consider a label L_θ representing a route θ that ends at customer i. We build a $\{0,1\}$-knapsack problem with N items and capacity $Q - q_\theta$. We associate a weight $w_j = q_j$ to each item $j \in \{1, \ldots, N\}$, equal to the demand of customer $j \in \mathcal{V} \setminus \{0\}$, and a value consisting of the dual value γ_j minus a lower bound on the average penalty:

$$v_{ij}(\theta) = \gamma_j - \pi(\tau_\theta + \min_{\omega \in \Omega} t_{ij}^\omega - \ell_j), \tag{4.22}$$

such that the following inequality holds:

$$-v_{ij}(\theta) \ \leq \sum_{\omega \in \Omega} \alpha^\omega \cdot \pi(a_{\theta \oplus (i,j)}(j; \mathbf{t}^\omega) - \ell_j) - \gamma_j. \tag{4.23}$$

We can express the knapsack problem as an integer linear program:

$$\max \quad \sum_{j \in \mathcal{V} \setminus \{0\}} v_{ij}(\theta)\, z_j \tag{4.24}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{V} \setminus \{0\}} q_j z_j \leq Q - q_\theta \tag{4.25}$$

$$z_j = 0 \text{ if } j \in \theta \qquad \forall j \in \mathcal{V} \setminus \{0\} \tag{4.26}$$

$$z_j \in \{0, 1\} \qquad \forall j \in \mathcal{V} \setminus \{0\} \tag{4.27}$$

where Constraints (4.26) enforce that item j is considered for inclusion in the knapsack only if it is not already in route θ. Let $\mathbf{z}^*$ be an optimal solution to the $\{0,1\}$-knapsack problem and let $\overline{\mathbf{z}}$ be a solution to its linear relaxation. Then, we define the following completion bound:

$$\widehat{T}_{ks}(i, Q - q_\theta) = -\sum_{j \in \mathcal{V} \setminus \{0\}} v_{ij}(\theta)\, \overline{z}_j. \tag{4.28}$$

Proposition 4.2 *Let θ be a route starting at the depot and ending at customer i with reduced cost $\overline{C}_\theta$ and cumulative demand q_θ. Let $L_{\theta'}$ be a label extension associated with the route $\theta' = \theta \oplus \mathcal{E}$, where $\mathcal{E} = (u_1, u_2, \ldots, u_L)$ is a path such that $u_j \notin \theta$, $\forall u_j \in \mathcal{E}$, and $q_{\theta'} \leq Q$. If $\overline{C}_\theta + \widehat{T}_{ks}(i, Q - q_\theta) \geq 0$, then the label extension has non-negative reduced cost, i.e., $\overline{C}_{\theta'} \geq 0$.*

Proof. We provide proof for the knapsack bound in Appendix 4.C. □

4.4.3 Column Generation for Penalty-based Approximation

We now describe how the column generation (CG) approach from the previous sections can be modified to solve the penalty-based approximation of Model (L). The RMP for the penalty-based model optimizes the objective function:

$$\min \quad \hat{f}_{\Theta'}(\mathbf{z}, \mathbf{x}), \tag{RMP-L}$$

while the constraints remain the same as in (RMP). Accordingly, we incorporate the late arrival penalty prediction model into the pricing problem:

$$\min_{\theta \in \Theta} \quad \overline{C}_\theta \triangleq C_\theta + \sum_{i \in \theta} h(\mathbf{f}(\mathbf{x}^{n+1}, \theta, i); \mathbf{H}) - \sum_{i \in \theta} \gamma_i - \mu. \tag{PP-L}$$

To solve the pricing problem in (PP-L), we make the following modifications to the labeling algorithm described in Section 4.4.2. First, when extending a label L_θ along an

arc (i, j), we generate a new path θ' with a reduced cost given by:

$$\overline{C}_{\theta'} = \overline{C}_{\theta} - c_{i0} + c_{ij} + c_{j0} + h(\mathbf{f}(\mathbf{x}^{n+1}, \theta', j); \mathbf{H}) - \gamma_j, \tag{4.29}$$

where we replace the sample averaged penalty in Equation (4.18) by the predicted penalty.

Second, we cannot define the earliest service start time τ_θ as in Equation (4.17) since we do not assume a set of scenarios to be available under the penalty-based approximation model. Instead, we rely on the following assumption to compute τ_θ:

Assumption 4.1 (Earliest Arrival Times) *Given a route θ, it holds that*

$$a_\theta(i; \tilde{\mathbf{t}}) \geq \min_{k \in \{1,\dots,n\}} a_\theta(i; \mathbf{t}^k) \qquad \forall i \in \theta.$$

Assumption 4.1 states that, for a given route θ, any realization of the vector of travel times will lead to an arrival time at customer $i \in \theta$ that is not earlier than the most optimistic arrival time that one would observe given the historical travel times $\{\mathbf{t}^k\}_{k \in \{1,\dots,n\}}$ from the training data set $\mathbf{T}$. In practice, there often exists periods in which highways have no traffic and cars can travel at free flow, e.g., during quiet night hours. We argue that Assumption 4.1 is reasonable if the historical data contains travel time observations recorded at such periods with low traffic. Even if the historical data does not contain such observations, our assumption is not limiting as one can always augment the data set with artificial scenarios by computing free-flow travel times for each arc. Given the above assumption, we define the earliest service start time:

$$\tau_\theta = \min_{k \in \{1,\dots,n\}} \left\{ s_\theta(i, \mathbf{t}^k) \right\}, \tag{4.30}$$

which is independent of the penalty prediction model $h(\cdot)$. We run the pricing algorithm from Section 4.4.2 with the above modifications to solve Problem (PP-L).

4.5 Design of Experiments

The goals of our experiments are to (*i*) compare different data-driven methods for the CS-VRPTW, based on the average cost calculated on a test data set containing travel times and features, (*ii*) analyze the value of incorporating features in the VRPTW, and (*iii*) investigate the performance of the proposed methods under different generative models for travel times and features.

We implemented the BP&C method in `C++`, using CPLEX as the underlying linear programming solver. All experiments were conducted on the Narval computing cluster from

the Digital Research Alliance of Canada, Canada's national high-performance computing system. We set a time limit of 5 hours per run. We provide the source code and data to reproduce our experiments at [to be disclosed after peer-review].

4.5.1 Instances

We base our experiments on the instances of Solomon (1987), a standard benchmark for the VRPTW. Each instance describes a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, a vehicle capacity Q, demands q_i and time windows $[e_i, \ell_i]$ of each node $i \in \mathcal{V}$, with the demand for the depot being equal to zero. We define the deterministic cost c_{ij} to be the Euclidean distance between each pair of customers $(i, j) \in \mathcal{A}$, and assume a quadratic penalty function, i.e., $\pi(u) = u^2$ for $u \geq 0$.

We adopted 29 instances from problem sets R1, C1, and RC1, which have short time windows, allowing only a few customers per route. Problem set R1 contains randomly generated geographical data, C1 contains clustered data, and RC1 has a mix of random and clustered structures.

For each instance, we generate synthetic historical data $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{T} \in \mathbb{R}^{n \times |\mathcal{A}|}$ with $n = 100$ training travel time scenarios and $p = 10$ features. We selected the values for n and p reflecting, e.g., a typical VRP application in city logistics where one might have daily observations of features and average travel times for a certain period of the day. For each $k \in \{1, \ldots, n\}$, we sample a feature vector from a p-variate distribution $\mathbf{x}^k \sim \mathbb{X}$ and we sample travel times from an $|\mathcal{A}|$-variate distribution $\mathbf{t}^k \sim \mathbb{T}(\mathbf{x}^k)$ conditioned on the observed feature vector $\mathbf{x}^k$. We consider three different generative models, which specify the distributions $\mathbb{X}$ and $\mathbb{T}$.

Linear model. For each arc $(i, j) \in \mathcal{A}$, the deterministic cost c_{ij} defines a nominal travel time $\underline{t}_{ij}$ corresponding to the free-flow travel time of the arc. We define the stochastic travel times as the nominal travel times $\underline{\mathbf{t}} = [\underline{t}_{ij}]_{(i,j)\in\mathcal{A}}$ plus a random noise term which depends linearly on the features:

$$\tilde{\mathbf{t}}_{\text{linear}}(\tilde{\mathbf{x}}) = \underline{\mathbf{t}} + \mathbf{B}^\top \tilde{\mathbf{x}} + \tilde{\boldsymbol{\varepsilon}} \tag{4.31}$$

where $\mathbf{B}$ is a $p \times |\mathcal{A}|$ matrix whose columns are given by the vectors $\mathbf{b}_{ij} \in \mathbb{R}^p$ for $(i, j) \in \mathcal{A}$. We sample values in $\mathbf{b}_{ij}$ from a uniform distribution with support ranging from 1% to 20% of the corresponding nominal travel time $\underline{t_{ij}} = c_{ij}$. Moreover, we assume that features are binary, $\mathbf{x} \in \{0, 1\}^p$, corresponding to categorical data, e.g., day of the week, holidays, or roadworks. The noise term $\tilde{\boldsymbol{\varepsilon}}$ follows a multivariate normal distribution with zero mean, and covariance matrix generated according to the method of Rostami et al. (2021), such that noise values at different arcs are correlated. Lastly, we assume that travel times on

each arc must be greater than or equal to the nominal travel time, and we truncate travel times whenever necessary.

Exponential model. For each arc $(i,j) \in \mathcal{A}$, we consider that features are related to the travel times via an exponential function:

$$\tilde{t}_{ij} = \underline{t_{ij}} + 0.2\ \underline{t_{ij}}\ \exp\left(2\,\mathbf{b}_{ij}^{\top}\,\tilde{\mathbf{x}}\right) + \tilde{\varepsilon}_{ij} \tag{4.32}$$

where features now follow a uniform distribution between 0 and 1. We generate the parameter vectors $\mathbf{b}_{ij} \in \mathbb{R}^p$ by sampling each element from a uniform distribution between 0.1 and 0.3, and we multiply each element by -1 with a probability of 0.2. Due to the exponential travel times, having normally distributed $\tilde{\varepsilon}_{ij}$ does not provide sufficient noise. Therefore, we assume that $\tilde{\varepsilon}_{ij}$ follows a log-normal distribution with zero mean and standard deviation $\sigma_{\varepsilon} = 1$.

Sigmoidal model. For each arc $(i,j) \in \mathcal{A}$, we generate travel times:

$$\tilde{t}_{ij} = \underline{t_{ij}} + \underline{t_{ij}}\,\sigma\Big(32(\,\frac{1}{2}\mathbf{b}_{ij}^{\top}\,\mathbb{1} - \mathbf{b}_{ij}^{\top}\,\tilde{\mathbf{x}})\Big) + \tilde{\varepsilon}_{ij} \tag{4.33}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. Features follow a uniform distribution between 0 and 1. The noise term $\tilde{\varepsilon}_{ij}$ follows a log-normal distribution with mean 0 and standard deviation $\sigma_{\varepsilon} = 1.2$. We generate the parameter vectors $\mathbf{b}_{ij}$ by sampling each element from a uniform distribution between 0.3 and 0.8, and we multiply each element by -1 with a probability of 0.2. Due to its characteristic shape, we can interpret the sigmoidal model as representing two feature-dependent *states of traffic*, e.g., a congested and a non-congested state.

4.5.2 Prescriptive Metrics

We use the term *prescription* to refer to any function $\hat{\mathbf{z}}(\mathbf{x})$ that provides a decision given the feature observation $\tilde{\mathbf{x}} = \mathbf{x}$. We evaluate the performance of a given prescription $\hat{\mathbf{z}}(\cdot)$ by its expected cost under the true joint distribution of travel times and features:

$$R(\hat{\mathbf{z}}) = \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathbb{X}}\Big[\mathbb{E}_{\tilde{\mathbf{t}}\sim\mathbb{T}}[f_{\Theta}(\hat{\mathbf{z}}(\mathbf{x}),\tilde{\mathbf{t}})\,|\,\tilde{\mathbf{x}}=\mathbf{x}]\Big] \tag{4.34}$$

In practice, calculating the above expectation is often intractable. We therefore estimate $R(\hat{\mathbf{z}})$ using a test data set, which we construct by sampling a set $\mathcal{X} = \{\mathbf{x}^k \sim \mathbb{X}\}_{k=1,\dots,n_{\mathcal{X}}}$ of features and, for each feature vector $\mathbf{x} \in \mathcal{X}$, a set $\mathcal{T}(\mathbf{x}) = \{\mathbf{t}^k \sim \mathbb{T}(\mathbf{x})\}_{k=1,\dots,n_{\mathcal{T}}}$ of travel times from the corresponding generative model. The empirical *test cost* $\hat{R}(\hat{\mathbf{z}})$ of a

prescription $\hat{\mathbf{z}}(\cdot)$ is an approximation of $R(\hat{\mathbf{z}})$ which is given by:

$$\hat{R}(\hat{\mathbf{z}}) = \frac{1}{n_{\mathcal{X}} \cdot n_{\mathcal{T}}} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} f_{\Theta}\left(\hat{\mathbf{z}}(\mathbf{x}), \mathbf{t}\right). \tag{4.35}$$

In Section 4.6, we provide test cost results for prescriptions based on different models, where we replace $\hat{\mathbf{z}}$ in Equation (4.35) by the solution to the corresponding model.

4.5.3 Full-information benchmarks

We compare our models against benchmark solutions that rely on knowledge of travel times and feature distributions. The following benchmark solutions are impractical in a real-world setting, as the decision-maker does not know the underlying distributions. However, these benchmark solutions provide us with a relative measure of how well the proposed practical models perform.

Full-information solution. Based on the test data set, we can approximate the full-information solution of the CS-VRPTW, given a feature vector $\mathbf{x} \in \mathcal{X}$, as:

$$\hat{\mathbf{z}}^*_{\text{FULL}}(\mathbf{x}) = \arg\min_{\mathbf{z} \in \mathcal{Z}_{\Theta}} \frac{1}{n_{\mathcal{T}}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} f_{\Theta}(\mathbf{z}, \mathbf{t}), \tag{4.36}$$

with optimal objective value:

$$\hat{v}^*_{\text{FULL}}(\mathbf{x}) = \min_{\mathbf{z} \in \mathcal{Z}_{\Theta}} \frac{1}{n_{\mathcal{T}}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} f_{\Theta}(\mathbf{z}, \mathbf{t}). \tag{4.37}$$

Given the prescription $\hat{\mathbf{z}}^*_{\text{FULL}}$, the empirical test cost is given by:

$$\hat{R}_{\text{FULL}} = \hat{R}(\hat{\mathbf{z}}^*_{\text{FULL}}) = \frac{1}{n_{\mathcal{X}} \cdot n_{\mathcal{T}}} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} f_{\Theta}\left(\hat{\mathbf{z}}^*_{\text{FULL}}(\mathbf{x}), \mathbf{t}\right) = \frac{1}{m_{\mathcal{X}}} \sum_{\mathbf{x} \in \mathcal{X}} \hat{v}^*_{\text{FULL}}(\mathbf{x}), \tag{4.38}$$

which provides a lower bound for the empirical test cost of any model. With this definition, we calculate the full-information percentage gap of a prescription $\hat{\mathbf{z}}$ as:

$$\rho(\hat{\mathbf{z}}) = \frac{\hat{R}(\hat{\mathbf{z}}) - \hat{R}_{\text{FULL}}}{\hat{R}_{\text{FULL}}}. \tag{4.39}$$

Predict with full information, then optimize. The performance of a prescription under the PTO framework depends on the choice of predictive model $\mathbf{g}(\cdot)$ providing travel time predictions $\hat{\mathbf{t}} = \mathbf{g}(\mathbf{x}; \boldsymbol{\varphi})$. For the purpose of benchmarking, we define the PTO-F

problem, which assumes that the predictive model perfectly predicts the expected travel times given observed features:

$$\mathbf{z}^*_{\text{PTO-F}}(\mathbf{x}) = \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} f_\Theta\left(\mathbf{z}, \bar{\mathbf{t}}\right) \quad \text{with } \bar{\mathbf{t}} = \mathbb{E}\left[\tilde{\mathbf{t}} \,\middle|\, \tilde{\mathbf{x}} = \mathbf{x}\right]. \tag{4.40}$$

A solution for the PTO-F problem requires knowledge of the joint distribution of travel times and features. We can approximate the PTO-F solution using the test data set:

$$\hat{\mathbf{z}}^*_{\text{PTO-F}}(\mathbf{x}) = \arg\min_{\mathbf{z}\in\mathcal{Z}_\Theta} f_\Theta\left(\mathbf{z}, \hat{\mathbf{t}}\right) \quad \text{with } \hat{\mathbf{t}} = \frac{1}{n_\mathcal{T}} \sum_{\mathbf{t}\in\mathcal{T}(\mathbf{x})} \mathbf{t}. \tag{4.41}$$

We note that although PTO-F often performs better than PTO with practical prediction models, PTO-F does not necessarily provide a lower bound for PTO regarding test cost since neither PTO nor PTO-F account for the structure of the downstream optimization problem. Specifically, the true conditional expected travel times do not necessarily correspond to the travel times leading to minimum cost.

4.5.4 Prescriptive methods

We compare the ten prescriptive methods summarized in Table 4.1. The top eight methods are derived from the data-driven prescriptive models of Section 4.3 and correspond to practical methods for solving the CS-VRPTW. The bottom two prescriptive methods are full-information benchmarks, included for comparison, and cannot be applied in practice as they require knowledge of the true travel times distribution (see Section 4.5.3).

Table 4.1: Data-driven prescriptive methods (top) and full-information benchmarks (bottom).

Method	Model	Description
D-avg	(P)	Predicted travel times given by average travel times $\hat{\mathbf{t}} = \sum_{k=1}^{n} \mathbf{t}^k/n$
PTO-OLS	(P)	Predicted travel times $\hat{\mathbf{t}} = \boldsymbol{\varphi}\mathbf{x}^{n+1}$ using OLS regression (*predict-then-optimize*)
PTO-kNN	(P)	Predicted travel times $\hat{\mathbf{t}}$ given by k-NN regression (*predict-then-optimize*)
SAA	(S)	Travel time scenarios are the set of historical travel times $\Omega = \{\mathbf{t}^1, \ldots, \mathbf{t}^n\}$ with $\alpha^\omega = 1/n$
SAA-kNN	(S)	Historical travel times with weights α^ω given by k-NN regression (Bertsimas and Kallus, 2019)
CSAA	(S)	Feature-dependent travel time scenarios (see Section 4.3.3)
RSAA	(S)	Feature-dependent travel time scenarios based on linear regression residuals (see Section 4.3.4)
P-NN	(L)	Penalty prediction model based on fully-connected neural network
PTO-F	(P)	*Predict-then-optimize* under true conditional expected travel times $\hat{\mathbf{t}} = \mathbb{E}_{\mathbf{t}\sim\mathbb{T}}[\mathbf{t} \,\vert\, \mathbf{x}^{n+1}]$
Full	(S)	Full-information lower bound under the true travel times distribution $\mathbf{t} \sim \mathbb{T}(\mathbf{x}^{n+1})$

D-avg is the point-based model under average travel times based on historical data. PTO-OLS is the point-based model under the *predict-then-optimize* framework, where the coefficients $\boldsymbol{\varphi} \in \mathbb{R}^{|\mathcal{A}|\times p}$ of the prediction model are trained by solving a multiple least squares regression on the historical data. PTO-kNN is the point-based model under the

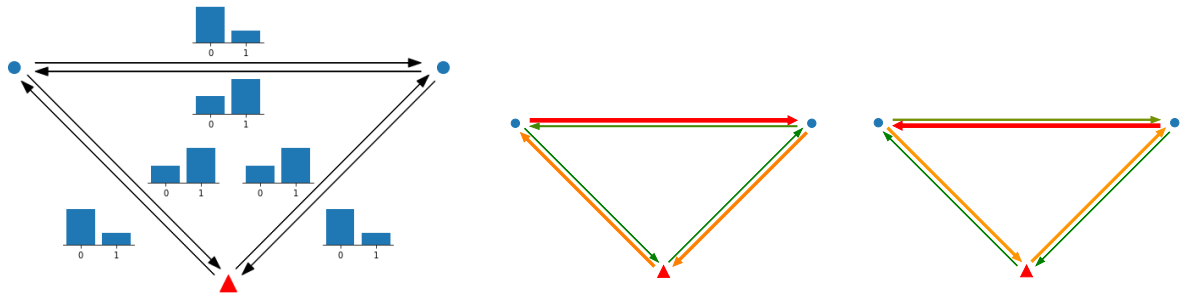
predict-then-optimize framework, where we used a k-nearest neighbor regression model to predict the travel times. The SAA model assumes that the travel time scenarios are given by the historical travel times. SAA-kNN corresponds to the approach of Bertsimas and Kallus (2019), where we adopted a k-nearest neighbors regression model for the weight function. As described in Section 4.3.3, CSAA generates a set of travel time scenarios Ω from a multivariate Gaussian distribution with mean and covariance given by $\hat{\mathbf{B}}^\top \mathbf{x}^{n+1}$ and $\hat{\mathbf{\Sigma}}$, respectively. RSAA generates a set of travel time scenarios Ω based on the residuals of a linear regression model trained on historical data. P-NN is the penalty-based approximation model in which the late arrival penalty prediction model is a fully-connected neural network with two hidden layers with 100 neurons each, trained with the Adam optimization algorithm to minimize the squared error loss with an ℓ_2 regularization term equal to 0.1. Note that methods D-avg and SAA ignore the features and only consider the historical travel times when searching for a solution for the current period $n + 1$.

4.6 Computational Results

We start our analysis with a small example illustrating the benefit of incorporating information from feature data in the VRPTW formulation, which eases the reader to develop an intuition why feature-based approaches can yield results superior to feature-agnostic approaches. We then report results on larger instances, where we discuss the test cost performance of the different prescriptive methods.

4.6.1 Illustrative example

We consider a network with $N = 2$ customers and a training data set with $n = 2$ samples and $p = 1$ binary feature. Figure 4.1a shows the customer locations and arcs connecting each pair of customers. A small bar chart next to each arc shows the corresponding travel times in the y-axis as a function of the feature value in the x-axis. We omitted the values in the y-axis since this illustrative example is not concerned with specific travel time values but rather aims to show the relation between travel times and features. Figure 4.1b and 4.1c show the two possible scenarios, representing travel time realizations when the feature value equals $x = 0$ and $x = 1$, respectively. The color and line thickness of each arc indicate the amount of congestion, with thicker lines representing more congestion, and different colors indicating whether the arc is strongly congested (red), mildly congested (orange), or free from congestion (green). Under the scenario displayed in Figure 4.1b, a route that starts at the depot and visits customers in a clockwise direction will experience more congestion than a counter-clockwise route. Therefore, if the decision-maker finds herself

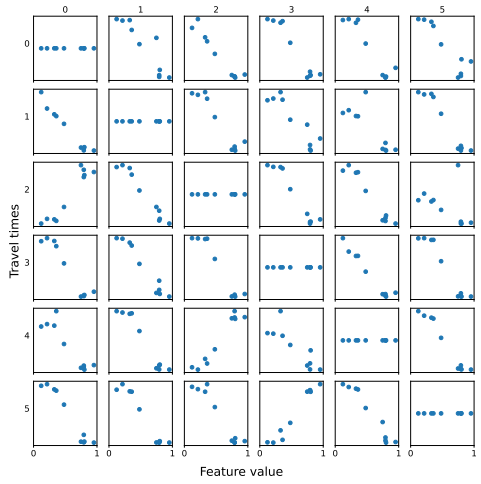


(a) Travel times as a function of feature value. (b) Travel times for $x = 0$. (c) Travel times for $x = 1$.

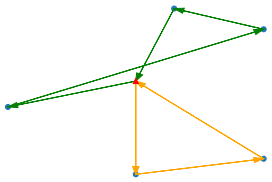
Figure 4.1: Joint distribution and different realizations of travel times and feature variable.

in a scenario where $x = 0$, following the counter-clockwise route is optimal. Figure 4.1c shows the reverse pattern, i.e., when $x = 1$, the counter-clockwise route is more congested than the clockwise route. Existing methods based on SAA, which are widely adopted in the field, would provide a clockwise route that is optimal when $x = 0$ but not when $x = 1$. In contrast to featureless methods, feature-dependent solutions for the CS-VRPTW can provide the optimal route in both cases.

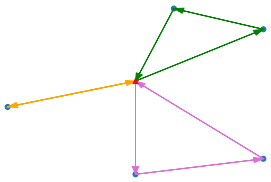
We now extend our previous example and consider a network with $N = 5$ customers and a training data set with $n = 10$ samples and $p = 1$ continuous feature with domain $x \in [0, 1]$. To capture different degrees of congestion, we assume sigmoidal travel times (see Section 4.5.1). In Figure 4.2a, each scatter plot in row $i \in \{0, \ldots, 5\}$ and column $j \in \{0, \ldots, 5\}$ shows the travel times and features in the training data set corresponding to arc $(i, j) \in \mathcal{A}$. For a more palatable exposition, we normalized the travel times of each arc based on its nominal free-flow travel time. We compare the optimal solutions of two featureless prescriptive methods, i.e., D-avg and SAA, against CSAA and full-information solutions. Figures 4.2c and 4.2b show the solution structures that emerge in our example. Note how the solution structures are fundamentally different from each other. In particular, Solution A requires three vehicles, while Solution B requires only two vehicles. In Figure 4.2d, we show the solution structures obtained by each method under different realizations of the feature variable. When $x = 0.28$, the featureless point-based approximation (D-avg) retrieves the full-information solution, but it fails to do so when $x = 0.83$. We see the opposite behavior for the featureless SAA method, i.e., it fails for $x = 0.28$ but can retrieve the full-information solution when $x = 0.83$. In contrast to the featureless approaches, CSAA provides feature-dependent solutions that match the full-information solution in both cases. This small example shows that ignoring information from feature data can lead to suboptimal solutions, underlying the benefit of the proposed CS-VRPTW formulation.



(a) Relation between travel times and features in the training data set for each arc.



(b) Solution structure A.



(c) Solution structure B.

Feature realization	D-avg	SAA	CSAA	Full
$x = 0.28$	**A**	B	**A**	**A**
$x = 0.83$	A	**B**	**B**	**B**

(d) Optimal solution structures of each method.

Figure 4.2: Data set and solution structures of our illustrative example.

4.6.2 Results

We analyze the suitability of the different prescriptive methods in solving the CS-VRPTW. Specifically, we investigate the performance of each method in terms of their test cost $\hat{R}$ and full-information percentage gaps ρ. To provide a meaningful comparison of the prescriptive methods, we only consider instances for which all methods could find a near-optimal solution with an optimality gap of at most 1%. We restrict the discussion in the subsequent analyses to instances with 25 and 50 customers, for which we obtained a reasonably large set of near-optimal solutions. We provide detailed results for instances with 25 and 50 customers in Appendices 4.D and 4.E, respectively. We focus on larger instances with 75 and 100 customers in Section 4.6.3, where we study the scalability of the BP&C algorithm. As we later discuss, the P-NN method has longer run times and could not solve many instances with 50 customers. Therefore, we show results for P-NN only for instances with 25 customers.

Average test cost performance. Table 4.3 summarizes the test cost performance of all methods, showing the full-information percentage gap of test costs averaged over all instances with a common number of customers N, generative model, and instance type. In the last column, we show the absolute test cost values of the full-information benchmark. Among the data-driven prescriptive methods, we highlight the lowest values on each row with boldface numbers.

In most settings, CSAA and RSAA achieve the lowest test costs among the data-driven prescriptive methods. In particular, they outperform the PTO-F benchmark, even though PTO-F has an unfair advantage of having access to the full distribution. As discussed in

Table 4.3: Overview of average test costs for different instance types and generative models.

N	Gen. model	Inst. type	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	P-NN	PTO-F	Full (Abs.)
25	Lin.	R	23.42	3.38	0.97	11.00	4.25	**0.06**	0.11	0.59	0.99	483.19
		C	**0.00**	**0.00**	**0.00**	**0.00**	0.48	**0.00**	**0.00**	0.14	0.00	192.66
		RC	17.32	6.00	2.97	16.48	3.14	0.42	**0.38**	2.56	2.11	374.16
	Exp.	R	65.10	15.07	3.71	15.73	6.37	2.24	**2.07**	2.54	2.18	624.45
		C	44.37	3.98	12.67	42.28	8.52	2.82	3.47	**2.16**	7.94	220.52
		RC	99.82	24.70	13.24	34.30	10.53	**7.75**	9.58	10.52	6.03	514.96
	Sig.	R	97.60	19.10	39.06	58.83	18.95	13.70	**12.89**	23.03	16.96	612.18
		C	85.70	**30.37**	74.45	72.38	45.89	41.00	42.48	47.35	81.26	231.30
		RC	233.19	**31.12**	89.02	120.53	34.35	32.77	37.99	52.42	49.80	516.79
50	Lin.	R	23.49	4.14	2.06	14.20	3.47	0.38	**0.31**	-	1.33	917.0
		C	1.14	5.22	0.49	1.14	0.89	**0.16**	**0.16**	-	0.49	369.6
		RC	55.09	5.76	4.87	23.38	7.84	0.71	**0.64**	-	3.69	896.9
	Exp.	R	49.12	13.31	6.96	13.75	6.27	**3.67**	4.34	-	3.76	1171.4
		C	45.44	10.37	5.25	30.75	3.42	2.89	**2.36**	-	5.21	432.2
		RC	82.81	34.14	22.35	25.20	28.06	**6.19**	9.63	-	4.80	1274.0
	Sig.	R	191.73	34.13	67.16	95.00	32.86	**24.37**	26.51	-	27.82	1229.0
		C	105.19	**23.49**	62.00	67.45	28.58	34.54	29.78	-	46.55	451.3
		RC	303.64	49.66	101.90	157.20	54.73	**44.97**	48.50	-	35.49	1413.9

previous works (cf. Bertsimas and Kallus 2019; Elmachtoub and Grigas 2021), travel time predictions minimizing a least squares deviation may not lead to optimal decisions since they do not account for the structure of the optimization objective. In line with previous works, our results show that PTO-F often finds suboptimal decisions despite using the true expected travel times under full distributional information.

Analysis of test cost distributions. We focus on the distribution of test costs for instances with 25 customers. Figure 4.3 presents full-information gaps of each method, summarized as boxplots with whiskers that extend to 1.5 times the interquartile range. Points outside this range are marked as outliers and noted with an "×".

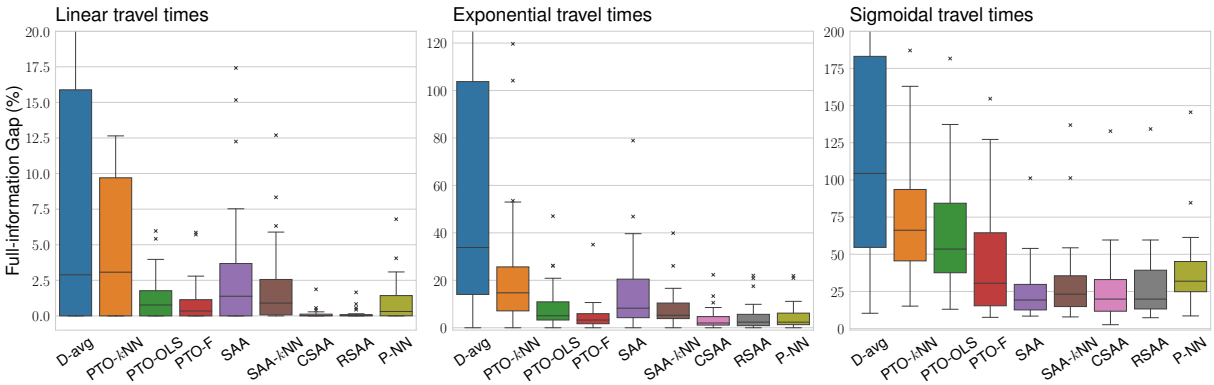


Figure 4.3: Distribution of test costs for instances with 25 customers from different generative models.

We first note that, as expected, CSAA and RSAA have the smallest gaps for instances

with linear travel times, as the data in this case conforms with the model assumptions. Second, we observe that P-NN provides competitive solutions in general and, especially for instances with exponential travel times, often outperforms all other methods. In particular, P-NN often improves upon PTO-based methods, highlighting the advantage of directly predicting the late arrival penalty rather than relying on travel time predictions. Still, CSAA and RSAA present equal or even superior performance in this non-linear setting. Third, for instances with sigmoidal travel times, we observe that SAA-based methods, i.e., SAA-kNN, CSAA, RSAA, and the classical SAA, have the smallest full-information gaps. In this highly non-linear setting, we benefit from using scenarios that better capture travel time variability, as illustrated in our example in Section 4.6.1. In general, we conclude that simpler methods that extend SAA using feature-dependent scenarios are the most promising, providing solutions that are closest to the full-information benchmark. Although it might be surprising that CSAA and RSAA can outperform the more sophisticated P-NN method, we argue that predicting late arrival penalties is not straightforward because the training data set must reflect the relation between the different combinatorial structures, encoded in the projected features, and the resulting penalty. The proposed penalty prediction model is not always capable of learning all the combinatorial intricacies of the VRP problem.

Trade-offs between first and second-stage costs. We analyze the contributions of the first-stage and second-stage costs to the total test cost. Figure 4.4 displays the average test cost of each method for instances with an exponential generative model, where we decompose the test cost $\hat{R}(\hat{\mathbf{z}})$ of a solution as the sum of the first-stage cost $\hat{C}$ and the second-stage cost $\hat{Q}$:

$$\hat{R}(\hat{\mathbf{z}}) = \frac{1}{n_{\mathcal{X}} \cdot n_{\mathcal{T}}} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} C(\hat{\mathbf{z}}) + Q(\hat{\mathbf{z}}, \mathbf{t}) = \hat{C}(\hat{\mathbf{z}}) + \hat{Q}(\hat{\mathbf{z}}). \tag{4.42}$$

The top, middle, and bottom plots show results for instances of types R, C, and RC, respectively. Full-information solutions have first-stage costs that are often larger than those of other methods, showing that it is often beneficial to incur larger first-stage costs and compensate them with comparatively small late arrival penalties. On the contrary, optimizing for the first-stage cost can lead to solutions with large penalties and, consequently, large overall test costs, as we see, e.g., in the middle plot for the D-avg method. Compared to practical PTO methods, i.e., PTO-OLS and PTO-kNN, our proposed CSAA and P-NN methods often present larger first-stage costs but smaller total test costs, bearing more similarity to the results from the full-information benchmark.

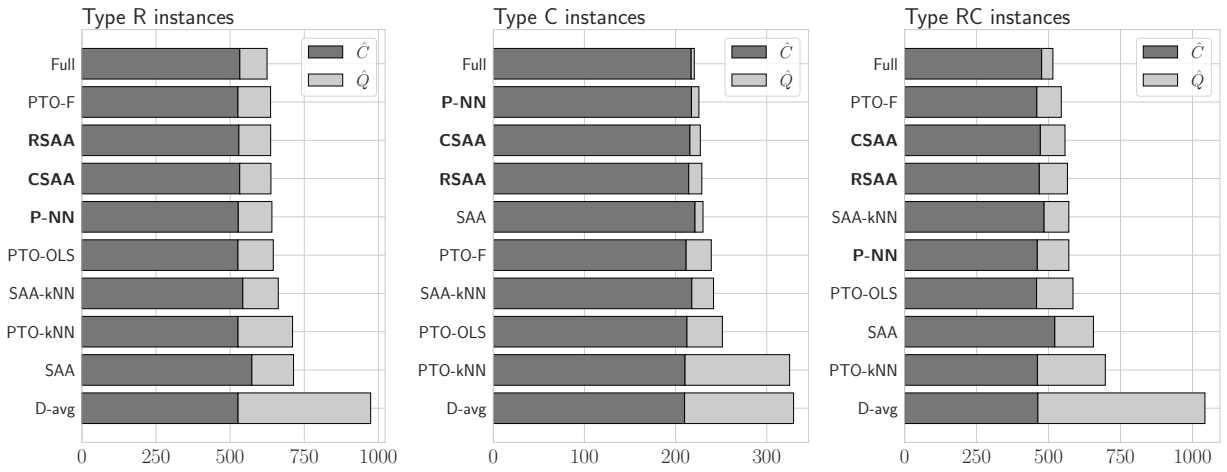


Figure 4.4: Average first-stage and second-stage test costs for instances from the exponential generative model.

4.6.3 Scaling to larger instances

Figure 4.5 compares the solution times of the different methods. For this plot, we considered all instances with 25 customers and a linear generative model. We observe that P-NN shows the longest run times, followed by SAA and the full-information benchmark. Recall that we solve all methods using the BP&C algorithm described in Section 4.4. Specifically for the P-NN method, we adopt the modified pricing problem described in Section 4.4.3. Consequently, with the exception of the P-NN method, the solution algorithms to all data-driven models differ only in the specific travel time scenarios given as input to the BP&C algorithm.

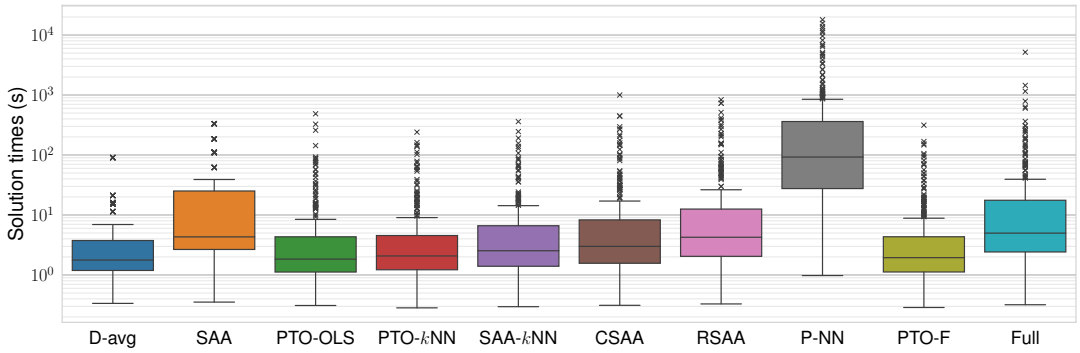


Figure 4.5: Solution times of different methods for instances with 25 customers with a linear generative model.

Figure 4.6 shows how solution times increase as we increase the number of customers. Here, we focus on the CSAA method and consider instances with a linear generative model. With 25 customers, we can solve most instances in less than 10 seconds. Due to the $\mathcal{NP}$-hardness of the problem, solution times increase drastically, and we observe some

runs reaching the 5-hour time limit already with 50 customers. With 100 customers, we can only solve very few instances within the time limit.

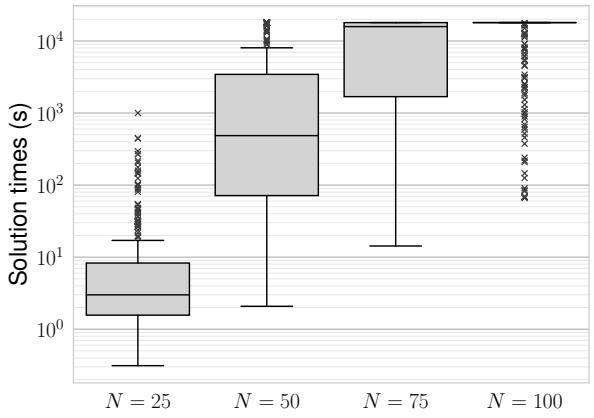


Figure 4.6: Solution times of the CSAA method for increasing number of customers on instances with a linear generative model.

4.7 Conclusions

This work connects the recent literature on contextual optimization with the established research field of VRPs. We introduced a novel formulation incorporating contextual information into the stochastic VRPTW. To solve the proposed formulation, we derived several data-driven prescriptive models by (i) applying existing contextual optimization methods to our problem setting and (ii) proposing novel methods based on conditional SAA and penalty-based prediction. From a computational perspective, solving the VRPTW is a challenging task for which many solution techniques exist in the literature. We showed how state-of-the-art techniques can be adapted when contextual information is available, leading to a customized BP&C algorithm that can solve instances with up to 100 customers. We analyzed the out-of-sample cost performance of the data-driven methods. We observed that the penalty-based approximation model that relies on penalty predictions provides competitive solutions, but in general, an SAA method based on feature-dependent scenarios yields solutions that are closest to the full-information benchmark.

This work raises several possibilities for future research. First, although we studied various data-driven methods and compared them against different benchmarks, we certainly did not exhaustively explore all possible approaches for solving the CS-VRPTW. Future research could investigate variations of the presented methods, e.g., the use of non-linear probabilistic models within the CSAA approach, or propose novel models and algorithms. Second, while extending the presented techniques to some VRP variants might be straightforward, e.g., with stochastic service times, other problem variants could lead

to more interesting research problems. Finally, since this paper focused on modeling aspects and exact solution methods, one open question is how to adapt existing heuristic methods to harness contextual information to efficiently solve larger instances.

Acknowledgments

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the research group Advanced Optimization in a Networked Economy (AdONE, GRK2201/277991500). This support is gratefully acknowledged. This research was enabled in part by support provided by Calcul Québec (`https://www.calculquebec.ca/`) and the Digital Research Alliance of Canada (`https://alliancecan.ca`).

Appendix 4.A Feature Projection Function.

The FPF is a function $\mathbf{f} : \mathbb{R}^p \times \Theta \times \mathcal{V} \setminus \{0\} \mapsto \mathbb{R}^{\bar{p}}$ that generates a $\bar{p}$-dimensional vector of projected features given a feature vector $\mathbf{x} \in \mathbb{R}^p$, a route $\theta \in \Theta$, a node $i \in \theta$, and travel time estimates $\hat{\mathbf{t}}$. The projected features are split into groups, as follows:

Table 4.4: Feature Projection Function: predictors of the penalty at customer $i \in \theta$ returned by $\mathbf{f}(\mathbf{x}, \theta, i; \hat{\mathbf{t}})$

Predictor	Group	Description
$x_1, \ldots, x_p$	(a)	Travel time covariates (original features)
e_i	(b)	Start of time window of customer $i \in \theta$
ℓ_i	(b)	End of time window of customer $i \in \theta$
$\ell_{\rho(i)}$	(b)	End of time window of customer $\rho(i) \in \theta$ that precedes $i \in \theta$
$c_{\rho(i),i}$	(b)	Transportation cost of the arc from $\rho(i)$ to i
$\hat{\sigma}^2_{\rho(i),i}$	(b)	Estimated variance in the travel time from $\rho(i)$ to i
k_i	(b)	Position of customer i along route θ
$a_\theta(i; \underline{\mathbf{t}})$	(c)	Lower bound on arrival time at customer $i \in \theta$ with free-flow travel times $\underline{\mathbf{t}}$
$s_\theta(i; \underline{\mathbf{t}})$	(c)	Lower bound on service start time at customer $i \in \theta$ with free-flow travel times $\underline{\mathbf{t}}$
$(a_\theta(i; \underline{\mathbf{t}}) - \ell_i)^+$	(c)	Lower bound on lateness
$\pi(a_\theta(i; \underline{\mathbf{t}}) - \ell_i)$	(c)	Lower bound on penalty
$\hat{t}_{\rho(i),i}$	(d)	Predicted travel time of the arc from $\rho(i)$ to i
$a_\theta(i; \hat{\mathbf{t}})$	(d)	Arrival times at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$s_\theta(i; \hat{\mathbf{t}})$	(d)	Service start time at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$(a_\theta(i; \hat{\mathbf{t}}) - \ell_i)^+$	(d)	Lateness at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$\pi(a_\theta(i; \hat{\mathbf{t}}) - \ell_i)$	(d)	Penalty at customer $i \in \theta$ given predicted travel times $\hat{\mathbf{t}}$
$\xi_\theta(i; \hat{\mathbf{t}})$	(e)	Variability model of service start time at customer $i \in \theta$ (see Appendix 4.A.1)

4.A.1 Variability of service start time.

In order to derive penalty predictors that take into account travel time variability and its impact on the service start time, we introduce a measure of variability of service start time, which we denote as *service start time risk*. Given a route θ and a customer $i \in \theta$, the distribution of $\max\{e_i, a_\theta(i; \tilde{\mathbf{t}})\}$ (the service start time at customer i) is, in general, truncated, because of possible early arrivals and waiting times at customer i and at other customers previously visited. Such a truncation decreases service start time variability; therefore, a covariate for service start time risk should consider both travel time variability and the likelihood of early arrivals along a route.

Let $\mathbf{\Sigma} = [\sigma_{ij,lm}]_{(i,j),(l,m)\in\mathcal{A}}$ be the travel times covariance, and let $\sigma^2_{ij} = \sigma_{ij,ij}$. Given a route $\theta = (v_1, \ldots, v_L)$, we denote by $\xi_\theta(i)$ the service start time risk at customer $i \in \theta$, and consider the following risk propagation model:

$$\xi_\theta(v_k) = \begin{cases} (1 - \mathcal{P}_\theta(v_k))\sigma^2_{0v_k}, & \text{if } k = 1, \\ (1 - \mathcal{P}_\theta(v_k))(\xi_\theta(v_{k-1}) + \sigma^2_{v_{k-1}v_k} + 2\sigma_{v_{k-2}v_{k-1},v_{k-1}v_k}), & \text{otherwise,} \end{cases} \tag{4.43}$$

where $\mathcal{P}_\theta(i) = \mathbb{P}(a_\theta(i; \tilde{\mathbf{t}}) < e_i \mid \mathbf{x}^{n+1})$ is the early arrival probability at customer $i \in \theta$

conditional on observed features $\mathbf{x}^{n+1}$, and $v_0 = 0$.

Model (4.43) propagates the variabilities of travel time and service start time along route θ as far as early arrival probabilities are low. When the early arrival probability $\mathcal{P}_\theta(i)$ is high, the service start time risk at customer i is low, since service occurs at instant e_i with high probability. In this case, the service start time risk at other customers along route θ following customer i also decreases. Note that the model accounts for travel time correlation between adjacent arcs in the network.

Service start time risk measures $\xi_\theta(i)$, $i \in \theta$, cannot be computed directly because the travel time distribution is unknown. Following our distribution-free approach, we estimate $\mathbf{\Sigma}$ and $\mathcal{P}_\theta(i)$ (and hence $\xi_\theta(i)$) from data. Let $\hat{\mathbf{\Sigma}} = [\hat{\sigma}_{ij,lm}]_{(i,j),(l,m)\in\mathcal{A}}$ be the estimated travel times covariance (as described in Section 4.3.3) and let $\hat{\sigma}^2_{ij} = \hat{\sigma}_{ij,ij}$. In the remainder of this section, we discuss how to estimate $\mathcal{P}_\theta(i)$ for any route θ. To this end, we let $\mathbf{g} : \mathbb{R}^p \mapsto \mathbb{R}^{|\mathcal{A}|}$ be a travel time prediction model. Further, we assume that for each customer $i \in \mathcal{V} \setminus \{0\}$ a set of training routes Θ_i is available, where $i \in \theta$ for all $\theta \in \Theta_i$. This is an unrestrictive assumption as these training routes may be arbitrary routes, e.g., generated by solving other VRP models. Finally, let $\rho_\theta(v_k)$ be the node that precedes v_k in route $\theta = (v_1, \ldots, v_L)$, that is, $\rho_\theta(v_k) = 0$ if $k = 1$, and $\rho_\theta(v_k) = v_{k-1}$ if $k \geq 2$.

Given a vector $\mathbf{x}$ of travel time covariates, let $\mathbf{w}_{i,\theta}(\mathbf{x})$ be the vector of early arrival covariates, with components as described in Table 4.5. Clearly, if $e_i = 0$, then we have $\mathcal{P}_\theta(i) = 0$. Further, note that if $C_\theta > e_i$, then $\mathcal{P}_\theta(i) = 0$, since we assume that $\tilde{t}_{ij} \geq c_{ij}$. Finally, if $\max_{j\in\theta\setminus\{i\}}\{e_j\} > e_i$, then we have again $\mathcal{P}_\theta(i) = 0$.

Table 4.5: Components of the early arrival covariates vector $\mathbf{w}_{i,\theta}(\mathbf{x})$ at customer $i \in \theta$ given feature vector $\mathbf{x}$

Predictor	Description
$x_1, \ldots, x_p$	Travel time covariates
e_i	Opening of time window
$a_\theta(i; \mathbf{g}(\mathbf{x}))$	Estimated arrival time at customer $i \in \theta$
$\hat{\sigma}^2_{\rho_\theta(i)i}$	Estimated variance in the travel time from $\rho_\theta(i)$ to i
$\max_{j\in\theta\setminus\{i\}}\{e_j\}$	Latest e_j along route θ before arriving at customer i
C_θ	Transportation cost of route θ up to customer i

Since our predicted quantity is a probability, a sensible learning model is a logistic regression. Let $S(z) = 1/(1 + \exp(-z))$ be the logistic function, and let $\mathcal{L}_{\text{nll}}(a, b) = -(b \log a + (1 - b) \log(1 - a))$ be the negative log likelihood loss function. For each customer $i \in \mathcal{V} \setminus \{0\}$, we train the parameters $\hat{\phi}^0_i$ and $\hat{\boldsymbol{\phi}}_i \in \mathbb{R}^{p+2}$ of the logit model: where we use $\mathbf{w}^k_{i,\theta} \coloneqq \mathbf{w}_{i,\theta}(\mathbf{x}^k)$, λ is the regularization parameter and $\|\boldsymbol{\phi}_i\|_1$ is the ℓ_1 norm of $\boldsymbol{\phi}_i$. Regularization by the ℓ_1 norm leads to sparsity in the model parameters.

Hence, for any route θ we estimate the early arrival probability at customer $i \in \theta$ by

$$\hat{\mathcal{P}}_\theta(i) = S(\hat{\phi}_i^0 + \hat{\boldsymbol{\phi}}_i^\top \mathbf{w}_{i,\theta}^{n+1}).$$

Appendix 4.B Dynamic programming algorithm for RCSP

We present a DP algorithm for obtaining a RCSP bound. In Algorithm 4.1, variable $T_1[\delta, i, q]$ stores a lower bound on the reduced cost of extending a route that ends at customer i with remaining capacity q, departing from i at time δ (i.e., the arrival time at customer i is equal to δ). We relax elementarity by allowing routes with cycles but we still remove 2-cycles. Similarly, $T_2[\delta, i, q]$ stores the second best lower bound on the reduced cost. Finally, $N[\delta, i, q]$ stores the customer following i on the route associated with the best lower bound.

Algorithm 4.1: Dynamic programming algorithm for RCSP

Result: matrix T_1 of lower bounds on the reduced costs of route extensions

$\ell_{\max} \leftarrow \max_{i \in \mathcal{V}^+}\{\ell_i\}$ // latest end of time window among all customers
$\Delta t \leftarrow \ell_{\max}/40$ // define a time step
for $\delta = 0, \Delta t, 2\Delta t, \ldots, \ell_{\max}$ **do**
  $T_1[\delta, i, q] \leftarrow \infty$, for $i \in \mathcal{V}^+$, $q = 1, \ldots, Q$ // initialize matrix T_1: lower bound on reduced costs
  $T_1[\delta, 0, q] \leftarrow 0$, for $q = 0, \ldots, Q$ // initialize matrix T_1
  $T_1[\delta, i, 0] \leftarrow c_{i0}$, for $i \in \mathcal{V}^+$ // initialize matrix B
  $T_2[\delta, i, q] \leftarrow \infty$, for $i \in \mathcal{V}$, $q = 0, \ldots, Q$ // initialize matrix T_2: second best cost
  $N[\delta, i, q] \leftarrow 0$, for $i \in \mathcal{V}$, $q = 0, \ldots, Q$ // initialize matrix N: next customer in the route
  for $q = 1, \ldots, Q$ **do**
    for $i \in \mathcal{V}^+$ **do**
      $T_1[\delta, i, q] \leftarrow T_1[\delta, i, q-1]$
      $T_2[\delta, i, q] \leftarrow T_2[\delta, i, q-1]$
      $N[\delta, i, q] \leftarrow N[\delta, i, q-1]$
      for $j \in \mathcal{V}^+$ **do**
        if $j = i$ *or* $q_j > q$ *or arc* (i, j) *is forbidden by branching* **then**
          continue // does not extend label L to customer j
        if $N[\delta, j, q - q_j] \neq i$ **then**
          $v \leftarrow c_{ij} - \gamma_j + T_1[\delta, j, q - q_j] + \pi(\delta - \ell_j)$ // xxx
        else
          $v \leftarrow c_{ij} - \gamma_j + T_2[\delta, j, q - q_j] + \pi(\delta - \ell_j)$ // avoid 2-cycles
        if $v < T_1[i, q]$ **then**
          $T_2[\delta, i, q] \leftarrow T_1[\delta, i, q]$ // move best to second best
          $T_1[\delta, i, q] \leftarrow v$ // set new best
          $N[\delta, i, q] \leftarrow j$ // set next customer
        else if $v < T_2[\delta, i, q]$ **then**
          $T_2[\delta, i, q] \leftarrow v$ // just update second best
return T_1

Appendix 4.C Proof of completion bounds

We provide proofs for the RCSP and knapsack bounds in the following.

Proof of Proposition 4.1. From the definition of the RCSP bound in Equation (4.21),

we have:

$$
\begin{aligned}
0 &\overset{(a)}{\leq} \overline{C}_\theta + \widehat{T}_{\text{RCSP}}(i, Q - q_\theta) \\
&\overset{(b)}{\leq} \overline{C}_\theta - c_{i0} + c_{iu_1} + \pi(\delta_{\theta\oplus u_1} - \ell_{u_1}) - \gamma_{u_1} + c_{u_1 0} + \widehat{T}_{\text{RCSP}}(u_1, Q - q_{\theta\oplus u_1}) \\
&\overset{(c)}{\leq} \overline{C}_\theta - c_{i0} + c_{iu_1} + \pi(\delta_{\theta\oplus u_1} - \ell_{u_1}) - \gamma_{u_1} + c_{u_1 0} \\
&\qquad - c_{u_1 0} + c_{u_1 u_2} + \pi(\delta_{\theta\oplus u_1 \oplus u_2} - \ell_{u_2}) - \gamma_{u_2} + c_{u_2 0} \\
&\qquad \dots \\
&\qquad - c_{u_{L-1} 0} + c_{u_{L-1} u_L} + \pi(\delta_{\theta\oplus\mathcal{E}} - \ell_{u_L}) - \gamma_{u_L} + c_{u_L 0} + \widehat{T}_{\text{RSCP}}(u_L, Q - q_{\theta\oplus\mathcal{E}}) \\
&\overset{(d)}{=} \overline{C}_\theta - c_{i0} + c_{iu_1} + \sum_{j=2}^{L} c_{u_{j-1}u_j} + c_{u_L 0} + \sum_{j=1}^{L} \Big(\pi(\delta_{\theta\oplus u_1 \oplus \dots \oplus u_j} - \ell_{u_j}) - \gamma_{u_j} \Big) + \widehat{T}_{\text{RSCP}}(u_L, Q - q_{\theta'}) \\
&\overset{(e)}{\leq} \overline{C}_\theta - c_{i0} + c_{iu_1} + \sum_{j=2}^{L} c_{u_{j-1}u_j} + c_{u_L 0} + \sum_{j=1}^{L} \Big(\sum_{\omega\in\Omega} \alpha^\omega \cdot \pi(a_{\theta'}(u_j; \mathbf{t}^\omega) - \ell_{u_j}) - \gamma_{u_j} \Big) + \widehat{T}_{\text{RSCP}}(u_L, Q - q_{\theta'}) \\
&\overset{(f)}{=} \overline{C}_{\theta'} + \widehat{T}_{\text{RSCP}}(u_L, Q - q_{\theta'}) \overset{(g)}{\leq} \overline{C}_{\theta'}.
\end{aligned}
$$

Inequality (b) follows from the fact that extending path θ to customer u_1 cannot lead to a smaller bound than the bound associated with the optimal path extension from the minimization operator in Equation (4.21). In Inequality (c), the same argument holds when extending path $\theta \oplus u_1$ to customers $u_2, \dots, u_j$. In Equation (d) we rearrange the terms, and Inequality (e) is due to the fact that, given a path θ ending at customer j:

$$\delta_\theta \leq \tau_\theta \leq a_\theta(j; \mathbf{t}^\omega), \quad \forall \omega \in \Omega \tag{4.44}$$

Finally, Equality (f) is due to the resource extension function for the reduced cost given by Equation (4.18), and Inequality (g) holds since going from u_L back to the depot incurs no additional cost and can not improve the completion bound. □

Proof of Proposition 4.2. From the resource extension function given by Equation (4.18), we have:

$$
\begin{aligned}
\overline{C}_{\theta'} &\overset{(a)}{=} \overline{C}_\theta - c_{i0} + c_{iu_1} + \sum_{j=2}^{L} c_{u_{j-1}u_j} + c_{u_L 0} + \sum_{j=1}^{L} \left(\sum_{\omega\in\Omega} \alpha^\omega \cdot \pi(a_{\theta'}(u_j; \mathbf{t}^\omega) - \ell_{u_j}) - \gamma_{u_j} \right) \\
&\overset{(b)}{\geq} \overline{C}_\theta + \sum_{j=1}^{L} \left(\sum_{\omega\in\Omega} \alpha^\omega \cdot \pi(a_{\theta'}(u_j; \mathbf{t}^\omega) - \ell_{u_j}) - \gamma_{u_j} \right) \\
&\overset{(c)}{\geq} \overline{C}_\theta + \sum_{j=1}^{L} \Big(\pi(\tau_\theta + \min_{\omega\in\Omega} t^\omega_{iu_j} - \ell_{u_j}) - \gamma_{u_j} \Big) \\
&\overset{(d)}{=} \overline{C}_\theta + \sum_{l\in\mathcal{V}\setminus\{0\}} -v_{il}(\theta)\, z^*_l \overset{(e)}{\geq} \overline{C}_\theta + \widehat{T}_{ks}(i, Q - q_\theta) \geq 0
\end{aligned}
$$

where Inequality (b) is due to the triangle inequality, which implies that the cost of a route cannot decrease if we add customers to it. Inequality (c) is a consequence of Equation (4.23) and the fact that adding more customers to a route between i and u_j can only increase the arrival time at customer u_j. Equality (d) holds by our definition of the knapsack values. Inequality (e) is due to the optimality of the knapsack solution and the definition of the completion bound. □

Appendix 4.D Detailed results for instances with 25 customers

The BP&C method could find optimal solutions for all instances with 25 customers within the predetermined time limit. In the following, we analyze the test cost results under the different generative models of travel times.

Linear generative model. Table 4.6 reports the test costs of different data-driven approaches for instances with 25 customers and a linear generative model. Among the practical models, we highlight with boldface numbers the models that achieve the lowest cost on each instance. For PTO-F and "Full" models, we highlight the results in boldface whenever one of the practical models achieves equal or lower test cost. CSAA achieves the lowest test costs among the practical models for most instances. For C-type instances, most models achieve the full information lower bound.

Exponential generative model. Table 4.7 reports the test costs for instances with 25 customers and an exponential generative model. We observe higher deviations from the lower bound in this setting than in the previous results under the linear generative model. On average, CSAA is superior to the other models. We also note that P-NN is superior on a number of instances. However, the lower bound is achieved only in a single instance, illustrating the greater difficulty of solving the conditional stochastic VRPTW under a nonlinear generative model.

Sigmoidal generative model. Table 4.8 reports test costs for instances with 25 customers and a sigmoidal generative model. As before, on average, CSAA is superior to the other models. For some instances, CSAA is outperformed by SAA. In this setting, SAA is superior to PTO and P-NN, in contrast to the results with linear and exponential generative models.

Comparison of generative models. Table 4.9 compares the average test costs of all considered approaches under the different generative models of travel times. Considering

the practical models, CSAA achieves the lowest test costs on average. PTO-F is superior to CSAA for the exponential and sigmoidal instances, indicating that the PTO framework could be the best approach in those settings, provided that one can train a model that can predict the conditional expected travel times perfectly. Further investigation is needed to assess how well the PTO framework can perform in practice under different predictive models. However, in all settings, the average test cost performance of CSAA is close to that of PTO-F, meaning that CSAA would be on par with PTO-F even in the unlikely case in which one could predict the true conditional expected travel times.

Table 4.6: Test cost results for instances with linear generative model and 25 customers

Instance	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	P-NN	PTO-F	Full (Abs.)
R101	2.88	1.48	1.03	1.63	1.06	0.09	**0.05**	1.68	1.03	649.6
R102	2.13	2.46	**0.02**	1.27	1.04	**0.02**	**0.02**	0.56	0.02	576.5
R103	8.62	0.17	2.69	2.77	**0.00**	0.06	0.09	0.09	0.00	465.4
R104	13.98	2.94	1.29	5.76	0.34	**0.00**	**0.00**	0.02	0.8	435.6
R105	10.5	3.4	0.28	8.13	2.51	**0.06**	**0.06**	0.07	0.28	544.9
R106	75.0	**0.00**	1.89	42.85	8.32	**0.00**	0.84	2.67	5.85	502.4
R107	0.16	0.16	0.25	0.25	0.91	**0.00**	**0.00**	0.3	0.25	437.6
R108	5.34	6.12	0.9	3.37	0.9	0.17	**0.05**	0.34	0.85	411.9
R109	1.34	1.98	**0.02**	0.75	0.77	**0.02**	**0.02**	**0.02**	0.02	455.0
R110	22.39	4.51	0.63	4.03	2.2	**0.04**	**0.04**	0.17	0.41	458.7
R111	108.97	12.23	0.89	31.46	27.09	**0.13**	**0.13**	1.02	1.44	450.4
R112	29.73	5.17	1.73	29.73	5.87	0.12	**0.02**	0.1	0.9	410.3
C101	**0.00**	**0.00**	**0.00**	**0.00**	3.1	**0.00**	**0.00**	**0.00**	0.00	196.5
C102	**0.00**	**0.00**	**0.00**	**0.00**	0.77	**0.00**	**0.00**	0.92	0.00	195.8
C103	-	-	-	-	-	-	-	-	-	-
C104	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.21	0.00	187.5
C105	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00	191.8
C106	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00	194.3
C107	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00	191.8
C108	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00	191.8
C109	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00	191.8
RC101	30.89	1.73	5.96	45.5	12.68	**0.02**	0.04	4.03	5.7	533.2
RC102	56.1	17.4	1.43	42.96	2.68	0.08	**0.03**	0.54	1.94	392.5
RC103	15.55	15.17	2.67	12.67	0.12	**0.03**	**0.03**	1.74	2.82	344.0
RC104	2.88	0.94	2.88	5.73	1.72	**0.38**	0.44	2.66	0.63	319.3
RC105	2.67	1.24	1.38	1.15	**0.48**	0.55	0.55	1.34	0.6	419.1
RC106	16.89	7.54	5.41	10.76	6.31	1.86	**1.65**	6.79	2.48	375.4
RC107	10.93	1.3	3.97	9.33	0.91	0.29	**0.16**	3.09	2.63	307.5
RC108	2.68	2.68	**0.07**	3.74	0.26	0.13	0.13	0.33	0.07	302.3
Average	14.99	3.16	1.26	9.42	2.86	**0.14**	0.16	1.02	1.03	369.03

Table 4.7: Test cost results for instances with exponential generative model and 25 customers

Instance	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	P-NN	PTO-F	Full (Abs.)
R101	8.3	4.08	**0.33**	3.29	2.49	0.7	0.52	**0.33**	0.24	1259.2
R102	8.21	11.71	2.74	5.81	5.13	**1.22**	1.33	1.99	0.92	779.6
R103	34.77	8.01	4.05	15.32	6.94	1.32	**1.03**	2.93	1.7	522.9
R104	14.99	3.11	7.26	13.58	8.48	3.74	**2.76**	4.5	3.01	510.9
R105	51.75	25.06	4.98	10.67	9.76	**4.11**	4.15	10.64	1.99	804.5
R106	7.61	8.06	1.3	2.56	2.27	1.91	1.19	**0.54**	1.08	538.6
R107	356.78	46.89	3.43	20.96	6.54	0.86	**0.34**	0.6	3.51	498.2
R108	34.38	10.64	3.67	12.74	4.2	**0.49**	0.79	1.2	3.67	466.3
R109	103.98	21.15	5.13	14.67	2.87	**2.73**	3.75	2.78	2.46	557.5
R110	25.27	4.3	4.32	5.36	1.96	**1.57**	1.98	1.63	2.14	509.4
R111	31.54	23.4	1.43	30.21	12.38	1.26	**0.99**	1.35	1.65	586.2
R112	103.67	14.45	5.89	53.55	13.41	6.93	6.0	**1.93**	3.83	460.1
C101	176.85	8.42	47.02	178.59	39.88	4.33	5.6	**1.66**	9.31	258.8
C102	33.29	6.4	5.75	20.71	4.32	1.21	**0.86**	2.51	8.26	231.3
C103	20.52	**0.35**	12.53	11.43	4.49	1.95	0.9	1.75	6.54	200.3
C104	-	-	-	-	-	-	-	-	-	-
C105	104.16	13.72	26.16	104.16	13.16	13.34	17.5	**8.32**	35.03	233.2
C106	19.29	2.05	9.0	22.45	5.39	**0.84**	2.05	2.16	3.49	263.3
C107	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**	0.78	192.8
C108	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	**0.10**	0.10	192.7
C109	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00	191.8
RC101	91.45	78.9	20.86	25.11	26.11	22.34	**20.85**	20.93	2.67	674.5
RC102	126.85	39.67	16.3	119.7	4.66	4.32	4.45	**2.88**	2.02	573.5
RC103	42.63	6.33	9.85	14.79	**4.94**	5.81	5.81	11.14	5.37	497.5
RC104	53.44	19.32	**4.91**	7.56	14.94	5.92	5.87	5.75	6.95	415.6
RC105	15.94	5.85	4.62	15.31	4.5	**2.44**	2.71	4.84	5.76	553.8
RC106	350.2	21.77	10.36	27.22	6.61	**2.07**	5.06	7.42	5.55	537.4
RC107	11.18	**5.46**	26.13	11.67	5.87	10.59	22.0	21.92	10.63	470.4
RC108	106.85	20.28	12.87	53.0	16.62	**8.54**	9.9	9.29	9.27	397.0
Average	69.1	14.65	8.99	28.62	8.17	**3.98**	4.62	4.71	4.93	477.76

Table 4.8: Test cost results for instances with sigmoidal generative model and 25 customers

Instance	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	P-NN	PTO-F	Full (Abs.)
R101	14.06	11.3	13.04	15.14	10.2	**6.31**	7.34	11.27	8.57	1100.6
R102	123.88	30.19	25.34	79.38	39.03	21.92	**14.00**	16.09	9.44	926.2
R103	111.72	19.45	26.71	65.46	12.78	10.55	**9.26**	19.47	13.97	529.0
R104	102.4	16.32	38.33	61.99	27.79	17.44	**14.36**	27.87	21.28	500.1
R105	163.67	27.7	35.37	37.17	21.83	26.02	**18.68**	25.07	12.17	702.8
R106	55.21	12.21	56.77	49.9	12.03	**10.34**	11.69	25.38	14.79	572.5
R107	106.36	**13.47**	91.1	76.81	16.12	15.57	13.74	36.45	28.34	514.4
R108	118.65	**8.65**	40.72	86.72	13.81	17.15	11.27	24.16	15.58	473.0
R109	81.88	53.94	54.22	106.62	30.46	**11.85**	15.78	31.13	21.04	567.9
R110	180.97	12.7	20.84	42.72	11.09	**8.16**	10.46	13.74	18.81	508.7
R111	24.16	11.01	20.08	20.39	14.9	**7.82**	8.13	16.33	14.68	495.9
R112	88.22	12.28	46.23	63.7	17.31	**11.27**	19.93	29.36	24.87	455.1
C101	189.53	**12.97**	137.29	67.08	14.62	21.97	18.33	39.35	127.22	272.2
C102	35.91	**17.81**	53.93	44.82	19.52	**17.81**	19.77	32.02	69.02	239.2
C103	79.54	10.61	53.04	63.37	15.79	**2.62**	7.42	8.56	29.73	210.2
C104	-	-	-	-	-	-	-	-	-	-
C105	191.87	**14.78**	181.6	186.98	136.84	132.73	134.21	145.46	154.58	243.5
C106	10.33	8.43	49.47	38.37	**7.88**	16.09	23.27	33.13	91.13	272.9
C107	101.25	101.2	**43.13**	101.25	101.34	59.64	59.64	**43.13**	101.25	216.3
C108	45.79	45.79	45.79	45.79	**39.76**	45.79	45.79	45.79	45.79	204.2
C109	**31.37**	**31.37**	**31.37**	**31.37**	**31.37**	**31.37**	**31.37**	**31.37**	31.37	191.9
RC101	52.94	46.07	60.23	61.72	35.21	37.71	35.0	**31.80**	7.6	783.1
RC102	90.07	**19.06**	105.58	92.49	24.62	51.48	39.6	44.91	64.77	558.9
RC103	175.45	**25.33**	64.98	87.81	33.82	33.37	39.18	53.55	36.04	448.9
RC104	223.16	49.96	70.28	96.86	36.56	**29.07**	39.9	60.37	62.82	404.5
RC105	285.53	**21.61**	86.97	162.95	31.7	25.52	40.26	46.15	47.67	571.6
RC106	320.05	27.7	83.45	66.89	17.76	**15.04**	24.95	36.59	64.34	471.4
RC107	432.47	**29.59**	111.35	245.93	40.74	37.07	**29.59**	61.34	47.04	485.3
RC108	285.87	**29.62**	129.35	149.61	54.36	32.88	55.46	84.66	68.14	410.6
Average	132.94	**25.75**	63.45	80.33	31.04	26.95	28.51	38.38	44.72	476.1

Table 4.9: Average test costs and percentage gaps to the full-knowledge model for instances with 25 customers

	Gen. model	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	P-NN	PTO-F	Full
$\hat{R}$	Linear	436.0	382.1	374.4	411.0	381.8	**369.6**	369.7	373.3	373.6	369.0
	Exponential	809.5	559.0	515.5	596.6	515.8	**497.3**	499.8	502.0	496.5	477.8
	Sigmoidal	1102.7	593.2	751.4	844.7	607.1	**588.9**	592.7	638.9	642.0	476.1
Gap(%)	Linear	18.16	3.55	1.46	11.38	3.47	**0.16**	0.17	1.17	1.25	0.00
	Exponential	69.42	16.99	7.89	24.86	7.95	**4.08**	4.61	5.06	3.91	0.00
	Sigmoidal	131.61	24.60	57.82	77.42	27.52	**23.69**	24.49	34.19	34.85	0.00

Appendix 4.E Detailed results for instances with 50 customers

With 50 customers, we could not solve all instances to optimality. To conduct a fair comparison between the models, we consider in the following analyses only instances for which all models could find optimal solutions or integer feasible solutions with an optimality gap of at most 0.01.

Linear generative model. Table 4.10 shows test cost results for 22 instances that could be solved to an optimality gap of at most 1%. Among the practical models, CSAA achieves the lowest test costs in all instances. The full-information lower bound is achieved in 5 out of the 22 instances. Notably, the test costs of CSAA are smaller than or equal to those of PTO-F in all instances.

Exponential generative model. Table 4.11 reports the test costs of different data-driven approaches for 24 instances that could be solved to an optimality gap of at most 1%. For most instances, CSAA has the lowest test costs among the practical models. Moreover, CSAA is also superior to PTO-F, on average.

Sigmoidal generative model. Table 4.12 reports the test cost of different data-driven approaches for 21 instances that could be solved to an optimality gap of at most 1%. As before, on average, CSAA is superior to the other practical models. However, we observe a larger gap to the full-information lower bound.

Comparison of generative models. We compare the average test costs achieved by the different methods across the different generative models in Table 4.13. For linear and exponential travel times, CSAA has the lowest average test costs. For sigmoidal travel times, we observe significantly larger gaps. In this setting, CSAA has a full-information gap of 32.93%, close to PTO-F, with 32.20%.

Table 4.10: Test cost results for instances with linear generative model and 50 customers

Instance	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	PTO-F	Full
R101	25.36	5.84	1.11	10.97	3.48	0.7	**0.66**	0.97	1237.0
R102	18.32	3.72	0.99	16.39	1.44	**0.16**	0.21	1.01	1155.8
R103	6.96	2.91	3.05	4.53	1.99	**0.27**	0.42	1.29	850.2
R104	-	-	-	-	-	-	-	-	-
R105	51.25	6.4	2.44	23.05	5.86	0.67	**0.28**	1.28	994.4
R106	-	-	-	-	-	-	-	-	-
R107	7.52	2.12	1.81	16.82	2.63	0.3	**0.18**	2.12	772.8
R108	-	-	-	-	-	-	-	-	-
R109	7.64	3.98	4.16	7.75	0.52	**0.23**	0.24	1.81	828.7
R110	35.0	3.22	1.45	14.33	5.92	0.44	**0.22**	0.99	758.5
R111	32.81	3.6	2.26	19.8	6.85	0.12	**0.08**	1.58	738.5
R112	-	-	-	-	-	-	-	-	-
C101	1.66	21.32	1.66	1.66	3.09	**0.00**	**0.00**	1.66	379.0
C102	4.7	4.7	**0.05**	4.7	1.55	**0.05**	**0.05**	**0.05**	367.8
C103	-	-	-	-	-	-	-	-	-
C104	-	-	-	-	-	-	-	-	-
C105	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	363.2
C106	1.52	9.57	1.52	1.52	1.52	**0.95**	**0.95**	1.52	387.5
C107	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	363.2
C108	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	363.2
C109	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	363.2
RC101	54.67	6.1	5.9	16.93	6.23	1.28	**1.09**	4.91	1130.4
RC102	93.37	9.48	3.92	24.37	6.37	**0.75**	0.81	2.4	987.5
RC103	45.81	5.42	4.35	22.79	11.45	**0.10**	0.46	2.13	774.5
RC104	-	-	-	-	-	-	-	-	-
RC105	47.79	2.21	7.64	40.57	17.12	0.6	**0.44**	4.59	917.1
RC106	27.19	5.22	4.55	17.09	3.17	0.91	**0.65**	4.77	879.3
RC107	56.66	5.72	1.82	18.36	2.18	0.26	**0.09**	2.71	692.7
RC108	-	-	-	-	-	-	-	-	-
Average	24.68	4.83	2.32	12.46	3.87	0.37	**0.33**	1.7	728.79

Table 4.11: Test cost results for instances with exponential generative model and 50 customers

Instance	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	PTO-F	Full
R101	10.48	10.82	3.14	9.02	4.51	**1.75**	2.29	1.63	1466.0
R102	45.24	15.37	3.4	8.94	6.75	2.81	**2.66**	1.28	1821.7
R103	20.0	18.88	5.54	11.45	9.37	3.93	**3.92**	3.05	1679.8
R104	31.36	8.07	21.56	18.97	**3.59**	8.09	13.81	6.43	794.4
R105	4.76	4.33	7.89	3.59	**3.01**	4.72	5.33	3.36	1252.7
R106	-	-	-	-	-	-	-	-	-
R107	198.4	23.5	13.08	34.34	11.52	**4.12**	4.6	5.21	907.8
R108	-	-	-	-	-	-	-	-	-
R109	24.21	11.09	6.54	7.95	5.43	**2.44**	3.23	5.33	975.9
R110	-	-	-	-	-	-	-	-	-
R111	158.52	16.74	4.38	21.72	4.78	**3.78**	3.92	6.69	887.8
R112	14.73	7.8	5.37	24.43	6.52	3.45	**3.23**	6.15	756.5
C101	4.79	1.85	6.97	4.9	2.48	**1.65**	1.69	7.05	492.3
C102	5.7	**0.13**	2.74	2.74	0.99	0.65	1.12	2.41	473.7
C103	-	-	-	-	-	-	-	-	-
C104	-	-	-	-	-	-	-	-	-
C105	124.2	6.62	12.26	70.73	5.23	**3.79**	4.06	12.74	438.0
C106	66.09	7.29	7.97	26.75	6.51	9.75	**5.75**	7.49	500.5
C107	113.86	60.36	3.89	113.86	7.64	**2.64**	**2.64**	3.89	393.8
C108	**0.14**	**0.14**	0.63	**0.14**	**0.14**	**0.14**	0.22	0.69	364.0
C109	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	363.2
RC101	120.37	44.11	5.5	18.79	64.71	**3.37**	4.64	2.06	1663.0
RC102	96.18	49.45	14.25	24.92	25.28	**7.31**	9.02	3.93	1605.4
RC103	44.07	15.06	7.25	18.17	6.69	**2.51**	4.28	5.2	1029.9
RC104	-	-	-	-	-	-	-	-	-
RC105	39.43	17.38	19.16	14.49	8.52	**4.09**	10.17	5.56	1235.6
RC106	109.79	55.04	13.84	46.25	35.6	**9.23**	13.02	4.87	1181.6
RC107	58.82	**6.72**	98.41	32.39	7.36	12.33	20.5	9.71	928.5
RC108	-	-	-	-	-	-	-	-	-
Average	58.69	17.31	11.99	23.39	10.3	**4.21**	5.46	4.76	964.19

Table 4.12: Test cost results for instances with sigmoidal generative model and 50 customers

Instance	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	PTO-F	Full
R101	87.09	31.48	43.26	54.85	27.75	**21.85**	29.31	8.52	2179.0
R102	131.73	53.93	53.77	69.1	53.27	**29.65**	30.23	18.79	1732.0
R103	241.33	46.09	76.05	90.64	**26.26**	26.61	35.35	17.02	1627.7
R104	-	-	-	-	-	-	-	-	-
R105	303.68	23.6	32.32	104.4	**16.23**	**16.23**	17.99	25.09	1278.4
R106	252.94	43.82	106.23	107.35	42.77	28.46	**22.93**	50.4	1021.1
R107	233.67	32.1	92.73	81.15	31.91	**23.07**	25.51	37.18	940.9
R108	-	-	-	-	-	-	-	-	-
R109	261.41	**16.81**	72.51	156.4	26.91	19.46	22.79	41.33	978.7
R110	167.15	**18.02**	80.05	107.21	35.25	26.35	23.18	44.34	862.7
R111	164.21	26.28	80.91	115.21	31.36	**23.87**	30.57	44.54	873.8
R112	170.69	28.15	83.54	144.91	37.41	29.05	**15.94**	33.91	795.3
C101	186.14	**16.02**	59.32	62.29	25.45	49.13	29.6	63.04	532.5
C102	128.54	**16.09**	108.06	124.41	24.31	21.4	24.45	48.58	506.0
C103	-	-	-	-	-	-	-	-	-
C104	-	-	-	-	-	-	-	-	-
C105	-	-	-	-	-	-	-	-	-
C106	-	-	-	-	-	-	-	-	-
C107	52.86	52.86	52.68	52.86	52.46	**52.26**	52.68	52.86	402.4
C108	-	-	-	-	-	-	-	-	-
C109	12.25	12.25	12.25	**12.00**	12.8	**12.00**	12.25	12.74	364.1
RC101	228.78	**33.32**	89.05	80.05	36.36	46.11	53.35	9.24	2619.1
RC102	292.43	97.24	88.67	168.9	79.34	**57.24**	68.13	41.78	1337.6
RC103	509.92	80.08	143.89	143.32	77.7	**32.20**	37.6	47.19	1021.2
RC104	-	-	-	-	-	-	-	-	-
RC105	262.02	**36.06**	84.24	294.37	76.76	50.57	51.29	27.66	1422.8
RC106	203.06	44.88	90.33	90.65	**35.20**	41.64	39.01	72.9	1100.6
RC107	477.15	**21.88**	149.12	237.37	36.3	34.21	26.74	54.2	982.0
RC108	-	-	-	-	-	-	-	-	-
Average	218.35	36.55	79.95	114.87	39.29	**32.07**	32.44	37.57	1128.9

Table 4.13: Average test costs and percentage gaps to the full-knowledge model for instances with 50 customers

	Gen. model	D-avg	SAA	PTO-OLS	PTO-kNN	SAA-kNN	CSAA	RSAA	PTO-F	Full
$\hat{R}$	Linear	953.4	764.5	749.1	839.7	762.1	732.1	**731.7**	743.5	728.8
	Exponential	1549.8	1160.9	1082.4	1159.9	1096.5	**1007.3**	1021.7	1006.1	964.2
	Sigmoidal	3690.0	1570.5	2029.8	2440.4	1588.8	**1500.7**	1524.4	1492.4	1128.9
Gap(%)	Linear	30.83	4.90	2.78	15.22	4.57	0.46	**0.40**	2.02	0.00
	Exponential	60.73	20.40	12.26	20.30	13.72	**4.47**	5.96	4.34	0.00
	Sigmoidal	226.87	39.12	79.80	116.18	40.73	**32.93**	35.04	32.20	0.00

5 Optimizing ride-hailing with a mix of on-demand and pre-booked customers under distributional shift

Abstract

We consider a mixed-service ride-hailing system that offers customers the option to request a ride on demand or to pre-book it in advance. For time-sensitive customers, pre-booking provides a service guarantee at a price premium. From the operator's perspective, pre-booking allows for planning ahead with higher certainty but may incur the duty to operate unfavorable trips that may even induce a shift in the demand distribution, e.g., in low-demand suburban neighborhoods. Against this background, we develop an optimization framework for a mixed-service ride-hailing system, allowing us to study the trade-offs between higher planning certainty and the rise of unfavorable rides due to shifts in the demand distribution. We propose a two-stage stochastic optimization formulation in which the first-stage problem consists of deciding which pre-booking requests to accept, while the second-stage problem involves assigning vehicles to requests and planning routes with uncertain on-demand requests. We present a sample average approximation (SAA) formulation and develop a scalable solution algorithm that approximates the second-stage subproblems using a polynomial-time algorithm. We conduct experiments based on the New York City network using historical yellow taxi trip data. We show that greedily accepting all pre-booking requests leads to a 14.5% reduction in the operator's profit compared to a purely on-demand baseline in environments with strong distributional shifts. In contrast, the proposed SAA solutions lead to profit increases ranging from 6.5% to 7.7%, in settings with weak and strong distributional shifts, respectively, while satisfying customer pick-up time windows. Additionally, we provide managerial insights to assist system operators in making informed design choices to counteract the effects of distributional shifts.

This chapter is based on the working paper:

Serrano B., Jacquillat A., Minner S., Schiffer M. (2024). Optimizing ride-hailing with a mix of on-demand and pre-booked customers under distributional shift.

5.1 Introduction

Over the past decades, technological advances, such as the widespread use of smartphones, precise global positioning systems, and accessible wireless Internet via mobile networks, have fueled the rise of ride-hailing platforms offering on-demand passenger transportation. Although users can often secure a ride within minutes in densely populated urban areas, driver supply can vary, e.g., based on the pick-up location and time of day, occasionally leading to longer wait times or a lack of available drivers. In this context, ride-hailing platforms such as Uber, Lyft, and Bolt have recently started to offer pre-booking services (Bolt 2023; Lyft 2023; Uber 2023) especially targeting time-sensitive customers heading to airports or on holiday travels. Customers pay an additional pre-booking fee to reserve a ride in advance, and the operator commits to sending a driver to the agreed pick-up location at the specified time. In line with the terminology from the literature (see, e.g., Abkarian, Mahmassani, and Hyland 2022; Engelhardt, Dandl, and Bogenberger 2022), we refer to such a system as a mixed-service system.

From an operational perspective, pre-booked rides offer higher planning certainty for the operator, as the travel demand is known in advance. However, the commitment to serving pre-booked customers may force the operator to reject potentially more profitable on-demand requests if the driver supply is insufficient to meet overall demand. Additionally, introducing a pre-booking service may induce travel demand in areas traditionally experiencing low driver availability, such as low-demand suburban neighborhoods, leading to a shift in the travel demand distribution, i.e., creating a mismatch between the trip distribution of pre-booked and on-demand ride requests. Consequently, implementing a pre-booking service introduces a trade-off between higher planning certainty and the rise of unfavorable rides due to shifts in the demand distribution.

Despite the extensive literature on optimizing ride-hailing systems, few studies have focused on mixed-service systems, particularly regarding shifts in the travel demand distribution. To address this research gap, we first illustrate the complexity of the problem by introducing a simplified queuing-theoretical model, having as few degrees of freedom as possible, while still capturing the main aspects of a mixed-service system. This illustrative analysis motivates the need for an optimization framework that more accurately represents the temporal and spatial details of the problem. Accordingly, we propose a novel two-stage stochastic optimization model that maximizes the sum of pre-booking fees and the expected total profit from serving pre-booked and on-demand customers, considering driving costs. Specifically, in the first stage, the operator decides to accept or reject incoming pre-booking requests, and in the second stage, assigns vehicles to requests and plans routes in response to incoming on-demand requests. Following a data-driven approach, we present a sample average approximation (SAA) formulation based on sce-

narios describing possible realizations of on-demand requests. However, solving the SAA model is challenging as it requires the solution of integer subproblems that must satisfy linking constraints with respect to first-stage decisions, leading to a computationally intractable problem. To address these challenges, we develop a scalable heuristic algorithm that solves approximate versions of the second-stage subproblems, modeled as K-disjoint shortest path problems (K-DSPPs). We conduct experiments based on New York City yellow taxi data and derive managerial insights indicating that a mixed-service system can lead to increased profit compared to the purely on-demand baseline system, even in environments with strong distributional shifts.

5.1.1 Related Works

Our work relates to the literature on optimizing ride-hailing systems, often modeled as a taxi routing problem. In this section, we first review related works in the field of ride-hailing systems, before we discuss related works that also investigate the concept of pre-booking within this setting. For a general overview of vehicle routing problems (VRPs), we refer the interested reader to Vidal, Laporte, and Matl (2020). Finally, we discuss papers that model transportation problems using queuing-theoretical models.

Optimizing ride-hailing systems: Previous works have modeled ride-hailing as a variant of the pick-up and delivery problem (PDP), in which vehicles transport goods from a pick-up location to a drop-off location. Similarly, the dial-a-ride problem (DARP) considers vehicles that transport customers instead of products. Berbeglia, Cordeau, and Laporte (2010) and Cordeau and Laporte (2007) surveyed works on the dynamic versions of the PDP and the DARP. Bertsimas, Jaillet, and Martin (2019) proposed an online re-optimization method that iteratively assigns incoming ride requests to vehicles by solving an offline optimization model in a rolling-horizon fashion.

The possibility of customers sharing rides also attracted research interest in the literature. For a review on dynamic ride-sharing, we refer to Agatz et al. (2012). Alonso-Mora et al. (2017) proposed an algorithm based on model-predictive control for optimizing a ride-sharing system. Soza-Parra, Kucharski, and Cats (2024) investigated how different travel demand patterns affect the shareability of a ride-pooling system, a metric that measures the extent to which different customers can share rides, taking into account the rides' compatibility in time and space, and customer preferences.

The emergence of autonomous vehicles has inspired new business models for passenger transportation. In particular, an autonomous mobility-on-demand (AMoD) system consists of a centrally-operated fleet of autonomous vehicles serving customers on demand (cf. Pavone 2015). For a general overview of methods for the analysis and control of

AMoD systems, we refer to Zardini et al. (2022). Rossi et al. (2018) proposed a network flow framework for controlling an AMoD system while also considering the effects of congestion. Tsao, Iglesias, and Pavone (2018) proposed a stochastic model predictive control algorithm that incorporates probabilistic forecasts of future requests for dispatching and rebalancing in AMoD systems. Jungel et al. (2023) proposed a hybrid combinatorial optimization method combined with machine learning for controlling an AMoD fleet. The authors learn control policies for dispatching vehicles to serve ride requests and for rebalancing idle vehicles in anticipation of future demand. Other works focused explicitly on the rebalancing of idle vehicles (see, e.g., Iglesias et al. 2018; Liu and Samaranayake 2020; Pavone et al. 2012; Zhang and Pavone 2016).

A large body of literature focused on applying reinforcement learning (RL) for controlling (autonomous) mobility-on-demand systems. Qin, Zhu, and Ye (2022) surveyed RL methods applied to ride-hailing and ride-sharing systems. Zhou et al. (2019) proposed a multi-agent RL method that extends a deep Q-learning network with Kullback-Leibler divergence optimization. Tang et al. (2019) proposed a novel RL method for vehicle dispatching and conducted offline simulations using data from DiDi. Sadeghi Eshkevari et al. (2022) described an RL method deployed by DiDi that dynamically solves a maximum bipartite matching problem using a state value function. Enders et al. (2023) and Hoppe et al. (2024) proposed a hybrid algorithm that uses multi-agent soft actor-critic to parameterize a weighted bipartite matching problem. Further works focused on the topic of fleet rebalancing, such as Gammelli et al. (2021), Jiao et al. (2021), and Skordilis et al. (2022), and Liang et al. (2022), among others.

Our work relates to the partially dynamic VRP, where part of the requests are known to the operator before the day of operation begins, while the remaining requests are revealed in an online fashion. Lund, Madsen, and Rygaard (1996) introduced the concept of *degree of dynamism*, defined as the number of dynamic requests divided by the total number of requests that enter the system during the whole time horizon. They proposed heuristic methods adapted from the insertion heuristic by Solomon (1987). Larsen, Madsen, and Solomon (2002) introduced the *effective degree of dynamism*, which extends the degree of dynamism by taking into account the requests' pick-up times.

Pre-booking in shared mobility systems: Ride-hailing systems offering pre-booking services are often denoted as reservation-based systems in the literature. Bilali et al. (2019) proposed an analytical model to study the influence of ride requests' reservation times in a ride-sharing system. However, the reservation times considered in their experiments are rather short, between 2 to 15 minutes, such that requests may still be considered on-demand. In contrast, we consider that pre-booking requests enter the system at least one day before operation.

Only a few papers have considered a mix of pre-booking and on-demand requests. Engelhardt, Dandl, and Bogenberger (2022) studied a mixed-service ride-pooling system, and proposed a solution algorithm that first provides an offline solution for pre-booked requests and then accommodates on-demand customers in an online fashion by adopting a framework based on Alonso-Mora et al. (2017). Engelhardt, Dandl, and Bogenberger (2022) showed that on-demand customers also benefit from decreased waiting and detour times when pre-booking is enabled. Their experimental setup assumes that on-demand and pre-booking requests follow the same distribution. In contrast, our work analyzes the impact of distributional shifts on the system performance, focusing on pre-booking acceptance decisions.

According to the review paper by Narayanan, Chaniotakis, and Antoniou (2020), pre-booking has been studied in the context of shared autonomous vehicle services by Lamotte, De Palma, and Geroliminis (2017), Levin (2017), and Ma et al. (2017), and Pimenta et al. (2017). However, most works considered only pre-booking requests, not a mix of pre-booking and on-demand requests. Duan et al. (2020) studied a ride-hailing system in which autonomous vehicles serve a mix of pre-booking and on-demand requests. Their framework consists of a centralized dispatcher that assigns short-term requests to vehicles, while each vehicle manages its own long-term route, using a heuristic method to respond to incoming long-term requests. Abkarian, Mahmassani, and Hyland (2022) studied a mixed-service system of autonomous vehicles, combining on-demand passenger transportation and carsharing, where customers can either rent a vehicle for a certain time slot or pre-book a ride specifying their origin and destination locations. The authors developed a dynamic simulation framework and proposed re-optimization methods for assigning vehicles to requests. In contrast to our work, they assumed that the operator cannot reject ride requests and that customers are willing to wait indefinitely for their ride.

Related to our work is the paper by Elting and Ehmke (2021), who considered the option of pre-booking a ride in a dynamic ride-sharing system, modeled as a variant of a dial-a-ride problem. They studied how the degree of dynamism (cf. Lund, Madsen, and Rygaard 1996) affects the system performance, e.g., in terms of rejection and occupancy rates.

Queuing models for shared mobility systems: There is extensive literature on using queuing models to analyze systems with customer or service differentiation. For an introduction to the topic, we refer to Hassin and Haviv (2003). Afèche (2013) considered a firm offering two service classes that are differentiated in prices and delays, e.g., corresponding to wait times or lead times. The authors adopted a single-server queuing system to model the firm's problem of designing revenue-maximizing price and lead-time menus and scheduling policies. Maglaras, Yao, and Zeevi (2017) extended the framework

of Afèche (2013) to a setting with multiple service classes using a multi-server system.

Recent literature has proposed the use of queuing models to analyze transportation systems. In particular, Braverman et al. (2019) modeled a ride-hailing system using a closed queuing network, where vehicles correspond to jobs moving through the network. The authors focused on the rebalancing of empty vehicles and proposed a fluid-based optimization problem to study the steady-state convergence of the network. Banerjee, Freund, and Lykouris (2021) introduced an approximation framework for closed queuing network models of vehicle-sharing systems. Their framework is based on a continuous-time Markov chain and models pricing, matching, and rebalancing while considering multiple objectives and system constraints. Chopra et al. (2023) studied a hybrid transportation system consisting of a mix of shuttle buses and on-demand vehicles. The authors analyzed the system using a queuing model where passengers wait in a single queue for a shuttle service. If a passenger arrives when the queue length is greater than the shuttle capacity, then the passenger is assigned to an on-demand vehicle. Zhang and Pavone (2016) and Zhang, Rossi, and Pavone (2019) introduced closed Jackson network models for analyzing and controlling (autonomous) mobility-on-demand systems. Despite recent advances in using queuing systems to model ride-hailing, no previous work has analyzed a mixed-service system that combines on-demand and pre-booked rides.

5.1.2 Contributions

Our work addresses the research gaps outlined above. Specifically, our contribution is four-fold. First, we present a queuing system that models the mixed-service ride-hailing system at a high level and motivates the need for a more accurate optimization framework that captures the spatial and temporal intricacies of the problem. Second, we introduce a novel two-stage stochastic optimization model and present a SAA formulation that estimates the value of the second-stage recourse function based on a set of scenarios describing possible realizations of on-demand requests. Third, we develop a scalable heuristic algorithm for solving the SAA model. The heuristic approximates the second-stage subproblems as K-disjoint shortest path problems by fixing pick-up times and relaxing the linking constraints that enforce first-stage decisions. The algorithm then combines the solutions to the approximated subproblems and determines a consensus on first-stage decisions based on majority voting. Fourth, we conduct extensive experiments and provide managerial insights regarding the introduction of pre-booking into a ride-hailing system, particularly in the presence of distributional shifts. Our results show that our SAA model increases profits compared to a purely on-demand baseline system, even in the presence of distributional shifts. Moreover, we show that greedily accepting all pre-booking requests can reduce the operator's profit in environments with strong distributional shifts.

The remainder of this paper is organized as follows. Section 5.2 describes our problem setting and presents a queuing-theoretical model for the mixed-service system. Section 5.3 introduces the optimization framework. Section 5.4 presents our experimental design and Section 5.5 discusses computational results. Finally, Section 5.6 concludes the paper.

5.2 Problem Setting & Analysis

We introduce our general problem statement before we present a simplified analysis based on a queuing-theoretical system, which motivates the need for a more detailed optimization framework.

5.2.1 General Problem Statement

We consider a profit-maximizing ride-hailing operator who centrally controls a fleet of vehicles V of fixed size. In our mixed-service system, customers can either pre-book a ride in advance or request a ride on demand. A ride request is defined by a tuple $i = (o_i, d_i, s_i, e_i, p_i)$, indicating its origin o_i, destination d_i, earliest pick-up time s_i, latest pick-up time e_i, and the price p_i paid to the operator. Accordingly, each customer i has a pick-up time window $[s_i, e_i]$. Each vehicle is associated with an initial time s_v from which it is available for service, and an initial location o_v. We consider a continuous planning horizon $[1, T]$. Pre-booking requests enter the system before the start of the planning horizon, and the operator decides to accept or reject each pre-booking request. By accepting a pre-booking request, the operator commits to serving the customer at their earliest pick-up time. During the planning horizon, on-demand requests may arrive at any time, and the operator decides to accept or reject each incoming request and dispatches vehicles to serve all accepted (pre-booking and on-demand) requests. Notably, the operator makes pre-booking acceptance decisions without knowledge of future on-demand requests and before determining vehicle routes. We denote the set of all pre-booking requests by D_1 and the set of all on-demand requests that arrive throughout the planning horizon by D_2.

Before presenting our optimization framework, we analyze the potential effects of distributional shifts on the operator's profit in a stylized setting based on a queuing-theoretical model. Although this simplified setting does not capture all the problem's nuances, particularly neglecting spatial aspects, it already reveals interesting unintuitive effects, strongly indicating the need for a further detailed study.

5.2.2 Queuing-theoretical Analyses

We introduce a simplified queuing model, with as few degrees of freedom as possible, that captures the core components of the mixed-service ride-hailing system, aiming to obtain managerial insights while abstracting the technical details of the system's operation. We first introduce the baseline queuing model for a purely on-demand system, before we present a queuing model for the mixed-service system.

Baseline Queuing System: In the baseline case, we consider that the system receives only on-demand requests, and the operator may decide to reject a percentage of the incoming requests to ensure that the wait time in the queue remains below a certain time limit. We model the system using an M/M/1 queue with Poisson distributed arrivals with arrival rate λ and service rate μ. The average wait time in an M/M/1 queue is given by: $\overline{W}_q = \frac{\lambda}{\mu(\mu-\lambda)}$. The operator observes an arrival rate Λ and decides on a share of incoming requests that are allowed to enter the M/M/1 queue such that the wait time in the queue is on average not longer than $W_{\max}$. Variable y models the operator's acceptance decision, determining the share of incoming requests that enter the queue. We formulate the operator's revenue maximization problem as:

$$
\begin{aligned}
R_{\text{baseline}} \;=\; \max \quad & \lambda p && (5.1)\\
\text{s.t.} \quad & \lambda \leq W_{\max}\,\mu(\mu-\lambda) && (5.2)\\
& \lambda = \Lambda y && (5.3)\\
& 0 \leq y \leq 1 && (5.4)
\end{aligned}
$$

which has the following optimal solution:

$$
y^* = \min\left\{1,\; \frac{1}{\Lambda}\frac{W_{\max}\,\mu^2}{(1+W_{\max}\mu)}\right\} \tag{5.5}
$$

Mixed-service Queuing System: We consider a system that receives a mix of pre-booking and on-demand requests, as depicted in Figure 5.1. We consider two customer types with arrival rates Λ_1 and Λ_2, and service times $1/\mu_1$ and $1/\mu_2$, respectively. Customer type 1 corresponds to pre-booking requests, and customer type 2 corresponds to on-demand requests. We model the system using a non-preemptive priority queue with two priority classes, where pre-booking requests have priority class 1 and on-demand requests have priority class 2. Accordingly, pre-booked requests have higher priority than

on-demand requests. The expressions for the wait times in each queue are given by:

$$\overline{W}_{q,1} = \frac{\rho_1/\mu_1 + \rho_2/\mu_2}{1-\rho_1} \tag{5.6}$$

$$\overline{W}_{q,2} = \frac{\rho_1/\mu_1 + \rho_2/\mu_2}{(1-\rho_1)(1-\rho_1-\rho_2)} \tag{5.7}$$

where $\rho_1 = \lambda_1/\mu_1$ and $\rho_2 = \lambda_2/\mu_2$ (cf. Larson and Odoni 1981).

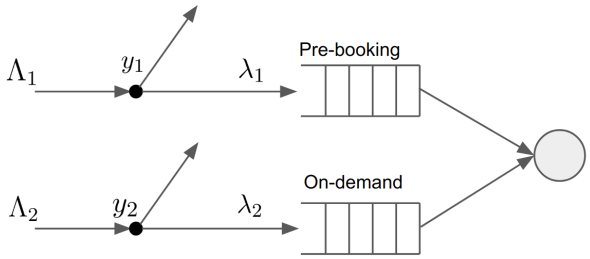


Figure 5.1: Non-preemptive priority queue with acceptance decisions and wait time constraints.

Let $y_1 \in [0,1]$ and $y_2 \in [0,1]$ be the acceptance decisions regarding pre-booking and on-demand requests, respectively. We formulate the operator's revenue maximization problem:

$$\begin{aligned}
R_{\text{mixed}} \;=\; \max \quad & \lambda_1 p_1 + \lambda_2 p_2 && (5.8)\\
\text{s.t.} \quad & \lambda_1 = \Lambda_1\, y_1 && (5.9)\\
& \lambda_2 = \Lambda_2\, y_2 && (5.10)\\
& \frac{\rho_1/\mu_1 + \rho_2/\mu_2}{1-\rho_1} \le W_1 && (5.11)\\
& \frac{\rho_1/\mu_1 + \rho_2/\mu_2}{(1-\rho_1)(1-\rho_1-\rho_2)} \le W_2 && (5.12)\\
& 0 \le y_1 \le 1 && (5.13)\\
& 0 \le y_2 \le 1 && (5.14)
\end{aligned}$$

which is equivalent to:

$$\begin{aligned}
\max \quad & p_1\Lambda_1\, y_1 + p_2\Lambda_2\, y_2 && (5.15)\\
\text{s.t.} \quad & \frac{\Lambda_1}{\mu_1}\left(\frac{1}{\mu_1} + W_1\right) y_1 + \frac{\Lambda_2}{\mu_2^2} y_2 \le W_1 && (5.16)\\
& -\frac{\Lambda_1^2}{\mu_1^2} W_2\, y_1^2 - \frac{\Lambda_1\Lambda_2}{\mu_1\mu_2} W_2\, y_1 y_2 + \frac{\Lambda_1}{\mu_1}\left(\frac{1}{\mu_1} + 2W_2\right) y_1 + \frac{\Lambda_2}{\mu_2}\left(\frac{1}{\mu_2} + W_2\right) y_2 \le W_2 && (5.17)\\
& 0 \le y_1 \le 1,\; 0 \le y_2 \le 1 && (5.18)
\end{aligned}$$

The model formulation is a non-convex quadratically constrained linear program. In the

following, we compare the mixed-service system against the baseline system in a numerical simulation.

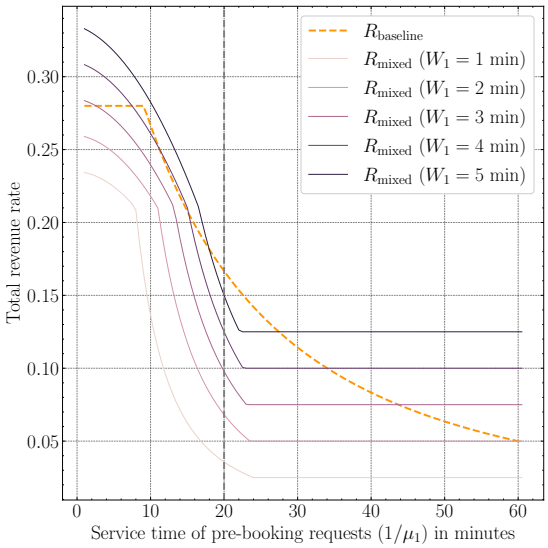


Figure 5.2: Revenue curve of queuing model

Simulation of Queuing System: In this simulation, we compare the revenue performance of the two systems by fixing the average service time of on-demand requests $1/\mu_2$, while varying the service time of pre-booking requests $1/\mu_1$. This corresponds to a setting where pre-booking and on-demand requests are characterized by different distributions, thereby modeling a distributional shift. Specifically, we consider a setting with $|V|= 2000$ vehicles, $|D_1|= |D_2|= 10000$ pre-booking and on-demand requests, and a time horizon of $T = 360$ minutes (6 hours). We compute the arrival rate for pre-booking and on-demand requests as $\Lambda_1 = \Lambda_2 = \frac{|D_1|}{|V|\times T} \approx 0.014$ requests per minute. We consider an average service time of 20 minutes for on-demand requests, leading to a service rate of $\mu_2 = 0.05$. We set the maximum wait time for on-demand requests to $W_2 = 10$ minutes and set the prices to $p_1 = 15$ and $p_2 = 10$, respectively, for pre-booked and on-demand requests.

In the baseline system, we assume that pre-booking customers enter the same queue as on-demand customers, such that there is no service differentiation. Thus, the arrival rate is $\Lambda = \Lambda_1 + \Lambda_2$ and we compute the mean service time as the weighted average among the service times of on-demand and pre-booking requests, leading to the service rate:

$$\mu = \frac{\Lambda_1 + \Lambda_2}{\Lambda_1/\mu_1 + \Lambda_2/\mu_2}. \tag{5.19}$$

Figure 5.2 shows the revenue in the y-axis as we increase the service time of pre-booked requests in the x-axis. The dashed orange line shows the revenue curve for the baseline system, while the solid lines show the revenue curves for the mixed-service system for

different values of the maximum pre-booking wait time $W_1 \in \{1, 2, 3, 4, 5\}$, in minutes. Under the baseline system, the operator accepts all requests up to the critical point when the average wait time in the queue reaches the maximum allowable wait time $W_2 = 10$ minutes, which corresponds to an average service time of $1/\mu = 14.548$ minutes. Accordingly, the baseline revenue remains constant as long as the pre-booking service time is less than $1/\mu_1 = 9.097$ minutes. After this point, revenue starts to decrease. Under the mixed-service system, the revenue also decreases as we increase the pre-booking service time. For this system, there is also a critical point, when the pre-booking service time is approximately between $1/\mu_1 = 23$ and $1/\mu_1 = 24$ minutes, after which the operator must reject all pre-booking requests in order to satisfy the corresponding wait time constraint. After this critical point, the acceptance rate of on-demand requests remains constant at $y_1 = W_1\mu_2^2/\Lambda_2$.

This analysis shows that introducing service differentiation between pre-booking and on-demand requests can result in either an increase or a decrease in revenue compared to the baseline system, depending on service quality constraints and distributional shifts between pre-booking and on-demand requests. Therefore, it is not straightforward to determine whether a mixed-service system would be more profitable than a purely on-demand system in practice. We acknowledge that this stylized setting does not capture some important aspects of real ride-hailing systems, e.g., spatial and temporal dynamics. In the following, we hence introduce an optimization framework that represents the system more faithfully, allowing us to conduct a more detailed and nuanced analysis.

5.3 Optimization framework

We model the problem faced by the operator as a two-stage stochastic optimization problem. The first-stage problem models the planning phase before operations start, in which a set of pre-booking requests D_1 enter the system, and the operator decides on accepting or rejecting each pre-booking request. The second-stage problem models the operations during the planning horizon, which involves acceptance decisions regarding on-demand customers and routing decisions for all accepted requests. We are interested in analyzing the pre-booking acceptance decisions of the first-stage problem. Therefore, to not bias our analyses, we compute an upper bound for the second stage, assuming that on-demand requests are all revealed at once, corresponding to a setting with perfect foresight. We denote by D_2 the uncertain set of on-demand requests that enter the system in the second stage.

5.3.1 Two-stage Stochastic Optimization Model

Let x_i be a binary decision variable, taking value 1 if the operator accepts the pre-booking ride request $i \in D_1$. Each pre-booking customer $i \in D_1$ incurs an additional pre-booking fee $p_{i,1}$. The two-stage stochastic optimization problem maximizes the revenue from pre-booking fees and the expected profit obtained from the second-stage problem.

$$\max \quad \sum_{i \in D_1} x_i \, p_{i,1} + \mathbb{E}_{D_2}[\Pi(\mathbf{x}; D_1, D_2)] \tag{5.20}$$

$$\text{s.t.} \quad x_i \in \{0, 1\} \qquad \forall i \in D_1, \tag{5.21}$$

where $\Pi(\mathbf{x}; D_1, D_2)$ is the optimal objective value of the second-stage problem. Uncertainty in the second-stage problem stems from the set of on-demand requests D_2.

We define the second-stage problem on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. Each node in $\mathcal{V} = D_1 \cup D_2 \cup V$ represents either a request $i \in D_1 \cup D_2$ or a vehicle $v \in V$. The arc set is composed of two sets of arcs $\mathcal{A} = \mathcal{A}_V \cup \mathcal{A}_D$, where the existence of an arc $(v, i) \in \mathcal{A}_V$ represents the possibility for customer $i \in D_1 \cup D_2$ to be the first request picked up by vehicle $v \in V$, while an arc $(i, j) \in \mathcal{A}_D$ represents the possibility for a vehicle to pick up customer $j \in D_1 \cup D_2$ immediately after servicing customer $i \in D_1 \cup D_2$. Let $\tau(\ell_1, \ell_2)$ be a function that computes the travel time from location ℓ_1 to location ℓ_2, and let $\delta(\ell_1, \ell_2)$ be a function that computes the driving distance from location ℓ_1 to ℓ_2. Each arc $(v, i) \in \mathcal{A}_V$ has a travel time $T_{vi} = \tau(o_v, o_i)$ from the initial location of vehicle v to the origin o_i of request i and a profit $\pi_{vi} = p_i - \beta\delta(o_v, o_i) - \beta\delta(o_i, d_i)$, i.e., the fare paid by i minus the cost of driving from o_v to the pick-up point of i and to its destination. Parameter β expresses the unit operating cost which is a combination of fuel consumption cost and driver's wage. Each arc $(i, j) \in \mathcal{A}_D$ has a travel time $T_{ij} = \tau(o_i, d_i) + \tau(d_i, o_j)$ representing the time to serve request i and to drive from the destination d_i to the origin o_j, such that we must have $s_i + T_{ij} \leq e_j$. In addition, each arc $(i, j) \in \mathcal{A}_D$ has a profit $\pi_{ij} = p_{j,2} - \beta\delta(d_i, o_j) - \beta\delta(o_j, d_j)$, i.e., the fare paid by j minus the cost of driving from the drop-off point of i to the pickup point of j (dead-heading) and cost of driving customer j to its destination (occupied trip). Based on the travel times defined above, we can equivalently define the arc set as:

$$\mathcal{A} = \{(i, j) \in \mathcal{V} \times (D_1 \cup D_2) : s_i + T_{ij} \leq e_j\}. \tag{5.22}$$

We denote by $N^+(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{A}\}$ the set of nodes that are reachable by the arcs going out of $i \in \mathcal{V}$ and we denote by $N^-(j) = \{i \in \mathcal{V} : (i, j) \in \mathcal{A}\}$ the set of nodes that can reach $j \in \mathcal{V}$ by the arcs going into j.

The second-stage vehicle dispatching problem assigns a sequence of requests to each vehicle, such that the operator's profit is maximized, subject to routing and pick-up time

window constraints. Let y_i denote the request acceptance decisions, i.e., $y_i = 1$ if request i is picked up by a vehicle. We represent routing decisions by binary variables z_{vi} and z_{ij}, where $z_{vi} = 1$ if request i assigned to vehicle v as a first request in the sequence, and $z_{i,j} = 1$ if request i is picked up by a vehicle immediately after request j. Variable $t_i \in [s_i, e_i]$ models the pick-up time of request i. Given pre-booking and on-demand request sets, D_1 and D_2, and first-stage decisions $\mathbf{x}$, we formulate the second-stage problem as follows:

$$\Pi(\mathbf{x}; D_1, D_2) = \max \quad \sum_{(i,j)\in\mathcal{A}} \pi_{ij}\, z_{ij} \tag{5.23}$$

$$\text{s.t.} \quad y_i = x_i \qquad \forall i \in D_1 \tag{5.24}$$

$$y_j = \sum_{i\in N^-(j)} z_{ij} \qquad \forall j \in D_1 \cup D_2 \tag{5.25}$$

$$\sum_{j\in N^+(i)} z_{ij} \leq y_i \qquad \forall i \in D_1 \cup D_2 \tag{5.26}$$

$$\sum_{i\in N^+(v)} z_{vi} \leq 1 \qquad \forall v \in V \tag{5.27}$$

$$t_j - t_i \geq (s_j - e_i) + (T_{ij} - (s_j - e_i))z_{ij} \qquad \forall (i,j) \in \mathcal{A}_D \tag{5.28}$$

$$t_i \geq s_i + (s_v + T_{vi} - s_i)z_{vi} \qquad \forall (v,i) \in \mathcal{A}_V \tag{5.29}$$

$$s_i \leq t_i \leq e_i \qquad \forall i \in D_1 \cup D_2 \tag{5.30}$$

$$z_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A} \tag{5.31}$$

$$y_i \in \{0,1\} \qquad \forall i \in D_1 \cup D_2 \tag{5.32}$$

We enforce that wait times for pre-booked customers are equal to zero, i.e., $t_i = s_i = e_i$ for all $i \in D_1$. Constraints (5.24) are linking constraints and ensure that pre-booking requests are served (or not) in accordance with first-stage acceptance decisions. Constraints (5.28) and (5.29) define the pick-up time of each accepted request. Constraints (5.30)–(5.32) define the variable domains.

Since the true underlying distribution of on-demand requests is unknown, we cannot compute the objective value in Model (5.20)–(5.21). Therefore, we approximate the two-stage stochastic optimization problem by SAA. Let $\mathcal{K}$ be a set of scenarios describing possible realizations of the on-demand request set D_2^k, for $k \in \mathcal{K}$. We introduce the SAA model formulation:

$$\max \quad \sum_{i\in D_1} x_i\, p_{i,1} + \frac{1}{|\mathcal{K}|} \sum_{k\in\mathcal{K}} \Pi(\mathbf{x}; D_1, D_2^k) \tag{5.33}$$

$$\text{s.t.} \quad x_i \in \{0,1\} \qquad \forall i \in D_1, \tag{5.34}$$

where we replace the expectation in the objective of Model (5.20)–(5.21) with the average

second-stage objective value among the scenarios.

5.3.2 Heuristic Algorithm

We develop a heuristic for solving the SAA model presented in Section 5.3.1. A core component of our heuristic is the formulation of an approximate second-stage subproblem, where we fix customer pick-up times, and remove the linking constraints related to first-stage decisions. The resulting approximate subproblem is a K-DSPP, which we can solve in polynomial time, e.g., using the algorithm from Schiffer et al. (2021).

To define the approximate second-stage subproblem, we rely on a directed graph $\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{A}})$. As before, the node set $\widehat{\mathcal{V}} = \mathcal{V}$ consists of requests $i \in D_1 \cup D_2$ and vehicles $v \in V$. The arc set $\widehat{\mathcal{A}} = \widehat{\mathcal{A}}_V \cup \widehat{\mathcal{A}}_D$ is such that $\widehat{\mathcal{A}}_V = \mathcal{A}_V$ and $\widehat{\mathcal{A}}_D$ is defined as:

$$\widehat{\mathcal{A}}_D = \{(i,j) \in (D_1 \cup D_2) \times (D_1 \cup D_2) : e_i + T_{ij} \leq e_j\}, \tag{5.35}$$

which ensures that pick-up times are fixed to $t_i = e_i$ for all $i \in D_1 \cup D_2$, resulting in an acyclic graph $\widehat{\mathcal{G}}$. Note that the arc set $\widehat{\mathcal{A}}$ is more restrictive than the arc set $\mathcal{A}$ defined in Equation (5.22). Similar to Section 5.3.1, we define $\widehat{N}^+(i) = \{j \in \mathcal{V} : (i,j) \in \widehat{\mathcal{A}}\}$ and $\widehat{N}^-(j) = \{i \in \mathcal{V} : (i,j) \in \widehat{\mathcal{A}}\}$. We formulate the approximate second-stage subproblem based on graph $\widehat{\mathcal{G}}$ as follows:

$$\widehat{\Pi}(D_1, D_2) = \max \quad \sum_{(i,j)\in\widehat{\mathcal{A}}} \pi_{ij}\, z_{ij} \tag{5.36}$$

$$\text{s.t.} \quad y_j = \sum_{i\in\widehat{N}^-(j)} z_{ij} \qquad \forall j \in D_1 \cup D_2 \tag{5.37}$$

$$\sum_{j\in\widehat{N}^+(i)} z_{ij} \leq y_i \qquad \forall i \in D_1 \cup D_2 \tag{5.38}$$

$$\sum_{i\in\widehat{N}^+(v)} z_{vi} \leq 1 \qquad \forall v \in V \tag{5.39}$$

$$z_{ij} \in \{0,1\} \qquad \forall (i,j) \in \widehat{\mathcal{A}} \tag{5.40}$$

$$y_i \in \{0,1\} \qquad \forall i \in D_1 \cup D_2 \tag{5.41}$$

where we removed linking constraints (5.24). Equation (5.35) ensures that the pick-up time windows are always satisfied, and therefore Constraints (5.28)–(5.30) become unnecessary in this model. The K-DSPP algorithm for solving the approximate subproblem $\widehat{\Pi}(D_1, D_2)$ has a complexity of $O(|\widehat{\mathcal{A}}|(|\widehat{\mathcal{V}}|+K)+K|\widehat{\mathcal{V}}|\log|\widehat{\mathcal{V}}|)$, where $K = |V|$ is the number of vehicles.

Our heuristic algorithm separately solves the approximate subproblem corresponding

Algorithm 5.1: Heuristic algorithm

Input: Set of vehicles V, pre-booking requests D_1, and on-demand scenarios D_2^k, for $k \in \mathcal{K}$

1 **for** $k \in \mathcal{K}$ **do**

2 Solve $\widehat{\Pi}(D_1, D_2^k)$ based on graph $\widehat{\mathcal{G}}^k = (\widehat{\mathcal{V}}^k, \widehat{\mathcal{A}}^k)$ using K-DSPP algorithm (Schiffer et al. 2021)

3 Obtain pre-booking acceptance decisions for scenario k: $x_i^k \leftarrow y_i, \quad \forall i \in D_1 \cup D_2$

4 Determine first-stage decisions $\mathbf{x}$ from $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{|\mathcal{K}|}$, e.g., based on majority vote: $x_i = \mathbb{1}(\frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} x_i^k \geq 0.5), \quad \forall i \in D_1$

5 **for** $k \in \mathcal{K}$ **do**

6 Update arc profits based on first-stage decisions:

$$\pi_{ij}^k \leftarrow \pi_{ij}^k + \begin{cases} M^k & \text{if } x_j = 1 \\ -M^k & \text{if } x_j = 0 \end{cases}, \quad \forall (i,j) \in \mathcal{A}^k : j \in D_1$$

7 Re-solve $\widehat{\Pi}(D_1, D_2^k)$ based on graph $\widehat{\mathcal{G}}^k = (\widehat{\mathcal{V}}^k, \widehat{\mathcal{A}}^k)$ with modified profits $\boldsymbol{\pi}$

to each scenario and then determines first-stage decisions by combining the individual solutions to the subproblems. We describe the heuristic in detail in Algorithm 5.1. At a high level, the algorithm consists of three main steps.

1. **Solving the approximate subproblems:** We separately solve the approximate subproblem for each scenario (lines 2–3).

2. **Consensus fixing:** We determine a consensus among the subproblems regarding first-stage decisions by majority voting (line 4).

3. **Arc weight adjustment and re-solving:** We adjust arc weights, i.e., profits, based on the first-stage decisions and re-solve each subproblem, with the aim of enforcing consistent first-stage decisions among the scenarios (lines 6–7). To determine the value of M^k in line 6, we solve a shortest path problem on graph $\widehat{\mathcal{G}}^k$ using the Bellman-Ford algorithm to obtain the optimal cost c^*, and set $M^k = -c^*$. Our experiments show that defining M^k in this manner provides a sufficiently large constant to enforce consistent first-stage decisions.

5.4 Experimental Design

This section introduces our real-world case study (Section 5.4.1), describes our evaluation metrics (Section 5.4.2), and details our baseline policies (Section 5.4.3).

5.4.1 Case Study

We adopt the data set from NYC Taxi & Limousine Commission, 2010 for our case study. In this paper, we focus on the Yellow Cab rides of Tuesday, January 26, 2010, from 9:00 a.m. to 3:00 p.m. Rides from and to Manhattan constitute 92.17% of all rides in the

considered time horizon. In order to study distributional shifts in the travel demand, we also consider rides from or to the boroughs of Brooklyn, Queens, and The Bronx. We exclude from our experiments rides from or to Staten Island and Newark Airport, amounting to 2.41% of all rides.

Profit calculation: The standard metered fare described in NYC Taxi & Limousine Commission, 2010 is a combination of the trip distance and vehicle speed with possible surcharges, e.g., for trips during the night or during rush hours, among others. Although the data contains information on the fare amount paid by each customer, the values are not always consistent with the described fare calculation and some entries have missing values. To avoid using inaccurate data, we use the standard metered fare to re-calculate the fare associated with each trip. As we do not have accurate data on vehicle speed, we calculate a simplified fare based only on the distance traveled, consisting of an initial charge of \$3.00 plus \$3.50 per mile (equivalently, \$2.1748 per kilometer). For simplicity, we do not consider additional surcharges. For calculating the profit, we assume an operational cost of \$2.50 per mile (cf. Bösch et al. 2018; Litman 2020), which is equivalent to $\beta = \$1.55$ per kilometer. The operator charges an additional fee to pre-booked customers corresponding to 10% of the regular fare amount.

Look-up table of distances and travel times: Computing the pair-wise distance and travel time between every pair of nodes in the graph can be excessively time-consuming and memory-intensive. To avoid this computational burden, we discretize the road network using the algorithm detailed in Appendix 5.B with a discretization radius of 200 meters, and pre-compute a look-up table with distances and travel times between region centers. To generate the look-up table, we compute the path with the shortest travel time between each pair of region centers. Travel times are based on data retrieved from Uber Movement for the year 2020 and take traffic conditions into account. If a request's pick-up and drop-off locations are in the same region, we assume a line distance and an average driving speed of 20 kilometers per hour.

5.4.2 Evaluation Metrics

In-sample performance: Given first-stage decisions $\hat{\mathbf{x}}$, obtained by solving the SAA model defined on scenarios D_2^k, for $k \in \mathcal{K}$, we evaluate the in-sample profit:

$$P(\hat{\mathbf{x}}) = \sum_{i \in D_1} \hat{x}_i \, p_i + \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \Pi(\hat{\mathbf{x}}, D_1, D_2^k) \tag{5.42}$$

which consists of the average second-stage objective value based on the in-sample scenarios.

Out-sample performance: Given first-stage decisions $\hat{\mathbf{x}}$, we estimate the out-of-sample profit based on a set of (testing) scenarios $\overline{D}_2^k$, for $k \in \overline{\mathcal{K}}$:

$$\overline{P}(\hat{\mathbf{x}}) = \sum_{i \in D_1} \hat{x}_i \, p_i + \frac{1}{|\overline{\mathcal{K}}|} \sum_{k \in \overline{\mathcal{K}}} \Pi(\hat{\mathbf{x}}, D_1, \overline{D}_2^k) \tag{5.43}$$

which consists of the average second-stage objective value based on out-of-sample scenarios.

5.4.3 Baseline Policies

We compare the solutions to the SAA model against the following baseline policies:

- **Accept-all:** this policy accepts all incoming pre-booking requests regardless of their characteristics. Note that if the number of pre-booking requests is large compared to the number of vehicles, it might be infeasible to serve all pre-booked requests without delays.
- **Reject-all:** corresponds to the setting in which the operator does not offer a pre-booking service. In this case, we assume that the pre-booking customers request a ride on demand instead. When solving the SAA model, we create a copy of all requests from set D_1 in each scenario D_2^k and create an empty set of pre-booking requests $D_1 = \varnothing$.

5.5 Results

First, we perform an assessment of the heuristic solutions based on small instances for which the MIP solver can find provably optimal solutions (Section 5.5.1). Second, we discuss results from a managerial perspective on large instances (Section 5.5.2). Third, we analyze structural properties of the SAA solutions compared to the baseline policies (Section 5.5.3). Fourth, we present and evaluate some simple and explainable heuristic policies that are easy to implement in practice (Section 5.5.4).

5.5.1 Assessment of Heuristic Solutions

We assess the heuristic solutions regarding scalability and solution quality. Figure 5.3a shows the run-time of the MIP solver and the heuristic algorithm as we increase the number of requests in the system, keeping the number of vehicles fixed to 100. The MIP solver reaches the time limit already for instances with two thousand requests. In

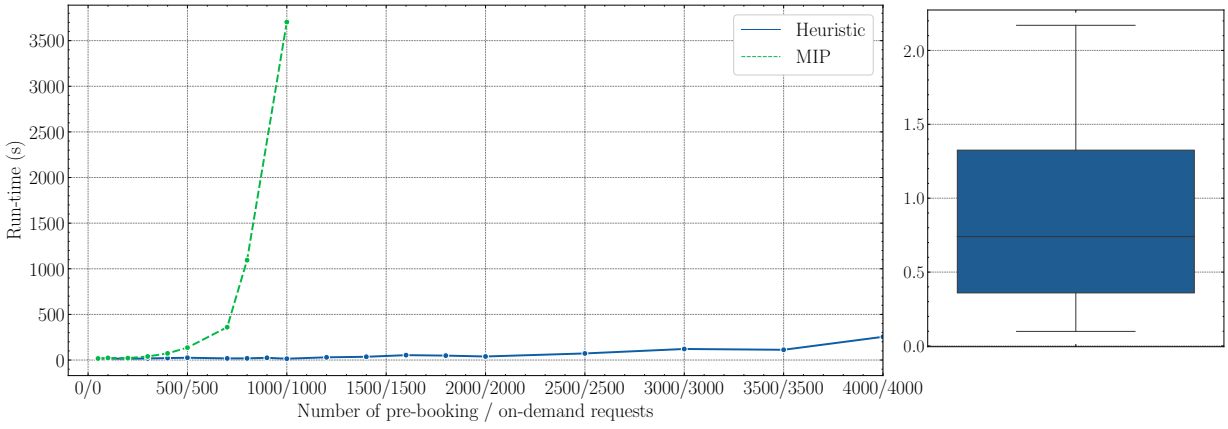
contrast, the heuristic algorithm can solve instances with ten thousand requests in under eight minutes.

We evaluate the quality of the heuristic solutions by computing the gap to the corresponding optimal solution, as defined by the following metric:

$$\text{Gap}(\%) = \frac{\hat{P} - P^*}{P^*}, \tag{5.44}$$

where $\hat{P}$ is the total profit given by a heuristic solution and P^* is the profit given by the corresponding optimal solution to the SAA model. Figure 5.3b shows the distribution of gap values for the heuristic solutions based on small instances with 100 to 1600 requests and 100 vehicles, for which the MIP solver could find optimal solutions. Our heuristic provides near-optimal solutions with an optimality gap not larger than 2.2% and with a median value below 1%.

Our results show that the heuristic algorithm can efficiently find near-optimal solutions, making it suitable for solving the SAA model and analyzing the mixed-service system. In the following sections, we adopt the heuristic algorithm to solve larger instances and derive managerial insights.



(a) Run-times

(b) Gap to optimality

Figure 5.3: Comparison between MIP solver and heuristic algorithm

5.5.2 Managerial Analysis

We compare the out-of-sample performance of our algorithm against the baseline policies. We focus on a base case with 2000 vehicles, 5000 pre-booking requests, and 3 scenarios, each containing 10000 on-demand requests sampled without replacement from the complete pool of rides over our planning horizon. We perform a sensitivity analysis where we fix the total number of requests to 15000, and we investigate the profit performance for

varying fleet sizes and for different amounts of pre-booking requests. For this analysis, we obtain the first-stage solution based on in-sample scenarios $\mathcal{K}$. We then fix the first-stage decisions and evaluate the profit based on out-of-sample scenarios $\overline{\mathcal{K}}$. In addition, we compute the full-information solution, where we obtain the first-stage decisions based on the out-of-sample scenarios $\overline{\mathcal{K}}$.

We analyze the effect of distributional shifts by sampling pre-booking requests from a different distribution than on-demand requests. Specifically, given the number of pre-booking requests m and given the share of pre-booking requests ν that are outside of Manhattan, i.e., with pick-up or drop-off locations outside of Manhattan, we construct D_1 by sampling $m\nu$ requests from boroughs outside Manhattan and $m(1-\nu)$ requests from within Manhattan.

Profit performance: Figure 5.4 shows the total profit from solutions to the SAA model and the baseline policies as we increase the share parameter ν. We observe several interesting insights. First, the SAA solutions have significantly higher profit than the reject-all baseline, regardless of the parameter value ν. Accordingly, optimizing the mixed-service system based on the SAA model leads to an increase in profit compared to the purely on-demand system, regardless of how strong the distributional shift is. Second, greedily accepting all pre-booking requests can lead to a significant decrease in profit compared to the purely on-demand system, especially in regimes with stronger distributional shifts. Third, in settings with weak distributional shifts, the greedy accept-all policy is more profitable than the reject-all policy, showing a performance comparable to the SAA solutions. Finally, we note that the SAA solution closely approximates the full-information solution. Although this might be surprising at first glance, we argue that the performance resulting from first-stage decisions is not heavily affected by the uncertainty from second-stage on-demand requests. Due to our model's design, most of the decisions are taken in the second stage while the first stage only involves pre-booking acceptance decisions. Accordingly, after committing to first-stage decisions, the operator still has much flexibility in the second-stage problem for deciding on the routes and whether to accept or reject on-demand requests. Therefore, if the operator makes first-stage decisions that are close to the full-information solution, the overall profit will likely be close to the full-information profit. In Appendix 5.C.1, we analyze the accuracy of the first-stage decisions given by the SAA solution compared to the full-information solution. In summary, we highlight that the primary driver of our results is the distributional shift rather than the uncertainty regarding on-demand requests. As previously mentioned, our experiments are based on an upper bound for the second-stage problem, which represents a best-case scenario for the operator. We note that results may differ if we consider online decision-making in the second-stage problem. We leave this investigation for future work.

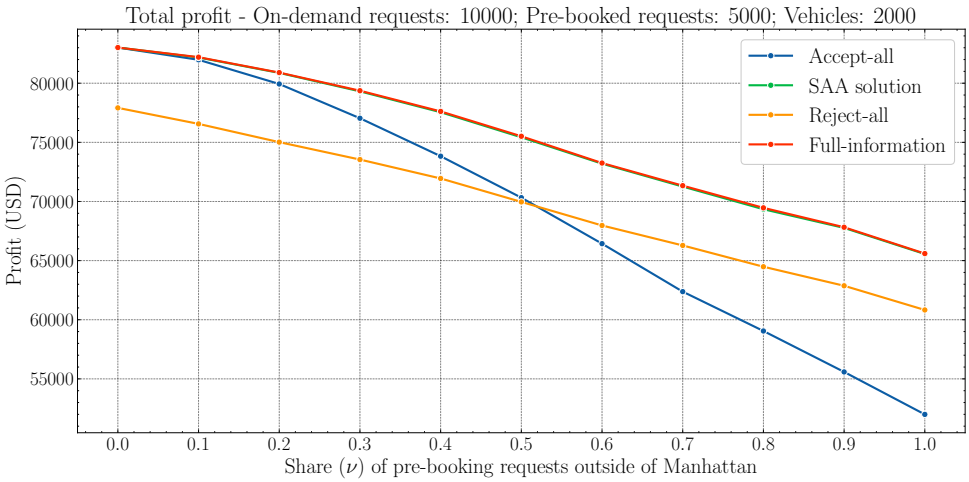


Figure 5.4: Out-of-sample performance of the heuristic algorithm

Result 5.1 *Introducing a pre-booking service into the ride-hailing system can lead to a significant increase in profit, regardless of how strong the distributional shift is.*

Result 5.2 *In settings with weak distributional shifts, accepting all pre-booking leads to an increase in profit compared to the purely on-demand baseline.*

Result 5.3 *In settings with strong distributional shifts, accepting all pre-booking requests can lead to a significant decrease in profit compared to the purely on-demand baseline.*

Varying the ratio of pre-booking requests: Figure 5.5 shows profit curves for different values of $\nu \in \{0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9\}$, where we vary the number of pre-booking requests while keeping the number of vehicles fixed to 2000 and the total number of requests fixed to 15000. As can be seen, in settings with strong distributional shifts, i.e., when $\nu \geq 0.4$ (bottom plots), increasing the ratio of pre-booking requests leads to a decrease in profit. This occurs because more pre-booked requests result in more trips outside of Manhattan, consequently increasing the number of unfavorable rides. At $\nu = 0.3$, we observe a shift in the curve's behavior. Specifically, when $\nu \in \{0.1, 0.2\}$, increasing the ratio of pre-booking requests results in an increase in profit for the SAA solution. Across all settings, the on-demand baseline shows a decrease in profit as the ratio of pre-booking requests increases.

Result 5.4 *Increasing the ratio of pre-booking requests leads to a decrease in profit in settings with strong distributional shift, but results in a profit increase in settings with weak distributional shift.*

Varying fleet size: Figure 5.6 displays profit curves for different values of $\nu \in \{0.1, 0.4, 0.6, 0.9\}$, where we vary the fleet size from 125 to 2000 vehicles while keeping the number of pre-booking requests fixed to 5000 and the number of on-demand requests fixed to 10000. In

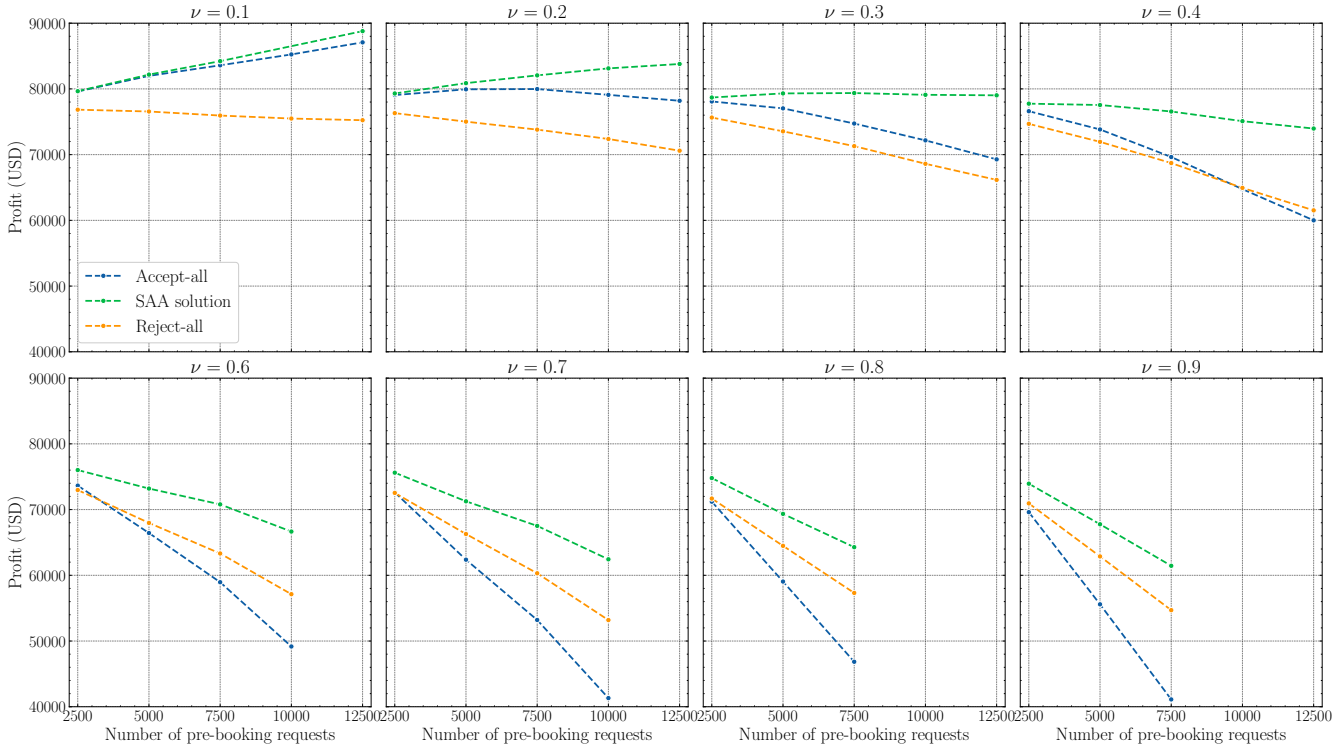


Figure 5.5: Profit performance for varying ratios of pre-booking requests and different values of ν

general, when the fleet size is small, the greedy accept-all policy shows the worst performance among all policies. In settings with a weak distributional shift, e.g., when $\nu \leq 0.4$, oversizing the fleet can compensate for the unfavorable rides resulting from the distributional shift, allowing the accept-all policy to eventually outperform the on-demand reject-all baseline. In settings with strong distributional shifts, e.g., when $\nu \geq 0.6$, the accept-all policy results in remarkably worse profits compared to the on-demand reject-all baseline, even with an oversized fleet.

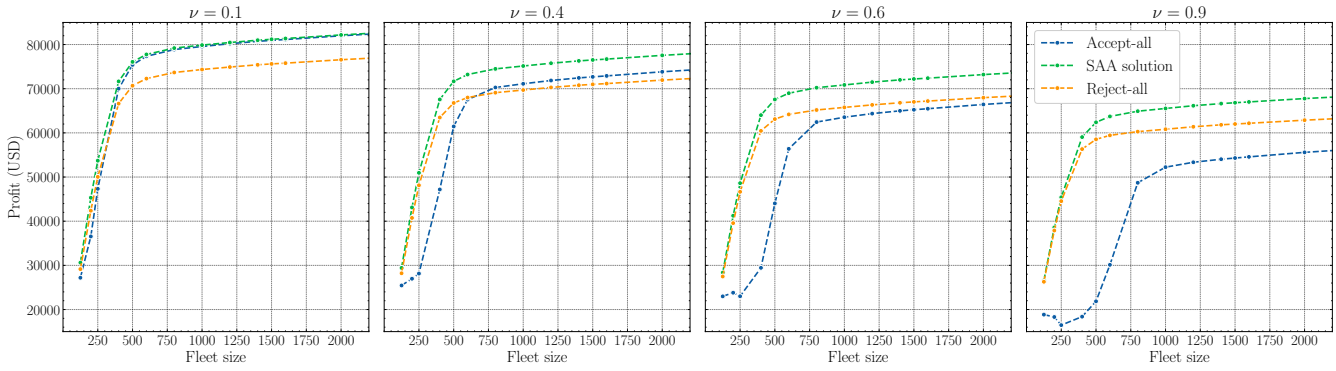


Figure 5.6: Profit performance for varying fleet sizes and different values of ν

Result 5.5 *An operator adopting a greedy accept-all policy can compensate for weak distributional shifts by oversizing their fleet size, whereas strong distributional shifts lead to*

a significant decrease in profit.

Regime switches: Considering the two baseline policies described in Section 5.4.3, we distinguish between two regimes, depending on which of the two policies outperforms the other. As previously observed in Figure 5.4, the reject-all baseline policy outperforms the accept-all policy in settings characterized by strong distributional shifts. As the value of ν decreases, a regime switch eventually occurs, resulting in the accept-all policy outperforming the reject-all baseline. We now characterize the regime switches for varying fleet sizes and ratios of pre-booking requests. Figure 5.7 shows the values of ν at which we observe a regime switch. As can be seen, systems with smaller fleets and large ratios of pre-booking requests experience a regime switch at lower values of ν. Ride-hailing operators adopting the accept-all policy may benefit from a system in which a regime switch occurs at high values of ν, as they would be profitable even with strong distributional shifts. The results from Figure 5.7 can support such operators in designing their system. Accordingly, by adjusting the fleet size or the ratio of pre-booking requests, the operator may sustain a regime where the accept-all policy outperforms the on-demand baseline. For example, in a system with a fixed ratio of pre-booking requests, increasing the fleet size can help the operator remain profitable even when faced with stronger distributional shifts.

Result 5.6 *In a mixed-service system utilizing the accept-all policy, the operator can improve profit performance compared to the reject-all policy by increasing their fleet size and reducing the number of pre-booking requests.*

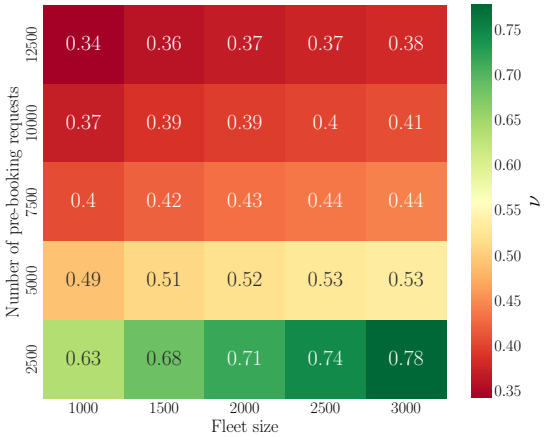


Figure 5.7: Regime switch for varying fleet sizes and number of pre-booking requests

5.5.3 Analysis of Structural Properties

In the following, we provide insights into the structure of the solutions by analyzing the total number of customers served by each vehicle, the spatial distribution of rides, and

customer profitability.

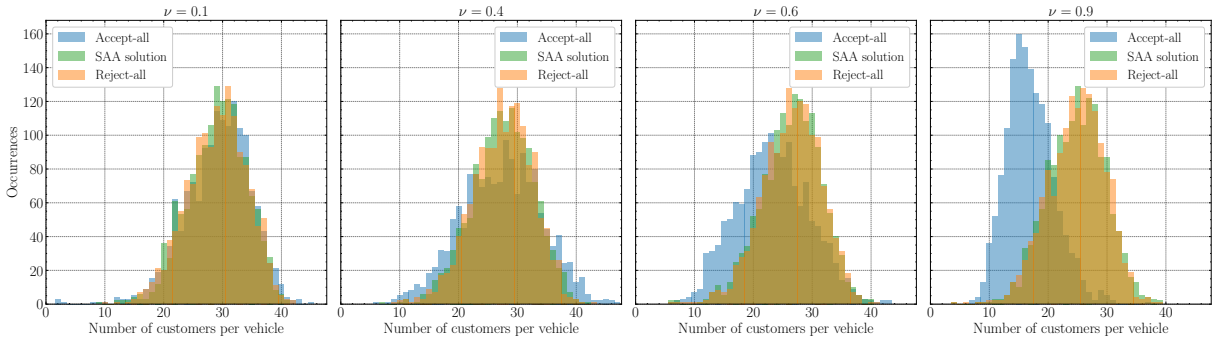
Number of served customers: Figure 5.8 illustrates the distribution of the number of customers assigned to each vehicle. Figures 5.8a and 5.8b present results for instances with 500 and 2000 vehicles, respectively, with varying parameter values $\nu \in \{0.1, 0.4, 0.6, 0.9\}$, as indicated at the top of each plot. When the fleet size is small, see Figure 5.8a, all vehicles serve a relatively large number of customers, such that almost all vehicles serve at least 10 customers in total. In settings with weak distributional shifts, as on the left plots, the distributions for all policies are similar. However, as the value of ν increases, the distribution for the accept-all policy changes, with both its mean and variance decreasing. In the setting with $\nu = 0.9$, under the accept-all policy, vehicles serve a large number of pre-booking requests outside of Manhattan, leaving the drivers with little time available to accommodate additional customers in their routes. When the fleet size is large, see Figure 5.8b, most vehicles tend to serve a small number of customers along their routes. In this case, the operator can easily serve most requests since the fleet is oversized, resulting in some vehicles spending a considerable amount of time idling.

Result 5.7 *With a small fleet, drivers serve many customers during the planning horizon, whereas with a large fleet, most vehicles serve only a few customers.*

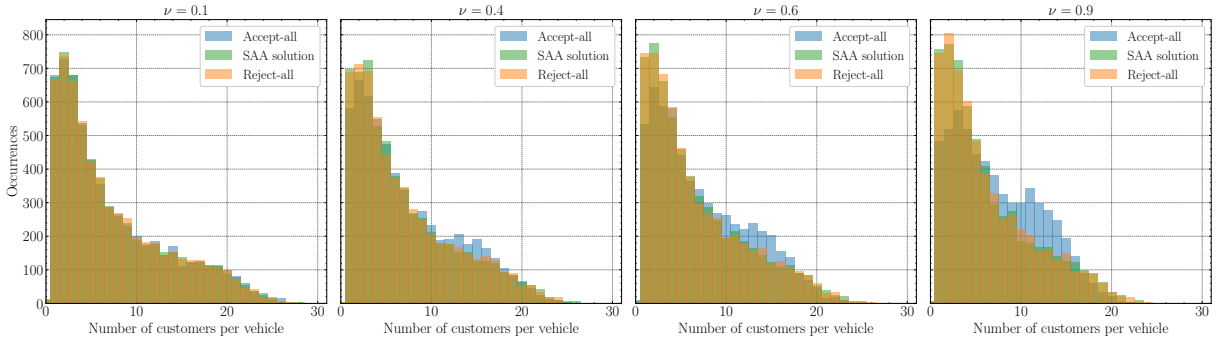
Result 5.8 *Both the SAA solution and the reject-all policy show similar distributions for the number of customers per vehicle, while the accept-all policy diverges significantly under strong distributional shifts.*

Spatial distribution of rides: Figure 5.9 illustrates trips selected by each policy that have either the origin or destination (or both) outside Manhattan. We focus on a small instance with 100 vehicles, 400 on-demand requests, and 120 pre-booking requests, with 30% of the pre-booked requests being outside Manhattan, i.e., $\nu = 0.3$. To provide a more comprehensible visualization, we omit rides within Manhattan. Under the reject-all policy, we observe fewer trips from/to outside Manhattan. Out of those trips that leave Manhattan, many have an origin or destination in the JFK International Airport. Moreover, for most of those trips leaving Manhattan, the pick-up and drop-off locations are usually not far from Manhattan. In the SAA solution, we observe slightly more trips in the close vicinity of Manhattan, e.g., around Downtown Brooklyn or in the Queens' neighborhoods that are closest to Manhattan. Finally, under the accept-all policy, we observe more trips, e.g., going through Brooklyn and Queens and some rather short trips within The Bronx.

Result 5.9 *Under the accept-all policy, we observe a more intense flow of vehicles serving unfavorable trips outside of Manhattan.*

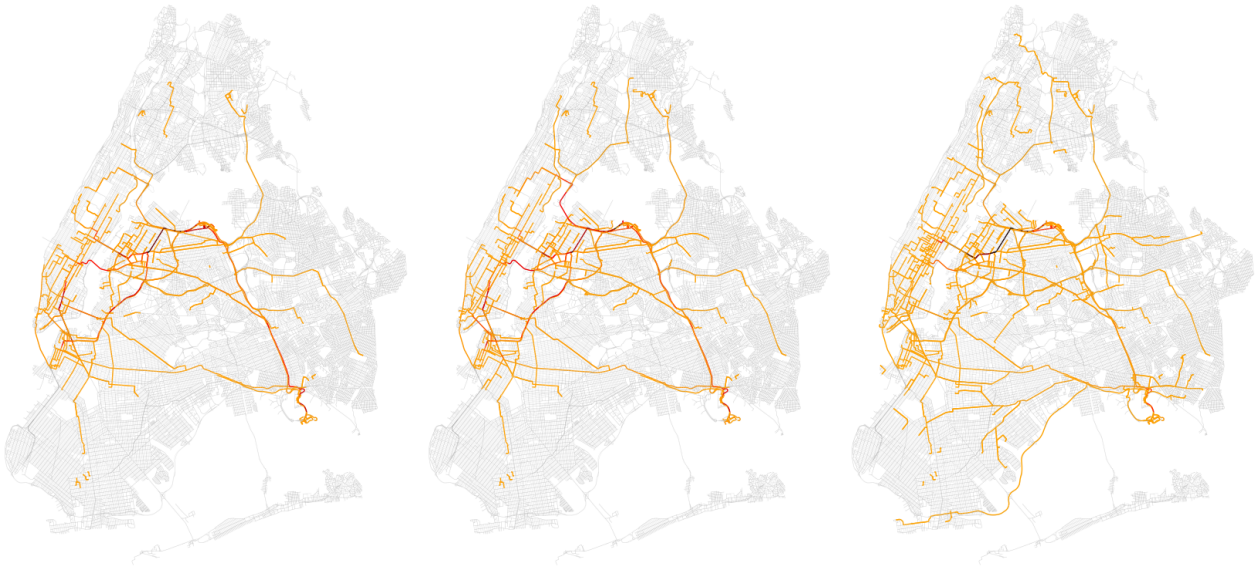


(a) Results for 500 vehicles



(b) Results for 2000 vehicles

Figure 5.8: Number of customers per vehicle



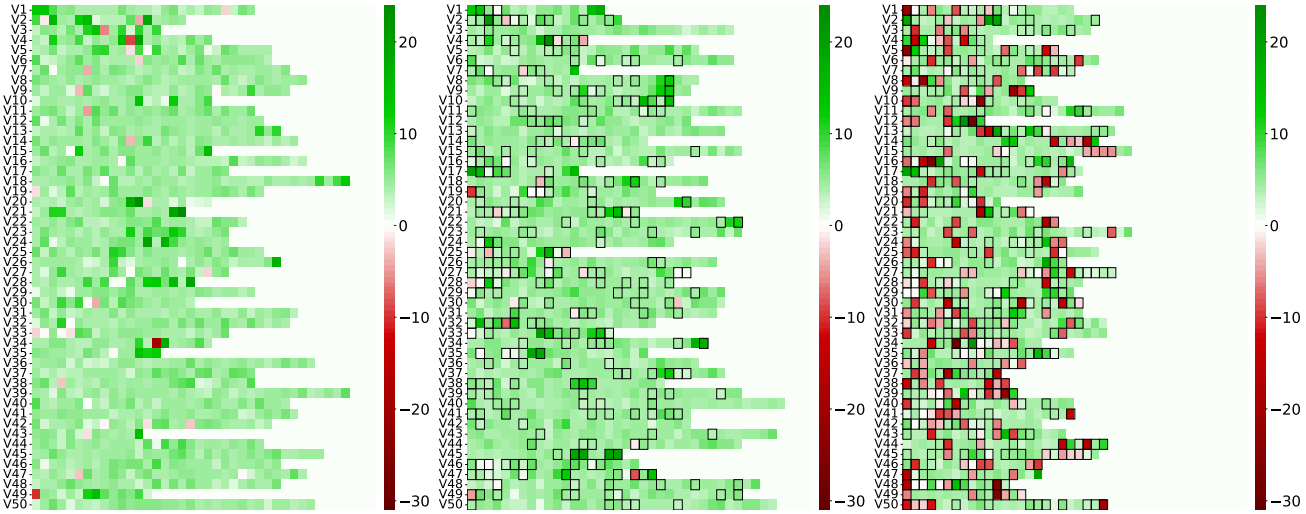
(a) Reject-all policy (b) SAA solution (c) Accept-all policy

Figure 5.9: Sample trips leaving Manhattan for a setting with $\nu = 0.3$

Customer profitability: Figure 5.10 illustrates the profits associated with each customer in the routes served under each policy. The y-axis represents each vehicle, labeled from V1 to V50, while the sequence of boxes along the x-axis denotes the rides served by each vehicle. The color of each box indicates the amount of profit or loss incurred by the

operator for that specific ride. Boxes representing pre-booked rides are highlighted by a black border around them. As can be seen, the solution based on the accept-all policy has many unfavorable pre-booking rides with negative profit. We observe unfavorable rides even in the optimal solution, often associated with pre-booked customers. Although the operator incurs a loss for serving those rides, they are more profitable than driving empty to the next customer, effectively allowing these customers to serve the function of relocating vehicles. Overall, we typically observe a positive total profit in routes that contain unfavorable trips.

Result 5.10 *Most unprofitable rides are associated with pre-booked customers, especially under the accept-all policy.*



(a) Reject-all policy (b) SAA solution (c) Accept-all policy

Figure 5.10: Profit heatmaps for a setting with $\nu = 0.3$.

5.5.4 Heuristic Policies

Based on the insights from Section 5.5.3, we devise some heuristics that exploit structural characteristics of the problem. The heuristic policies described in this section have the advantages of being explainable and easily implementable in practice, which are desirable characteristics in certain business settings.

- **Accept-all in Manhattan:** we accept all pre-booking requests with pick-up and drop-off locations in Manhattan, and reject all requests outside Manhattan.
- **Distance-based policy:** accept all pre-booking requests that (a) are within Manhattan, or (b) have a ride distance shorter than a given threshold. The reasoning

behind this policy is that by accepting shorter and rejecting longer rides outside of Manhattan, the vehicles will have more time available for serving favorable on-demand requests.

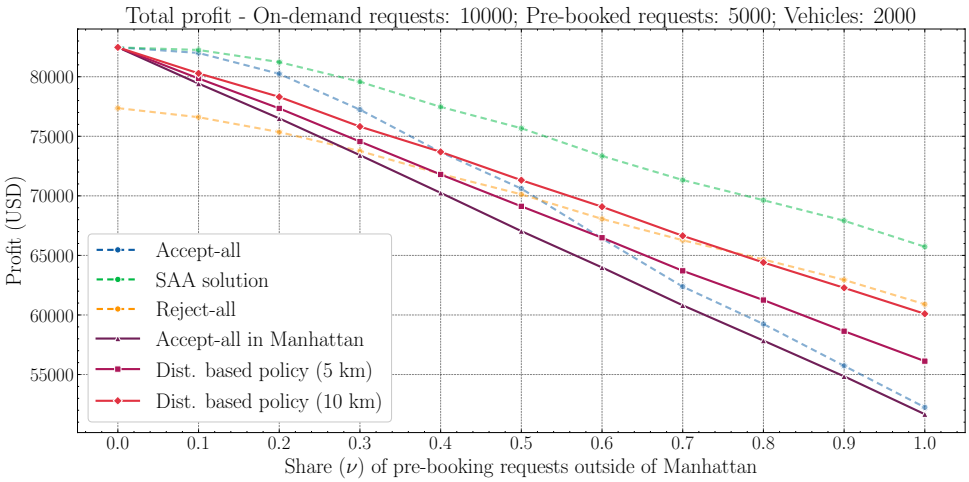


Figure 5.11: Performance of distance-based policies

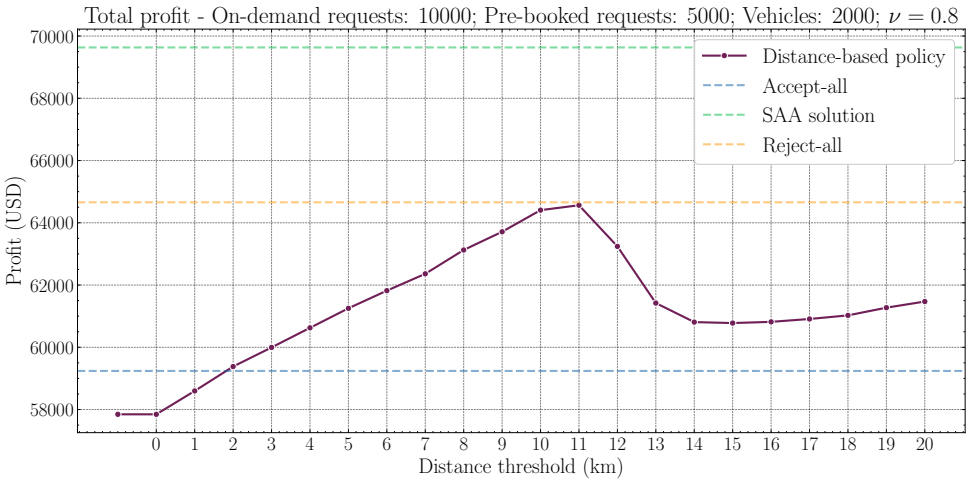


Figure 5.12: Threshold sensitivity analysis for $\nu = 0.8$

Figure 5.11 shows the profit performance of the heuristic policies compared to the SAA solution and the baseline policies. In general, it is more profitable to adopt the SAA model. However, in cases where the operator requires explainability, the distance-based policy with a threshold of 10 km provides a compromise between reject-all and accept-all policies, being robust to distributional shifts. To complement these results, we performed a sensitivity analysis to determine the optimal threshold value for the distance-based heuristic. Figure 5.12 illustrates the profit performance of the distance-based heuristic as we vary the distance threshold for a setting with $\nu = 0.8$. We observe that setting the distance threshold to 11 km provides the most profit in this setting. We

present further sensitivity analyses regarding the distance threshold for different settings in Appendix 5.C.2. In general, we observe that a distance threshold between 10 and 11 km is optimal across different settings.

Result 5.11 *A distance-based policy offers a compromise in performance between the reject-all and accept-all policies, having the benefit of being more explainable than the SAA model and simpler to implement in practice.*

5.6 Conclusions

In this paper, we investigated the performance of a mixed-service ride-hailing system, where customers have the option to either request a ride on demand or pre-book a ride in advance. To model this system, we introduced a two-stage stochastic optimization model and presented a sample average approximation (SAA) formulation. We developed a scalable heuristic algorithm that solves simplified subproblems separately and then determines first-stage decisions by reaching a consensus based on majority voting. We evaluated the performance of the SAA model by comparing its solutions to baseline policies in computational experiments. Our results showed that a mixed-service system can be more profitable than a purely on-demand system, regardless of distributional shifts. In addition, we provided managerial insights to operators employing a greedy accept-all policy, highlighting strategies to compensate for the effects of distributional shifts. We further analyzed the structural properties of our proposed solutions with visualizations that give an intuition into the system's behavior. Lastly, we showed that a distance-based heuristic policy offers a reasonable trade-off between performance, explainability, and ease of implementation.

This work raises several interesting directions for future research. First, we considered a setting in which all on-demand requests are revealed simultaneously, corresponding to the assumption of perfect foresight from the operator's perspective. Future research could model the problem in a dynamic setting, where the operator makes online decisions as information is revealed. Second, instead of assuming the set of requests to be given a priori, one could model the customer choices, i.e., between pre-booking, requesting a ride on demand, or opting for an alternative mode of transportation, based on prices and expected wait times. Third, although we focused on a centrally operated system, many ride-hailing platforms adopt a business model in which drivers independently choose to accept or reject customers. Future work could investigate the extent to which the insights from our work apply to a setting with decentralized decision-making. Fourth, future research could investigate how the distribution of pre-booked and on-demand rides would evolve over time. In particular, an open question is whether the travel distributions and

especially the distributional shift would have a long-term convergence. Finally, although we can reasonably expect a certain level of distributional shift in mixed-service systems, the collection of real-world travel data would provide a more accurate characterization of the travel distributions and empirical evidence of distributional shifts.

Acknowledgments

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the research group Advanced Optimization in a Networked Economy (AdONE, GRK2201/277991500).

Appendix 5.A Graph Pruning

We describe pruning strategies for creating a sparser graph $\hat{\mathcal{G}}$ from a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$.

Temporal pruning. We remove edges $(i, j) \in \mathcal{A}$ from the graph $\mathcal{G}$ for which the driver must wait more than $W_{\max}$ seconds to start serving a request:

$$s_j - (e_i + T_{ij}) > W_{\max} \tag{5.45}$$

S-Neighborhood pruning. Given a sparsity parameter S, we create a sparser graph $\hat{\mathcal{G}}$ by keeping the S-lowest cost out-going and incoming arcs for every node in $\mathcal{V}$ (cf. Bertsimas, Jaillet, and Martin 2019).

Appendix 5.B Discretization into Regions

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be the graph representing the road network of New York City. We follow the incremental pruning approach of Alonso-Mora et al. (2017) to discretize the road network into regions. Given a list C_0 of all vertices, we select a subset $C \subseteq C_0$ of vertices representing region centers such that no two centers are within a given radius r of each other, i.e., $||i - j|| > r$ for all $i, j \in \mathcal{V}'$. Algorithm 5.2 presents the incremental pruning procedure from Alonso-Mora et al. (2017), which uses the `BallTree` data structure to partition the space, allowing for fast radius bounded nearest neighbor lookup. The `BallTree` data structure allows us to query all points within a distance r of a point c using the `Query`(c, r) function. Figure 5.13 shows the region centers used in the case study, using a discretization radius of 200 meters.

Algorithm 5.2: Incremental pruning of region centers (Alonso-Mora et al., 2017)

Input: List C_0 of intersections from the road network and discretization radius r

$C = C_0$; $i = 0$
$\mathcal{T} \leftarrow$ `BallTree`(C_0)
while $i < |C|$ **do**
  $c_i = C[i]$ `// i-th element in C`
  $C \leftarrow C \setminus \mathcal{T}$.`Query`$(c_i, r)$
  $i = i + 1$

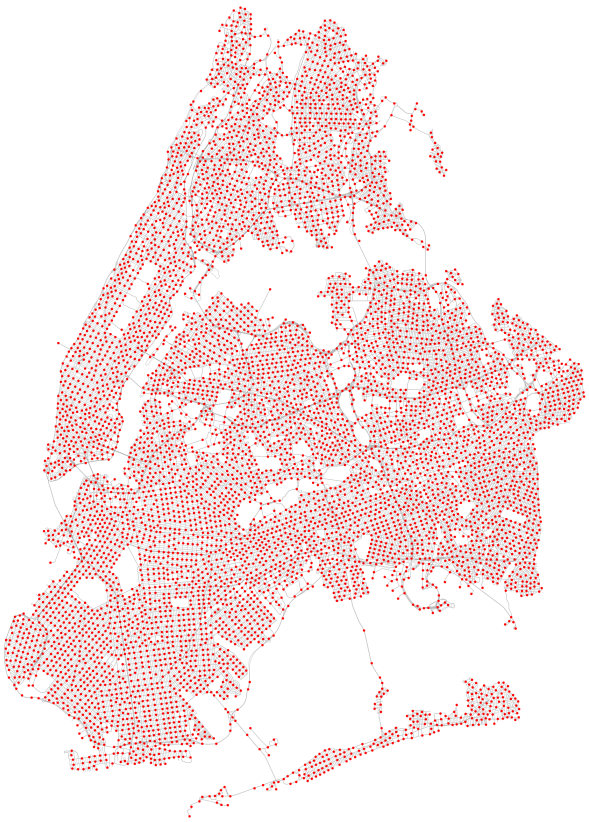


Figure 5.13: Region centers of the road network determined by incremental pruning

Appendix 5.C Extended Results

5.C.1 Analysis of First-stage Decisions

Accuracy performance of first-stage decisions. We assess the ability of each policy to recover the full-information first-stage solution, adopting the following accuracy measure:

$$\text{Accuracy} = \frac{1}{|D_1|} \sum_{i \in D_1} \mathbb{1}(x_i^* = \hat{x}_i) \tag{5.46}$$

where $\mathbb{1}$ is the indicator function, $\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_{|D_1|}]$ is the binary vector defining the policy's first-stage decisions, and $\mathbf{x}^* = [x_1^*, \ldots, x_{|D_1|}^*]$ is the vector of full-information first-stage decisions. Figure 5.14 shows accuracy values from heuristic solutions to the SAA model and the baseline policies as we increase the share parameter ν. The SAA solutions achieve accuracy values close 1, indicating that the model can easily make first-stage decisions that are close to the full-information solution. Since there are still many decisions that can be taken in the second-stage problem, the operator has much flexibility to compensate for the uncertainty stemming from on-demand requests. However, we emphasize again that our experiments are based on the assumption of perfect foresight, which provides a best-case analysis for the operator, and we can expect the results to differ in a dynamic setting with online decision-making.

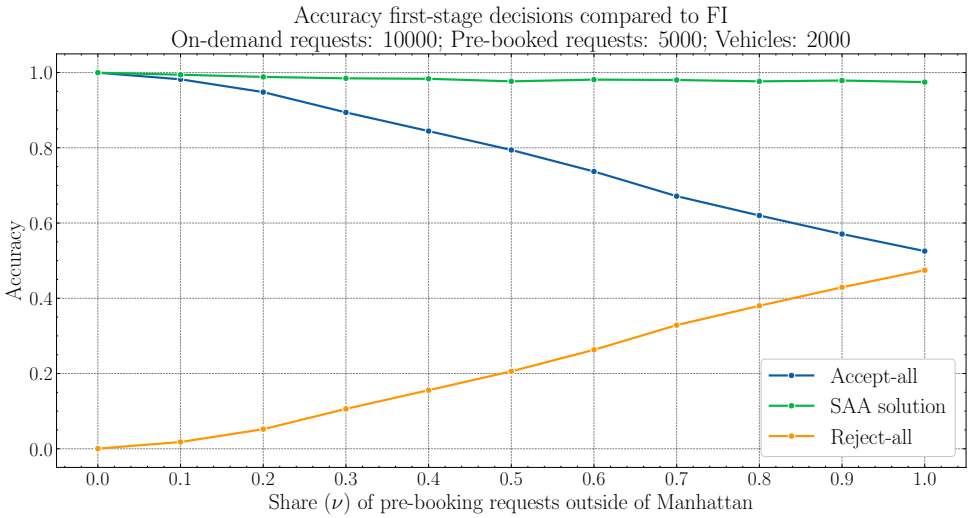


Figure 5.14: Accuracy performance of the heuristic solution compared to the full-information first-stage decisions (Large instance).

5.C.2 Sensitivity Analysis for Distance-based Policy

Figures 5.15 and 5.16 show the profit performance of the distance-based heuristic as we vary the distance threshold for settings with $\nu = 0.2$ and $\nu = 0.5$, respectively. Figure 5.17 aggregates in a single plot the profit performance of the distance-based policy for setting with $\nu \in [0, 1]$.

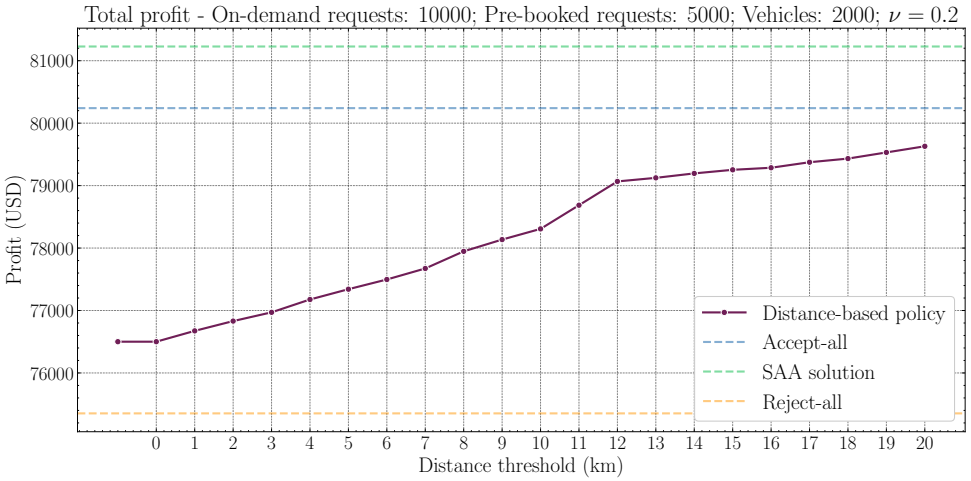


Figure 5.15: Threshold sensitivity analysis for $\nu = 0.2$.

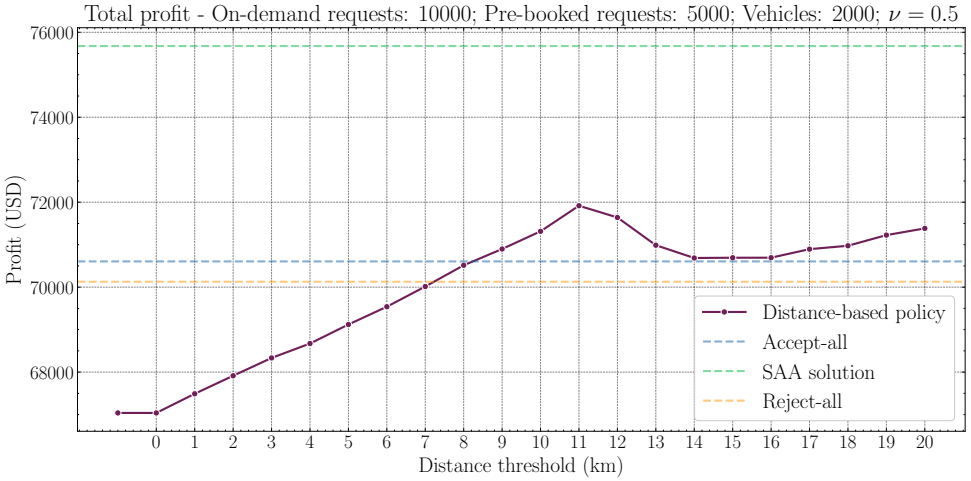


Figure 5.16: Threshold sensitivity analysis for $\nu = 0.5$.

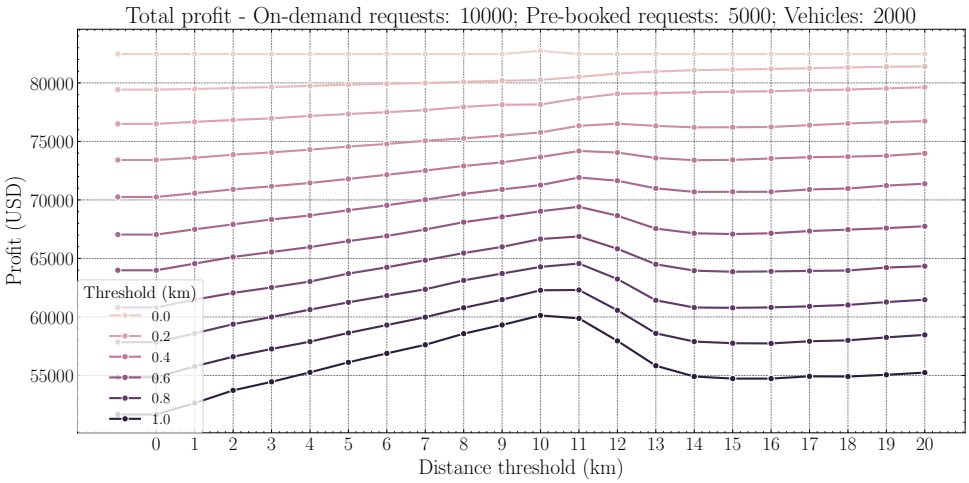


Figure 5.17: Threshold sensitivity analysis.

6 Conclusion

6.1 Summary

This thesis addressed research gaps in the literature on contextual and data-driven decision-making, particularly focusing on three key application areas within supply chain and transportation systems. The following provides a summary of each methodological chapter.

Chapter 3 proposed a novel formulation based on bilevel optimization that automatically performs feature selection while learning a decision function for the feature-based newsvendor problem. In the proposed Bilevel Feature Selection (BFS) models, the upper-level problem directly selects the subset of relevant features, while the lower level learns the optimal coefficients of the decision function, using only the features selected by the upper-level. Computational experiments on synthetic data showed that the proposed methods achieved accuracy performance above 96% regarding the recovery of ground-truth features, resulting in more explainable models compared to existing methods.

Chapter 4 bridged the gap between the well-established research field of vehicle routing problems (VRPs) with the emerging literature on contextual optimization, by introducing the novel contextual stochastic vehicle routing problem with time windows (VRPTW) formulation. The chapter explored several data-driven prescriptive models including novel methods based on conditional sample average approximation (CSAA) and penalty-based prediction. Customized branch-price-and-cut (BP&C) algorithms, adapted from state-of-the-art techniques, could solve instances with up to 100 customers. Computational results showed that while the penalty-based approximation model offered competitive solutions, the CSAA method generally achieved the smallest gaps compared to the full-information benchmark, providing up to a 13.2% reduction in test cost compared to the classical SAA baseline.

Chapter 5 considered a mixed-service ride-hailing system and introduced a novel two-stage stochastic optimization model in which the first-stage problem consists of deciding which pre-booking requests to accept before the uncertain on-demand requests enter the system. The chapter presented an SAA formulation and developed a scalable heuristic

algorithm that separately solves approximate subproblems and then reaches a consensus on first-stage decisions by majority voting. Computational results showed that optimizing the mixed-service system with the proposed SAA method lead to profit increases of up to 7.7% compared to the purely on-demand system, even in settings with strong distributional shifts. The chapter further discussed structural properties of the proposed solutions and managerial insights that may benefit ride-hailing operators.

In summary, this thesis contributed new models and methods that effectively addressed methodological challenges related to contextual and data-driven decision-making in the areas of inventory management, logistics, and urban mobility.

6.2 Limitations and Perspectives

While this thesis makes several contributions, some aspects remained out of scope. In the following, we discuss specific limitations related to each chapter and outline possible avenues for future research.

In Chapter 3, while our computational experiments demonstrated the superiority of the proposed methods for feature recovery, a few comments are in order regarding their applicability to real-world problems. First, the computational experiments were carried out using synthetic data due to the lack of an accessible real-world data set containing a sufficiently large set of features. Further research could investigate the effectiveness of the proposed methods on real data and evaluate the applicability of the findings to a real-world setting. Second, the proposed BFS and BFS with cross-validation (BFS-CV) may require tailored solution methods in order to scale to instances with a large number of features. Accordingly, future work could investigate the application of decomposition strategies from mixed integer programming in this context. Third, to complement our experimental results, an interesting research direction could be to investigate theoretical properties of the BFS models, e.g., regarding asymptotic convergence or theoretical performance guarantees for out-of-sample data.

Chapter 4 provides several possibilities for follow-up works, two of which we discuss next. First, as in Chapter 3, the computational experiments were based on synthetic data. Therefore, future research could extend the experimental design to real-world data. We note that these experiments may involve considerable effort in data collection, as it would require not only the street network of a real city, but also a data set containing travel times on each street segment and relevant feature observations. Accordingly, it may be necessary to combine multiple separate data sources to build the feature data, which could include information related to the weather, road works, or events in the city, among others. Second, future research could investigate variations of the presented methods, e.g.,

extending the CSAA and residual-based sample average approximation (RSAA) methods with non-linear probabilistic models, to more accurately capture the complex non-linear relationships present in real-world settings.

Chapter 5 focused on a relatively new problem setting, offering several opportunities for future research. We discuss three potential extensions of the proposed problem setting. First, the assumption of a centrally-operated system may not reflect how most ride-hailing platforms operate. In reality, drivers often have the autonomy to decide whether to accept or reject ride requests and routes suggested by the operator. Therefore, future work could investigate to what extent the insights gained from this centralized setting translate to settings with decentralized decision-making. Second, follow-up works could investigate different pricing strategies for pre-booking requests. Third, future work could model how customers choose between pre-booking or on-demand rides considering factors such as the prices and expected wait times corresponding to each service.

Finally, it is worth noting that all problem settings studied in this thesis are static. Accordingly, extending the methods presented in this thesis to dynamic settings, where decisions are made in an online fashion in response to revealed information, appears as a natural next step.

Bibliography

Abkarian, Hoseb, Hani S. Mahmassani, and Michael Hyland (2022). "Modeling the mixed-service fleet problem of shared-use autonomous mobility systems for on-demand ridesourcing and carsharing with reservations". In: *Transportation Research Record* 2676(8), pp. 363–375. DOI: 10.1177/03611981221083617.

Adulyasak, Yossiri and Patrick Jaillet (May 2016). "Models and Algorithms for Stochastic and Robust Vehicle Routing with Deadlines". In: *Transportation Science* 50(2), pp. 608–626. DOI: 10.1287/trsc.2014.0581.

Afèche, Philipp (2013). "Incentive-compatible revenue management in queueing systems: Optimal strategic delay". In: *Manufacturing & Service Operations Management* 15(3), pp. 423–443. DOI: 10.1287/msom.2013.0449.

Agatz, Niels, Alan Erera, Martin Savelsbergh, and Xing Wang (2012). "Optimization for dynamic ride-sharing: A review". In: *European Journal of Operational Research* 223(2), pp. 295–303. DOI: 10.1016/j.ejor.2012.05.028.

Agor, Joseph and Osman Y. Özaltın (2019). "Feature selection for classification models via bilevel optimization". In: *Computers & Operations Research* 106, pp. 156–168. ISSN: 0305-0548. DOI: 10.1016/j.cor.2018.05.005.

Alonso-Mora, Javier, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus (2017). "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment". In: *Proceedings of the National Academy of Sciences* 114(3), pp. 462–467. DOI: 10.1073/pnas.1611675114.

Arlot, Sylvain and Alain Celisse (2010). "A survey of cross-validation procedures for model selection". In: *Statistics Surveys* 4, pp. 40–79. DOI: 10.1214/09-SS054.

Ashlagi, Itai, Maximilien Burq, Chinmoy Dutta, Patrick Jaillet, Amin Saberi, and Chris Sholley (2022). "Edge-weighted online windowed matching". In: *Mathematics of Operations Research* 48(2), pp. 999–1016. DOI: 10.1287/moor.2022.1289.

Bai, Jiaru, Kut C. So, Christopher S. Tang, Xiqun Chen, and Hai Wang (2019). "Coordinating supply and demand on an on-demand service platform with impatient customers". In: *Manufacturing & Service Operations Management* 21(3), pp. 556–570. DOI: 10.1287/msom.2018.0707.

Ban, Gah-Yi (2020). "Confidence intervals for data-driven inventory policies with demand censoring". In: *Operations Research* 68(2), pp. 309–326. DOI: 10.1287/opre.2019.1883.

Ban, Gah-Yi, Jérémie Gallien, and Adam J. Mersereau (2019). "Dynamic procurement of new products with covariate information: The residual tree method". In: *Manufacturing & Service Operations Management* 21(4), pp. 798–815. DOI: 10.1287/msom.2018.0725.

Ban, Gah-Yi and Cynthia Rudin (2019). "The big data newsvendor: Practical insights from machine learning". In: *Operations Research* 67(1), pp. 90–108. DOI: 10.1287/opre.2018.1757.

Banerjee, Siddhartha, Daniel Freund, and Thodoris Lykouris (2021). "Pricing and optimization in shared vehicle systems: An approximation framework". In: *Operations Research* 70(3), pp. 1783–1805. DOI: 10.1287/opre.2021.2165.

Barnhart, Cynthia, Ellis L. Johnson, George L. Nemhauser, Martin W.P. Savelsbergh, and Pamela H. Vance (June 1998). "Branch-and-Price: Column Generation for Solving Huge Integer Programs". In: *Operations Research* 46(3), pp. 316–329. DOI: 10.1287/opre.46.3.316.

Baty, Léo, Kai Jungel, Patrick S. Klein, Axel Parmentier, and Maximilian Schiffer (2024). "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows". In: *Transportation Science* 58(4), pp. 708–725. DOI: 10.1287/trsc.2023.0107.

Bektaş, Tolga, Panagiotis P. Repoussis, and Christos D. Tarantilis (2014). "Chapter 11: Dynamic Vehicle Routing Problems". In: *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. SIAM. Chap. 11, pp. 299–347. DOI: 10.1137/1.9781611973594.ch11.

Ben-Tal, Aharon, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen (2013). "Robust solutions of optimization problems affected by uncertain probabilities". In: *Management Science* 59(2), pp. 341–357. DOI: 10.1287/mnsc.1120.1641.

Benati, Stefano and Sergio García (2014). "A mixed integer linear model for clustering with variable selection". In: *Computers & Operations Research* 43, pp. 280–285. DOI: 10.1016/j.cor.2013.10.005.

Bengio, Yoshua, Andrea Lodi, and Antoine Prouvost (2021). "Machine learning for combinatorial optimization: a methodological tour d'horizon". In: *European Journal of Operational Research* 290(2), pp. 405–421. DOI: 10.1016/j.ejor.2020.07.063.

Bennett, Kristin P., Jing Hu, Xiaoyun Ji, Gautam Kunapuli, and Jong-Shi Pang (2006). "Model selection via bilevel optimization". In: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1922–1929. DOI: 10.1109/IJCNN.2006.246935.

Bennett, Kristin P., Gautam Kunapuli, Jing Hu, and Jong-Shi Pang (June 2008). "Bilevel Optimization and Machine Learning". In: *Computational Intelligence: Research Frontiers: IEEE World Congress on Computational Intelligence (WCCI), Hong Kong, China*. Ed. by Jacek M. Zurada, Gary G. Yen, and Jun Wang. Springer, Berlin, Heidelberg, pp. 25–47. DOI: 10.1007/978-3-540-68860-0_2.

Bent, Russell W. and Pascal Van Hentenryck (2004). "Scenario-based planning for partially dynamic vehicle routing with stochastic customers". In: *Operations Research* 52(6), pp. 977–987. DOI: 10.1287/opre.1040.0124.

Beraldi, Patrizia, Maria Elena Bruni, Demetrio Laganà, and Roberto Musmanno (2015). "The mixed capacitated general routing problem under uncertainty". In: *European Journal of Operational Research* 240(2), pp. 382–392. DOI: 10.1016/j.ejor.2014.07.023.

Berbeglia, Gerardo, Jean-François Cordeau, and Gilbert Laporte (2010). "Dynamic pickup and delivery problems". In: *European Journal of Operational Research* 202(1), pp. 8–15. DOI: 10.1016/j.ejor.2009.04.024.

Berbeglia, Gerardo, Jean-François Cordeau, and Gilbert Laporte (2012). "A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem". In: *INFORMS Journal on Computing* 24(3), pp. 343–355. DOI: 10.1287/ijoc.1110.0454.

Bergstra, James and Yoshua Bengio (2012). "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13(10), pp. 281–305. URL: http://jmlr.org/papers/v13/bergstra12a.html.

Bergstra, James, Daniel Yamins, and David Cox (June 2013). "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research. PMLR: Atlanta, Georgia, USA, pp. 115–123. URL: https://proceedings.mlr.press/v28/bergstra13.html.

Bertsimas, Dimitris, Vishal Gupta, and Nathan Kallus (2018). "Robust sample average approximation". In: *Mathematical Programming* 171(1), pp. 217–282. DOI: 10.1007/s10107-017-1174-z.

Bertsimas, Dimitris, Patrick Jaillet, and Sébastien Martin (2019). "Online vehicle routing: The edge of optimization in large-scale applications". In: *Operations Research* 67(1), pp. 143–162. DOI: 10.1287/opre.2018.1763.

Bertsimas, Dimitris and Nathan Kallus (2019). "From predictive to prescriptive analytics". In: *Management Science* 66(3), pp. 1025–1044. DOI: 10.1287/mnsc.2018.3253.

Bertsimas, Dimitris, Angela King, and Rahul Mazumder (2016). "Best subset selection via a modern optimization lens". In: *The Annals of Statistics* 44(2), pp. 813 –852. DOI: 10.1214/15-AOS1388.

Bertsimas, Dimitris and Christopher McCord (2018). "Optimization over Continuous and Multi-dimensional Decisions with Observational Data". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc.

Bertsimas, Dimitris and Christopher McCord (2019). "From predictions to prescriptions in multistage optimization problems". In: *arXiv preprint arXiv:1904.11637*. DOI: 10.48550/arXiv.1904.11637.

Bertsimas, Dimitris, Christopher McCord, and Bradley Sturt (2023). "Dynamic optimization with side information". In: *European Journal of Operational Research* 304(2), pp. 634–651. DOI: 10.1016/j.ejor.2022.03.030.

Bertsimas, Dimitris and Aurélie Thiele (2005). *A data-driven approach to newsvendor problems*. Tech. rep. Cambridge, MA: Massachusetts Institute of Technology.

Bertsimas, Dimitris and Aurélie Thiele (2006). "A robust optimization approach to inventory theory". In: *Operations Research* 54(1), pp. 150–168. DOI: 10.1287/opre.1050.0238.

Besbes, Omar and Omar Mouchtaki (2023). "How big should your data really be? Data-driven newsvendor: Learning one sample at a time". In: *Management Science* 69(10), pp. 5848–5865. DOI: 10.1287/mnsc.2023.4725.

Besbes, Omar and Alp Muharremoglu (2013). "On implications of demand censoring in the newsvendor problem". In: *Management Science* 59(6), pp. 1407–1424. DOI: 10.1287/mnsc.1120.1654.

Beutel, Anna-Lena and Stefan Minner (2012). "Safety stock planning under causal demand forecasting". In: *International Journal of Production Economics* 140(2), pp. 637–645. DOI: 10.1016/j.ijpe.2011.04.017.

Bilali, Aledia, Florian Dandl, Ulrich Fastenrath, and Klaus Bogenberger (2019). "An analytical model for on-demand ride sharing to evaluate the impact of reservation, detour and maximum waiting time". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 1715–1720. DOI: 10.1109/ITSC.2019.8917280.

Bolt (2023). *Introducing Scheduled Rides – a convenient way to book upcoming trips*. https://bolt.eu/en/blog/scheduled-rides/. [Online; accessed 21-May-2024].

Bösch, Patrick M., Felix Becker, Henrik Becker, and Kay W. Axhausen (2018). "Cost-based analysis of autonomous mobility services". In: *Transport Policy* 64, pp. 76–91. DOI: 10.1016/j.tranpol.2017.09.005.

Braverman, Anton, Jim G. Dai, Xin Liu, and Lei Ying (2019). "Empty-car routing in ridesharing systems". In: *Operations Research* 67(5), pp. 1437–1452. DOI: 10.1287/opre.2018.1822.

Cao, Dong and Mingyuan Chen (2006). "Capacitated plant selection in a decentralized manufacturing environment: A bilevel optimization approach". In: *European Journal of Operational Research* 169(1), pp. 97–110. DOI: 10.1016/j.ejor.2004.05.016.

Chen, Xin, Melvyn Sim, David Simchi-Levi, and Peng Sun (2007). "Risk aversion in inventory management". In: *Operations Research* 55(5), pp. 828–842. DOI: 10.1287/opre.1070.0429.

Cheung, Wang Chi and David Simchi-Levi (2019). "Sampling-based approximation schemes for capacitated stochastic inventory control models". In: *Mathematics of Operations Research* 44(2), pp. 668–692. DOI: 10.1287/moor.2018.0940.

Choi, Tsan-Ming (2012). *Handbook of newsvendor problems: Models, extensions and applications*. Vol. 176. Springer, New York. DOI: 10.1007/978-1-4614-3600-3.

Chopra, Sunil, Sebastien Martin, Partha Sarathi Mishra, and Karen Smilowitz (2023). "Mobility-on-Demand Meets Shuttles on the Same Mile". In: *Available at SSRN 4322824*. URL: https://ssrn.com/abstract=4322824.

Cordeau, Jean-Francois, Michel Gendreau, Gilbert Laporte, Jean-Yves Potvin, and Frédéric Semet (2002). "A guide to vehicle routing heuristics". In: *Journal of the Operational Research Society* 53(5), pp. 512–522. DOI: 10.1057/palgrave.jors.2601319.

Cordeau, Jean-François and Gilbert Laporte (2007). "The dial-a-ride problem: models and algorithms". In: *Annals of Operations Research* 153, pp. 29–46. DOI: 10.1007/s10479-007-0170-8.

Costa, Luciano, Claudio Contardo, and Guy Desaulniers (July 2019). "Exact Branch-Price-and-Cut Algorithms for Vehicle Routing". In: *Transportation Science* 53(4), pp. 946–985. DOI: 10.1287/trsc.2018.0878.

Cramer, Judd and Alan B. Krueger (2016). "Disruptive Change in the Taxi Business: The Case of Uber". In: *American Economic Review* 106(5), pp. 177–182. DOI: 10.1257/aer.p20161002.

Dabia, Said, Stefan Ropke, Tom Van Woensel, and Ton De Kok (2013). "Branch and price for the time-dependent vehicle routing problem with time windows". In: *Transportation Science* 47(3), pp. 380–396. DOI: 10.1287/trsc.1120.0445.

Dalle, Guillaume, Léo Baty, Louis Bouvier, and Axel Parmentier (2022). "Learning with combinatorial optimization layers: a probabilistic approach". In: *arXiv preprint arXiv:2207.13513*. DOI: 10.48550/arXiv.2207.13513.

Desaulniers, Guy, Jacques Desrosiers, and Simon Spoorendonk (2011). "Cutting planes for branch-and-price algorithms". In: *Networks* 58(4), pp. 301–310. DOI: 10.1002/net.20471.

Dickerson, John P., Karthik A. Sankararaman, Aravind Srinivasan, and Pan Xu (June 2021). "Allocation Problems in Ride-sharing Platforms: Online Matching with Offline Reusable Resources". In: *ACM Transactions on Economics and Computation* 9(3). ISSN: 2167–8375. DOI: 10.1145/3456756.

Dinh, Thai, Ricardo Fukasawa, and James Luedtke (2018). "Exact algorithms for the chance-constrained vehicle routing problem". In: *Mathematical Programming* 172(1), pp. 105–138. DOI: 10.1007/s10107-017-1151-6.

Donti, Priya L., Brandon Amos, and J. Zico Kolter (2017). "Task-based end-to-end model learning in stochastic optimization". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Curran Associates Inc.: Long Beach, California, USA, 5490–5500. ISBN: 9781510860964.

Duan, Leyi, Yuguang Wei, Jinchuan Zhang, and Yang Xia (2020). "Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode". In: *Transportation Research Part C: Emerging Technologies* 111, pp. 397–420. DOI: 10.1016/j.trc.2019.12.020.

Elmachtoub, Adam N. and Paul Grigas (2021). "Smart "predict, then optimize"". In: *Management Science* 68(1), pp. 9–26. DOI: 10.1287/mnsc.2020.3922.

Elting, Steffen and Jan Fabian Ehmke (2021). "Potential of shared taxi services in rural areas – a case study". In: *Transportation Research Procedia* 52, pp. 661–668. DOI: 10.1016/j.trpro.2021.01.079.

Enders, Tobias, James Harrison, Marco Pavone, and Maximilian Schiffer (June 2023). "Hybrid multi-agent deep reinforcement learning for autonomous mobility on demand systems". In: *Proceedings of the 5th Annual Learning for Dynamics & Control Conference*. Ed. by Nikolai Matni, Manfred Morari, and George J. Pappas. Vol. 211. Proceedings of Machine Learning Research. PMLR, pp. 1284–1296. URL: https://proceedings.mlr.press/v211/enders23a.html.

Engelhardt, Roman, Florian Dandl, and Klaus Bogenberger (2022). "Simulating ride-pooling services with pre-booking and on-demand customers". In: *arXiv preprint arXiv:2210.06972*. DOI: 10.48550/arXiv.2210.06972.

Errico, Fausto, Guy Desaulniers, Michel Gendreau, Walter Rei, and Louis-Martin Rousseau (2018). "The vehicle routing problem with hard time windows and stochastic service times". In: *EURO Journal on Transportation and Logistics* 7(3), pp. 223–251. DOI: 10.1007/s13676-016-0101-4.

Feillet, Dominique (2010). "A tutorial on column generation and branch-and-price for vehicle routing problems". In: *4or* 8(4), pp. 407–424. DOI: 10.1007/s10288-010-0130-z.

Feng, Guiyun, Guangwen Kong, and Zizhuo Wang (2020). "We are on the way: Analysis of on-demand ride-hailing systems". In: *Manufacturing & Service Operations Management* 23(5), pp. 1237–1256. DOI: 10.1287/msom.2020.0880.

Fletcher, Roger and Sven Leyffer (2002). "Nonlinear programming without a penalty function". In: *Mathematical Programming* 91(2), pp. 239–269. DOI: 10.1007/s101070100244.

Florio, Alexandre M., Richard F. Hartl, and Stefan Minner (2020). "New exact algorithm for the vehicle routing problem with stochastic demands". In: *Transportation Science* 54(4), pp. 1073–1090. DOI: 10.1287/trsc.2020.0976.

Fontaine, Pirmin and Stefan Minner (2014). "Benders decomposition for discrete–continuous linear bilevel problems with application to traffic network design". In: *Transportation Research Part B: Methodological* 70, pp. 163–172. DOI: 10.1016/j.trb.2014.09.007.

Franceschi, Luca, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil (July 2018). "Bilevel programming for hyperparameter optimization and meta-learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR: Stockholm, Sweden, pp. 1568–1577. URL: https://proceedings.mlr.press/v80/franceschi18a.html.

Furuhata, Masabumi, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig (2013). "Ridesharing: The state-of-the-art and future directions". In: *Transportation Research Part B: Methodological* 57, pp. 28–46. ISSN: 0191-2615. DOI: 10.1016/j.trb.2013.08.012.

Gallego, Guillermo and Ilkyeong Moon (1993). "The distribution free newsboy problem: Review and extensions". In: *Journal of the Operational Research Society* 44(8), pp. 825–834. DOI: 10.1057/jors.1993.141.

Gallego, Guillermo, Jennifer K. Ryan, and David Simchi-Levi (2001). "Minimax analysis for finite-horizon inventory models". In: *IIE Transactions* 33(10), pp. 861–874. DOI: 10.1080/07408170108936879.

Gammelli, Daniele, Kaidi Yang, James Harrison, Filipe Rodrigues, Francisco C. Pereira, and Marco Pavone (2021). "Graph neural network reinforcement learning for autonomous mobility-on-demand systems". In: *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2996–3003. DOI: 10.1109/CDC45484.2021.9683135.

Garg, Nikhil and Hamid Nazerzadeh (2022). "Driver surge pricing". In: *Management Science* 68(5), pp. 3219–3235. DOI: 10.1287/mnsc.2021.4058.

Gauvin, Charles, Guy Desaulniers, and Michel Gendreau (2014). "A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands". In: *Computers & Operations Research* 50, pp. 141–153. ISSN: 0305-0548. DOI: 10.1016/j.cor.2014.03.028.

Gendreau, Michel, Gianpaolo Ghiani, and Emanuela Guerriero (2015). "Time-dependent routing problems: A review". In: *Computers & Operations Research* 64, pp. 189–197. DOI: 10.1016/j.cor.2015.06.001.

Gendreau, Michel, François Guertin, Jean-Yves Potvin, and Éric Taillard (1999). "Parallel tabu search for real-time vehicle routing and dispatching". In: *Transportation science* 33(4), pp. 381–390. DOI: 10.1287/trsc.33.4.381.

Gendreau, Michel, Ola Jabali, and Walter Rei (2014). "Chapter 8: Stochastic vehicle routing problems". In: *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. SIAM, pp. 213–239. DOI: 10.1137/1.9781611973594.ch8.

Ghosal, Shubhechyya, Chin Pang Ho, and Wolfram Wiesemann (2024). "A unifying framework for the capacitated vehicle routing problem under risk and ambiguity". In: *Operations Research* 72(2), pp. 425–443. DOI: 10.1287/opre.2021.0669.

Ghosal, Shubhechyya and Wolfram Wiesemann (2020). "The distributionally robust chance-constrained vehicle routing problem". In: *Operations Research* 68(3), pp. 716–732. DOI: 10.1287/opre.2019.1924.

Gmira, Maha, Michel Gendreau, Andrea Lodi, and Jean-Yves Potvin (2021). "Tabu search for the time-dependent vehicle routing problem with time windows on a road network". In: *European Journal of Operational Research* 288(1), pp. 129–140. DOI: 10.1016/j.ejor.2020.05.041.

Golden, Bruce L., Subramanian Raghavan, and Edward A. Wasil (2008). *The vehicle routing problem: latest advances and new challenges*. Vol. 43. Springer, New York. DOI: 10.1007/978-0-387-77778-8.

Gómez, Andrés and Oleg A. Prokopyev (2021). "A mixed-integer fractional optimization approach to best subset selection". In: *INFORMS Journal on Computing* 33(2), pp. 551–565. DOI: 10.1287/ijoc.2020.1031.

Gounaris, Chrysanthos E., Wolfram Wiesemann, and Christodoulos A. Floudas (2013). "The robust capacitated vehicle routing problem under demand uncertainty". In: *Operations Research* 61(3), pp. 677–693. DOI: 10.1287/opre.1120.1136.

Guyon, Isabelle and André Elisseeff (2003). "An introduction to variable and feature selection". In: *Journal of Machine Learning Research* 3, pp. 1157–1182. URL: https://dl.acm.org/doi/abs/10.5555/944919.944968.

Hanasusanto, Grani A., Daniel Kuhn, Stein W. Wallace, and Steve Zymler (2015). "Distributionally robust multi-item newsvendor problems with multimodal demand distributions". In: *Mathematical Programming* 152(1), pp. 1–32. DOI: 10.1007/s10107-014-0776-y.

Hassin, Refael and Moshe Haviv (2003). *To queue or not to queue: Equilibrium behavior in queueing systems*. Vol. 59. Springer Science & Business Media.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Vol. 2. Springer, New York. DOI: 10.1007/978-0-387-21606-5.

Hildebrandt, Florentin D., Barrett W. Thomas, and Marlin W. Ulmer (2023). "Opportunities for reinforcement learning in stochastic dynamic vehicle routing". In: *Computers & Operations Research* 150, p. 106071. DOI: 10.1016/j.cor.2022.106071.

Hoppe, Heiko, Tobias Enders, Quentin Cappart, and Maximilian Schiffer (July 2024). "Global rewards in multi-agent deep reinforcement learning for autonomous mobility on demand systems". In: *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*. Ed. by Alessandro Abate, Mark

Cannon, Kostas Margellos, and Antonis Papachristodoulou. Vol. 242. Proceedings of Machine Learning Research. PMLR, pp. 260–272. URL: https://proceedings.mlr.press/v242/hoppe24a.html.

Hu, Ming and Yun Zhou (2020). "Price, wage, and fixed commission in on-demand matching". In: *Available at SSRN 2949513*.

Huber, Jakob, Sebastian Müller, Moritz Fleischmann, and Heiner Stuckenschmidt (2019). "A data-driven newsvendor problem: From data to decision". In: *European Journal of Operational Research* 278(3), pp. 904–915. DOI: 10.1016/j.ejor.2019.04.043.

Iglesias, Ramon, Federico Rossi, Kevin Wang, David Hallac, Jure Leskovec, and Marco Pavone (2018). "Data-driven model predictive control of autonomous mobility-on-demand systems". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 6019–6025. DOI: 10.1109/ICRA.2018.8460966.

Jaillet, Patrick, Jin Qi, and Melvyn Sim (2016). "Routing optimization under uncertainty". In: *Operations Research* 64(1), pp. 186–200. DOI: 10.1287/opre.2015.1462.

Jiao, Yan, Xiaocheng Tang, Zhiwei (Tony) Qin, Shuaiji Li, Fan Zhang, Hongtu Zhu, and Jieping Ye (2021). "Real-world ride-hailing vehicle repositioning using deep reinforcement learning". In: *Transportation Research Part C: Emerging Technologies* 130, p. 103289. ISSN: 0968-090X. DOI: 10.1016/j.trc.2021.103289.

Joe, Waldy and Hoong Chuin Lau (2020). "Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 30, pp. 394–402. DOI: 10.1609/icaps.v30i1.6685.

Jungel, Kai, Axel Parmentier, Maximilian Schiffer, and Thibaut Vidal (2023). "Learning-based Online Optimization for Autonomous Mobility-on-Demand Fleet Control". In: *arXiv preprint arXiv:2302.03963*.

Kenyon, Astrid S. and David P. Morton (2003). "Stochastic vehicle routing with random travel times". In: *Transportation Science* 37(1), pp. 69–82. DOI: 10.1287/trsc.37.1.69.12820.

Khouja, Moutaz (1999). "The single-period (news-vendor) problem: Literature review and suggestions for future research". In: *Omega* 27(5), pp. 537–553. DOI: 10.1016/S0305-0483(99)00017-1.

Kimura, Keiji and Hayato Waki (2018). "Minimization of Akaike's information criterion in linear regression analysis via mixed integer nonlinear program". In: *Optimization Methods and Software* 33(3), pp. 633–649. DOI: 10.1080/10556788.2017.1333611.

Kleywegt, Anton J., Alexander Shapiro, and Tito Homem-de Mello (2002). "The sample average approximation method for stochastic discrete optimization". In: *SIAM Journal on Optimization* 12(2), pp. 479–502. DOI: 10.1137/S1052623499363220.

Kogan, Konstantin and Sheldon Lou (2003). "Multi-stage newsboy problem: A dynamic model". In: *European Journal of Operational Research* 149(2), pp. 448–458. DOI: 10.1016/S0377-2217(02)00450-2.

Kotary, James, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder (Aug. 2021). "End-to-End Constrained Optimization Learning: A Survey". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. Ed. by Zhi-Hua Zhou. International Joint Conferences on Artificial Intelligence Organization, pp. 4475–4482. DOI: 10.24963/ijcai.2021/610.

Kuhn, Max and Kjell Johnson (2019). *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC. DOI: 10.1201/9781315108230.

Lamotte, Raphaël, Andre De Palma, and Nikolas Geroliminis (2017). "On the use of reservation-based autonomous vehicles for demand management". In: *Transportation Research Part B: Methodological* 99, pp. 205–227. DOI: 10.1016/j.trb.2017.01.003.

Laporte, Gilbert (2009). "Fifty years of vehicle routing". In: *Transportation Science* 43(4), pp. 408–416. DOI: 10.1287/trsc.1090.0301.

Laporte, Gilbert, Francois Louveaux, and Hélène Mercure (1992). "The vehicle routing problem with stochastic travel times". In: *Transportation Science* 26(3), pp. 161–170. DOI: 10.1287/trsc.26.3.161.

Larsen, Allan, O.B.G.D. Madsen, and Marius Solomon (2002). "Partially dynamic vehicle routing—models and algorithms". In: *Journal of the Operational Research Society* 53, pp. 637–646. DOI: 10.1057/palgrave.jors.2601352.

Larson, Richard C. and Amedeo R. Odoni (1981). *Urban operations research*. Prentice-Hall.

Lau, Hon-Shiang and Amy Hing-Ling Lau (1996). "The newsstand problem: A capacitated multiple-product single-period inventory problem". In: *European Journal of Operational Research* 94(1), pp. 29–42. DOI: 10.1016/0377-2217(95)00192-1.

Lee, Chungmok, Kyungsik Lee, and Sungsoo Park (2012). "Robust vehicle routing problem with deadlines and travel time/demand uncertainty". In: *Journal of the Operational Research Society* 63(9), pp. 1294–1306. DOI: 10.1057/jors.2011.136.

Levi, Retsef, Georgia Perakis, and Joline Uichanco (2015). "The data-driven newsvendor problem: New bounds and insights". In: *Operations Research* 63(6), pp. 1294–1306. DOI: 10.1287/opre.2015.1422.

Levi, Retsef, Robin O. Roundy, and David B. Shmoys (2007). "Provably near-optimal sampling-based policies for stochastic inventory control models". In: *Mathematics of Operations Research* 32(4), pp. 821–839. DOI: 10.1287/moor.1070.0272.

Levin, Michael W. (2017). "Congestion-aware system optimal route choice for shared autonomous vehicles". In: *Transportation Research Part C: Emerging Technologies* 82, pp. 229–247. DOI: 10.1016/j.trc.2017.06.020.

Li, Xiangyong, Peng Tian, and Stephen C.H. Leung (2010). "Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm". In: *International Journal of Production Economics* 125(1), pp. 137–145. DOI: 10.1016/j.ijpe.2010.01.013.

Liang, Enming, Kexin Wen, William H.K. Lam, Agachai Sumalee, and Renxin Zhong (2022). "An integrated reinforcement learning and centralized programming approach for online taxi dispatching". In: *IEEE Transactions on Neural Networks and Learning Systems* 33(9), pp. 4742–4756. DOI: 10.1109/TNNLS.2021.3060187.

Litman, Todd (2020). *Autonomous vehicle implementation predictions: Implications for transport planning*. Tech. rep.

Liu, Sheng, Long He, and Zuo-Jun Max Shen (July 2021). "On-Time Last-Mile Delivery: Order Assignment with Travel-Time Predictors". In: *Management Science* 67(7), pp. 4095–4119. DOI: 10.1287/mnsc.2020.3741.

Liu, Yang and Samitha Samaranayake (2020). "Proactive rebalancing and speed-up techniques for on-demand high capacity ridesourcing services". In: *IEEE Transactions on Intelligent Transportation Systems* 23(2), pp. 819–826. DOI: 10.1109/TITS.2020.3016128.

Lokhandwala, Mustafa and Hua Cai (2018). "Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC". In: *Transportation Research Part C: Emerging Technologies* 97, pp. 45–60. DOI: 10.1016/j.trc.2018.10.007.

Lund, Karsten, Oli B.G. Madsen, and Jens Moberg Rygaard (1996). *Vehicle routing problems with varying degrees of dynamism*. IMM, Institute of Mathematical Modelling, Technical University of Denmark.

Lyft (2023). *Our on-time pickup promise*. https://www.lyft.com/blog/posts/our-on-time-pickup-promise. [Online; accessed 21-May-2024].

Lyu, Guodong, Wang Chi Cheung, Chung-Piaw Teo, and Hai Wang (2019). "Multi-objective online ride-matching". In: *Available at SSRN* 3356823.

Ma, Hongyao, Fei Fang, and David C. Parkes (2021). "Spatio-temporal pricing for ridesharing platforms". In: *Operations Research* 70(2), pp. 1025–1041. DOI: 10.1287/opre.2021.2178.

Ma, Jiaqi, Xiaopeng Li, Fang Zhou, and Wei Hao (2017). "Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach". In: *Transportation Research Part C: Emerging Technologies* 84, pp. 124–141. DOI: 10.1016/j.trc.2017.08.022.

Mackay, Matthew, Paul Vicol, Jonathan Lorraine, David Duvenaud, and Roger Grosse (2019). "Self-Tuning Networks: Bilevel optimization of hyperparameters using structured best-response functions". In: *International Conference on Learning Representations (ICLR)*. New Orleans, Louisiana, USA. URL: https://openreview.net/forum?id=r1eEG20qKQ.

Maglaras, Costis, John Yao, and Assaf Zeevi (2017). "Optimal Price and Delay Differentiation in Large-Scale Queueing Systems". In: *Management Science* 64(5), pp. 2427–2444. DOI: 10.1287/mnsc.2016.2713.

Maldonado, Sebastián, Juan Pérez, Richard Weber, and Martine Labbé (2014). "Feature selection for support vector machines via mixed integer linear programming". In: *Information Sciences* 279, pp. 163–175. DOI: 10.1016/j.ins.2014.03.110.

Mandi, Jayanta, Emir Demirović, Peter J. Stuckey, and Tias Guns (2020). "Smart predict-and-optimize for hard combinatorial optimization problems". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34(2), pp. 1603–1610. DOI: 10.1609/aaai.v34i02.5521.

Mandl, Christian and Stefan Minner (2023). "Data-driven optimization for commodity procurement under price uncertainty". In: *Manufacturing & Service Operations Management* 25(2), pp. 371–390. DOI: 10.1287/msom.2020.0890.

Marinakis, Yannis, Georgia-Roumbini Iordanidou, and Magdalene Marinaki (2013). "Particle swarm optimization for the vehicle routing problem with stochastic demands". In: *Applied Soft Computing* 13(4), pp. 1693–1704. DOI: 10.1016/j.asoc.2013.01.007.

Mendoza, Jorge E., Louis-Martin Rousseau, and Juan G. Villegas (2016). "A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints". In: *Journal of Heuristics* 22, pp. 539–566. DOI: 10.1007/s10732-015-9281-6.

Miranda, Douglas Moura and Samuel Vieira Conceição (2016). "The vehicle routing problem with hard time windows and stochastic travel and service time". In: *Expert Systems with Applications* 64, pp. 104–116. DOI: 10.1016/j.eswa.2016.07.022.

Mišić, Velibor V. and Georgia Perakis (2020). "Data analytics in operations management: A review". In: *Manufacturing & Service Operations Management* 22(1), pp. 158–169. DOI: 10.1287/msom.2019.0805.

Miyashiro, Ryuhei and Yuichi Takano (2015). "Mixed integer second-order cone programming formulations for variable selection in linear regression". In: *European Journal of Operational Research* 247(3), pp. 721–731. DOI: 10.1016/j.ejor.2015.06.081.

Molina, L.C., L. Belanche, and A. Nebot (2002). "Feature selection algorithms: A survey and experimental evaluation". In: *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. Maebashi City, Japan, pp. 306–313. DOI: 10.1109/ICDM.2002.1183917.

Moon, Ilkyeong and Guillermo Gallego (1994). "Distribution free procedures for some inventory models". In: *Journal of the Operational Research Society* 45(6), pp. 651–658. DOI: 10.1057/jors.1994.103.

Moreira-Matias, Luis, João Gama, Michel Ferreira, João Mendes-Moreira, and Luis Damas (2013a). "On predicting the taxi-passenger demand: A real-time approach". In: *Progress in Artificial Intelligence: 16th Portuguese Conference on Artificial Intelligence, EPIA 2013, Angra do Heroísmo, Azores, Por-*

tugal, September 9-12, 2013. Proceedings 16. Ed. by Luís Correia, Luís Paulo Reis, and José Cascalho. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 54–65. ISBN: 978-3-642-40669-0. DOI: 10.1007/978-3-642-40669-0_6.

Moreira-Matias, Luis, João Gama, Michel Ferreira, João Mendes-Moreira, and Luis Damas (2013b). "Predicting taxi–passenger demand using streaming data". In: *IEEE Transactions on Intelligent Transportation Systems* 14(3), pp. 1393–1402. DOI: 10.1109/TITS.2013.2262376.

Narayanan, Santhanakrishnan, Emmanouil Chaniotakis, and Constantinos Antoniou (2020). "Shared autonomous vehicle services: A comprehensive review". In: *Transportation Research Part C: Emerging Technologies* 111, pp. 255–293. DOI: 10.1016/j.trc.2019.12.008.

Nazari, Mohammadreza, Afshin Oroojlooy, Lawrence Snyder, and Martin Takáč (2018). "Reinforcement Learning for Solving the Vehicle Routing Problem". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/9fb4651c05b2ed70fba5afe0b039a550-Paper.pdf.

Novoa, Clara and Robert Storer (2009). "An approximate dynamic programming approach for the vehicle routing problem with stochastic demands". In: *European Journal of Operational Research* 196(2), pp. 509–515. DOI: 10.1016/j.ejor.2008.03.023.

Nowozin, Sebastian and Christoph H. Lampert (2011). "Structured learning and prediction in computer vision". In: *Foundations and Trends® in Computer Graphics and Vision* 6(3–4), pp. 185–365. DOI: 10.1561/0600000033.

NYC Taxi & Limousine Commission (2010). *New York City Taxi & Limousine Commission - TLC Trip Record Data*. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page. [Online; accessed 10-December-2023].

Oroojlooyjadid, Afshin, Lawrence V. Snyder, and Martin Takáč (2020). "Applying deep learning to the newsvendor problem". In: *IISE Transactions* 52(4), pp. 444–463. DOI: 10.1080/24725854.2019.1632502.

Osorio, Carolina (2019). "Dynamic origin-destination matrix calibration for large-scale network simulators". In: *Transportation Research Part C: Emerging Technologies* 98, pp. 186–206. DOI: 10.1016/j.trc.2018.09.023.

Oyola, Jorge, Halvard Arntzen, and David L. Woodruff (2017). "The stochastic vehicle routing problem, a literature review, Part II: solution methods". In: *EURO Journal on Transportation and Logistics* 6(4), pp. 349–388. ISSN: 2192-4376. DOI: 10.1007/s13676-016-0099-7.

Oyola, Jorge, Halvard Arntzen, and David L. Woodruff (2018). "The stochastic vehicle routing problem, a literature review, part I: models". In: *EURO Journal on Transportation and Logistics* 7(3), pp. 193–221. DOI: 10.1007/s13676-016-0100-5.

Özkan, Erhun and Amy R. Ward (2020). "Dynamic matching for real-time ride sharing". In: *Stochastic Systems* 10(1), pp. 29–70. DOI: 10.1287/stsy.2019.0037.

Park, Young Woong and Diego Klabjan (2020). "Subset selection for multiple linear regression via optimization". In: *Journal of Global Optimization* 77(3), pp. 543–574. DOI: 10.1007/s10898-020-00876-1.

Parmentier, Axel (2021). "Learning structured approximations of combinatorial optimization problems". In: *arXiv preprint arXiv:2107.04323*. DOI: 10.48550/arXiv.2107.04323.

Parmentier, Axel and Vincent T'kindt (2023). "Structured learning based heuristics to solve the single machine scheduling problem with release times and sum of completion times". In: *European Journal of Operational Research* 305(3), pp. 1032–1041. DOI: 10.1016/j.ejor.2022.06.040.

Pavone, Marco (2015). *Autonomous Mobility-on-Demand Systems for Future Urban Mobility*. Ed. by Markus Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 399–416. ISBN: 978-3-662-45854-9. DOI: 10.1007/978-3-662-45854-9_19.

Pavone, Marco, Amin Saberi, Maximilian Schiffer, and Matt Wu Tsao (2022). "Technical Note–Online hypergraph matching with delays". In: *Operations Research* 70(4), pp. 2194–2212. DOI: 10.1287/opre.2022.2277.

Pavone, Marco, Stephen L. Smith, Emilio Frazzoli, and Daniela Rus (2012). "Robotic load balancing for mobility-on-demand systems". In: *The International Journal of Robotics Research* 31(7), pp. 839–854. DOI: 10.1177/0278364912444766.

Perakis, Georgia and Guillaume Roels (2008). "Regret in the newsvendor model with partial information". In: *Operations Research* 56(1), pp. 188–203. DOI: 10.1287/opre.1070.0486.

Pessoa, Artur, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck (2020). "A generic exact solver for vehicle routing and related problems". In: *Mathematical Programming* 183(1), pp. 483–523. DOI: 10.1007/s10107-020-01523-z.

Petruzzi, Nicholas C. and Maqbool Dada (1999). "Pricing and the newsvendor problem: A review with extensions". In: *Operations Research* 47(2), pp. 183–194. DOI: 10.1287/opre.47.2.183.

Pillac, Victor, Michel Gendreau, Christelle Guéret, and Andrés L. Medaglia (2013). "A review of dynamic vehicle routing problems". In: *European Journal of Operational Research* 225(1), pp. 1–11. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2012.08.015.

Pimenta, Victor, Alain Quilliot, Hélène Toussaint, and Daniele Vigo (2017). "Models and algorithms for reliability-oriented dial-a-ride with autonomous electric vehicles". In: *European Journal of Operational Research* 257(2), pp. 601–613. DOI: 10.1016/j.ejor.2016.07.037.

Psaraftis, Harilaos N., Min Wen, and Christos A. Kontovas (2016). "Dynamic vehicle routing problems: Three decades and counting". In: *Networks* 67(1), pp. 3–31. DOI: 10.1002/net.21628.

Qin, Yan, Ruoxuan Wang, Asoo J. Vakharia, Yuwen Chen, and Michelle M.H. Seref (2011). "The newsvendor problem: Review and directions for future research". In: *European Journal of Operational Research* 213(2), pp. 361–374. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2010.11.024.

Qin, Zhiwei (Tony), Hongtu Zhu, and Jieping Ye (2021). "Reinforcement learning for ridesharing: A survey". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 2447–2454. DOI: 10.1109/ITSC48978.2021.9564924.

Qin, Zhiwei (Tony), Hongtu Zhu, and Jieping Ye (2022). "Reinforcement learning for ridesharing: An extended survey". In: *Transportation Research Part C: Emerging Technologies* 144, p. 103852. ISSN: 0968-090X. DOI: 10.1016/j.trc.2022.103852.

Raza, Syed Mohib, Mohammad Sajid, and Jagendra Singh (2022). "Vehicle routing problem using reinforcement learning: Recent advancements". In: *Advanced Machine Intelligence and Signal Processing*. Ed. by Deepak Gupta, Koj Sambyo, Mukesh Prasad, and Sonali Agarwal. Springer Nature Singapore: Singapore, pp. 269–280. ISBN: 978-981-19-0840-8. DOI: 10.1007/978-981-19-0840-8_20.

Rei, Walter, Michel Gendreau, and Patrick Soriano (2010). "A hybrid Monte Carlo local branching algorithm for the single vehicle routing problem with stochastic demands". In: *Transportation Science* 44(1), pp. 136–146. DOI: 10.1287/trsc.1090.0295.

Rencher, Alvin C. and William F. Christensen (July 2012). *Methods of Multivariate Analysis*. John Wiley & Sons, Inc. DOI: 10.1002/9781118391686. URL: https://doi.org/10.1002%2F9781118391686.

Rios, Brenner Humberto Ojeda, Eduardo C. Xavier, Flávio K. Miyazawa, Pedro Amorim, Eduardo Curcio, and Maria João Santos (2021). "Recent dynamic vehicle routing problems: A survey". In: *Computers & Industrial Engineering* 160, p. 107604. DOI: 10.1016/j.cie.2021.107604.

Rios, Ignacio, Roger J.B. Wets, and David L. Woodruff (2015). "Multi-period forecasting and scenario generation with limited data". In: *Computational Management Science* 12(2), pp. 267–295. DOI: 10.1007/s10287-015-0230-5.

Ritzinger, Ulrike, Jakob Puchinger, and Richard F. Hartl (2016). "A survey on dynamic and stochastic vehicle routing problems". In: *International Journal of Production Research* 54(1), pp. 215–231. DOI: 10.1080/00207543.2015.1043403.

Rossi, Federico, Rick Zhang, Yousef Hindy, and Marco Pavone (2018). "Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms". In: *Autonomous Robots* 42, pp. 1427–1442. DOI: 10.1007/s10514-018-9750-5.

Rostami, Borzou, Guy Desaulniers, Fausto Errico, and Andrea Lodi (Mar. 2021). "Branch-Price-and-Cut Algorithms for the Vehicle Routing Problem with Stochastic and Correlated Travel Times". In: *Operations Research* 69(2), pp. 436–455. DOI: 10.1287/opre.2020.2037.

Sachs, Anna-Lena and Stefan Minner (2014). "The data-driven newsvendor with censored demand observations". In: *International Journal of Production Economics* 149, pp. 28–36. ISSN: 0925-5273. DOI: 10.1016/j.ijpe.2013.04.039.

Sadana, Utsav, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal (2024). "A survey of contextual optimization methods for decision-making under uncertainty". In: *European Journal of Operational Research*. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2024.03.020.

Sadeghi Eshkevari, Soheil, Xiaocheng Tang, Zhiwei Qin, Jinhan Mei, Cheng Zhang, Qianying Meng, and Jia Xu (2022). "Reinforcement Learning in the Wild: Scalable RL Dispatching Algorithm Deployed in Ridehailing Marketplace". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '22. Association for Computing Machinery: Washington DC, USA, 3838—3848. ISBN: 9781450393850. DOI: 10.1145/3534678.3539095.

Scarf, Herbert (1958). "A min-max solution of an inventory problem". In: *Studies in the Mathematical Theory of Inventory and Production*. Ed. by K.J. Arrow, S. Karlin, and H. Scarf. Stanford University Press, Stanford, pp. 201–209.

Schiffer, Maximilian, Gerhard Hiermann, Fabian Rüdel, and Grit Walther (2021). "A polynomial-time algorithm for user-based relocation in free-floating car sharing systems". In: *Transportation Research Part B: Methodological* 143, pp. 65–85. DOI: 10.1016/j.trb.2020.11.001.

Secomandi, Nicola and Francois Margot (2009). "Reoptimization approaches for the vehicle-routing problem with stochastic demands". In: *Operations Research* 57(1), pp. 214–230. DOI: 10.1287/opre.1080.0520.

See, Chuen-Teck and Melvyn Sim (2010). "Robust approximation to multiperiod inventory management". In: *Operations Research* 58(3), pp. 583–594. DOI: 10.1287/opre.1090.0746.

Serrano, Breno, Stefan Minner, Maximilian Schiffer, and Thibaut Vidal (2024). "Bilevel optimization for feature selection in the data-driven newsvendor problem". In: *European Journal of Operational Research* 315(2), pp. 703–714. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2024.01.025.

Shapiro, Alexander (2003). "Monte Carlo Sampling Methods". In: *Stochastic Programming*. Ed. by A. Ruszczynski and A. Shapiro. Vol. 10. Handbooks in Operations Research and Management Science. Elsevier, Amsterdam, pp. 353–425. DOI: 10.1016/S0927-0507(03)10006-0.

Shapiro, Alexander, Darinka Dentcheva, and Andrzej Ruszczyński (2014). *Lectures on Stochastic Programming: Modeling and Theory*. SIAM. DOI: 10.1137/1.9781611976595.

Shlezinger, Nir, Yonina C. Eldar, and Stephen P. Boyd (2022). "Model-based deep learning: On the intersection of deep learning and optimization". In: *IEEE Access* 10, pp. 115384–115398. DOI: 10.1109/ACCESS.2022.3218802.

Skordilis, Erotokritos, Yi Hou, Charles Tripp, Matthew Moniot, Peter Graf, and David Biagioni (2022). "A modular and transferable reinforcement learning framework for the fleet rebalancing problem". In: *IEEE Transactions on Intelligent Transportation Systems* 23(8), pp. 11903–11916. DOI: 10.1109/TITS.2021.3108733.

Soeffker, Ninja, Marlin W. Ulmer, and Dirk C. Mattfeld (2022). "Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review". In: *European Journal of Operational Research* 298(3), pp. 801–820. DOI: 10.1016/j.ejor.2021.07.014.

Solomon, Marius M. (Apr. 1987). "Algorithms for the vehicle routing and scheduling problems with time window constraints". In: *Operations Research* 35(2), pp. 254–265. DOI: 10.1287/opre.35.2.254.

Soza-Parra, Jaime, Rafał Kucharski, and Oded Cats (2024). "The shareability potential of ride-pooling under alternative spatial demand patterns". In: *Transportmetrica A: Transport Science* 20(2), p. 2140022. DOI: 10.1080/23249935.2022.2140022.

Spieser, Kevin, Kyle Treleaven, Rick Zhang, Emilio Frazzoli, Daniel Morton, and Marco Pavone (2014). "Toward a Systematic Approach to the Design and Evaluation of Automated Mobility-on-Demand Systems: A Case Study in Singapore". In: *Road Vehicle Automation*. Ed. by Gereon Meyer and Sven Beiker. Springer International Publishing: Cham, pp. 229–245. ISBN: 978-3-319-05990-7. DOI: 10.1007/978-3-319-05990-7_20.

Takano, Yuichi and Ryuhei Miyashiro (2020). "Best subset selection via cross-validation criterion". In: *TOP* 28(2), pp. 475–488. DOI: 10.1007/s11750-020-00538-1.

Tan, Kay Chen, Chun Yew Cheong, and Chi Keong Goh (2007). "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation". In: *European Journal of Operational Research* 177(2), pp. 813–839. DOI: 10.1016/j.ejor.2005.12.029.

Tang, Xiaocheng, Zhiwei (Tony) Qin, Fan Zhang, Zhaodong Wang, Zhe Xu, Yintai Ma, Hongtu Zhu, and Jieping Ye (2019). "A Deep Value-network Based Approach for Multi-Driver Order Dispatching". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Association for Computing Machinery: Anchorage, AK, USA, 1780–1790. ISBN: 9781450362016. DOI: 10.1145/3292500.3330724. URL: https://doi.org/10.1145/3292500.3330724.

Taş, D., Michel Gendreau, Nico Dellaert, Tom Van Woensel, and A.G. De Kok (Aug. 2014a). "Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach". In: *European Journal of Operational Research* 236(3), pp. 789–799. DOI: 10.1016/j.ejor.2013.05.024.

Taş, Duygu, Nico Dellaert, Tom van Woensel, and Ton De Kok (2014b). "The time-dependent vehicle routing problem with soft time windows and stochastic travel times". In: *Transportation Research Part C: Emerging Technologies* 48, pp. 66–83. DOI: 10.1016/j.trc.2014.08.007.

Tian, Yu-Xin and Chuan Zhang (2023). "An end-to-end deep learning model for solving data-driven newsvendor problem with accessibility to textual review data". In: *International Journal of Production Economics* 265, p. 109016. DOI: 10.1016/j.ijpe.2023.109016.

Toth, Paolo and Daniele Vigo (2014). *Vehicle routing: problems, methods, and applications, Second Edition*. Ed. by Daniele Vigo and Paolo Toth. SIAM: Philadelphia, PA. DOI: 10.1137/1.9781611973594.

Tsao, Matthew, Ramon Iglesias, and Marco Pavone (2018). "Stochastic Model Predictive Control for Autonomous Mobility on Demand". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 3941–3948. DOI: 10.1109/ITSC.2018.8569459.

Uber (2023). *Uber blog: Introducing Uber Reserve*. https://www.uber.com/en-GB/blog/uber-reserve-for-drivers/. [Online; accessed 21-May-2024].

Ulmer, Marlin W., Ninja Soeffker, and Dirk C. Mattfeld (2018). "Value function approximation for dynamic multi-period vehicle routing". In: *European Journal of Operational Research* 269(3), pp. 883–899. DOI: 10.1016/j.ejor.2018.02.038.

Ulmer, Marlin Wolf (2017). *Approximate dynamic programming for dynamic vehicle routing*. Vol. 61. Operations Research/Computer Science Interfaces Series. Springer Cham. DOI: 10.1007/978-3-319-55511-9.

Van Parys, Bart P.G. and M. Amine Bennouna (2022). "Robust Two-Stage Optimization with Covariate Data". In: *Available on Optimization Online*. URL: https://optimization-online.org/?p=20614.

Vidal, Thibaut, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins (2013). "Heuristics for multi-attribute vehicle routing problems: A survey and synthesis". In: *European Journal of Operational Research* 231(1), pp. 1–21. DOI: 10.1016/j.ejor.2013.02.053.

Vidal, Thibaut, Gilbert Laporte, and Piotr Matl (2020). "A concise guide to existing and emerging vehicle routing problem variants". In: *European Journal of Operational Research* 286(2), pp. 401–416. DOI: 10.1016/j.ejor.2019.10.010.

Wang, Charles X. and Scott Webster (2009). "The loss-averse newsvendor problem". In: *Omega* 37(1), pp. 93–105. DOI: 10.1016/j.omega.2006.08.003.

Wang, Hai and Hai Yang (2019). "Ridesourcing systems: A framework and review". In: *Transportation Research Part B: Methodological* 129, pp. 122–155. DOI: 10.1016/j.trb.2019.07.009.

Wang, Zizhuo, Peter W. Glynn, and Yinyu Ye (2016). "Likelihood robust optimization for data-driven problems". In: *Computational Management Science* 13(2), pp. 241–261. DOI: 10.1007/s10287-015-0240-3.

Xu, Zhe, Zhixin Li, Qingwen Guan, Dingshui Zhang, Qiang Li, Junxiao Nan, Chunyang Liu, Wei Bian, and Jieping Ye (2018). "Large-Scale Order Dispatch in On-Demand Ride-Hailing Platforms: A Learning and Planning Approach". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. Association for Computing Machinery: London, United Kingdom, pp. 905–913. ISBN: 9781450355520. DOI: 10.1145/3219819.3219824.

Yan, Chiwei, Helin Zhu, Nikita Korolko, and Dawn Woodard (2020). "Dynamic pricing and matching in ride-hailing platforms". In: *Naval Research Logistics* 67(8), pp. 705–724. DOI: 10.1002/nav.21872.

Yue, Jinfeng, Bintong Chen, and Min-Chiang Wang (2006). "Expected value of distribution information for the newsvendor problem". In: *Operations Research* 54(6), pp. 1128–1136. DOI: 10.1287/opre.1060.0318.

Zardini, Gioele, Nicolas Lanzetti, Marco Pavone, and Emilio Frazzoli (2022). "Analysis and control of autonomous mobility-on-demand systems". In: *Annual Review of Control, Robotics, and Autonomous Systems* 5(1), pp. 633–658. DOI: 10.1146/annurev-control-042920-012811.

Zhang, Junlong, William H.K. Lam, and Bi Yu Chen (2013). "A stochastic vehicle routing problem with travel time uncertainty: trade-off between cost and customer service". In: *Networks and Spatial Economics* 13, pp. 471–496. DOI: 10.1007/s11067-013-9190-x.

Zhang, Junlong, William H.K. Lam, and Bi Yu Chen (2016). "On-time delivery probabilistic models for the vehicle routing problem with stochastic demands and time windows". In: *European Journal of Operational Research* 249(1), pp. 144–154. DOI: 10.1016/j.ejor.2015.08.050.

Zhang, Lingyu, Tao Hu, Yue Min, Guobin Wu, Junying Zhang, Pengcheng Feng, Pinghua Gong, and Jieping Ye (2017). "A Taxi Order Dispatch Model based On Combinatorial Optimization". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Association for Computing Machinery: Halifax, NS, Canada, pp. 2151–2159. ISBN: 9781450348874. DOI: 10.1145/3097983.3098138.

Zhang, Rick and Marco Pavone (2016). "Control of robotic mobility-on-demand systems: a queueing-theoretical perspective". In: *The International Journal of Robotics Research* 35(1-3), pp. 186–203. DOI: 10.1177/0278364915581863.

Zhang, Rick, Federico Rossi, and Marco Pavone (2019). "Analysis, Control, and Evaluation of Mobility-on-Demand Systems: A Queueing-Theoretical Approach". In: *IEEE Transactions on Control of Network Systems* 6(1), pp. 115–126. DOI: 10.1109/TCNS.2018.2800403.

Zhang, Yanfei and Junbin Gao (2017). "Assessing the performance of deep learning algorithms for newsvendor problem". In: *Neural Information Processing*. Ed. by Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed M. El-Alfy. Springer International Publishing: Cham, pp. 912–921. DOI: 10.1007/978-3-319-70087-8_93.

Zhang, Yu, Roberto Baldacci, Melvyn Sim, and Jiafu Tang (2019). "Routing optimization with time windows under uncertainty". In: *Mathematical Programming* 175, pp. 263–305. DOI: 10.1007/s10107-018-1243-y.

Zhang, Yu, Zhenzhen Zhang, Andrew Lim, and Melvyn Sim (2021). "Robust Data-Driven Vehicle Routing with Time Windows". In: *Operations Research* 69(2), pp. 469–485. DOI: 10.1287/opre.2020.2043.

Zhao, Jiuxia, Minjia Mao, Xi Zhao, and Jianhua Zou (2021). "A hybrid of deep reinforcement learning and local search for the vehicle routing problems". In: *IEEE Transactions on Intelligent Transportation Systems* 22(11), pp. 7208–7218. DOI: 10.1109/TITS.2020.3003163.

Zhou, Ming, Jiarui Jin, Weinan Zhang, Zhiwei Qin, Yan Jiao, Chenxi Wang, Guobin Wu, Yong Yu, and Jieping Ye (2019). "Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. CIKM '19. Association for Computing Machinery: Beijing, China, 2645–2653. ISBN: 9781450369763. DOI: 10.1145/3357384.3357799. URL: https://doi.org/10.1145/3357384.3357799.

Zhu, Liping, Mian Huang, and Runze Li (2012). "Semiparametric quantile regression with high-dimensional covariates". In: *Statistica Sinica* 22(4), pp. 1379–1401. DOI: 10.5705/ss.2010.199.