Integrated Robust and Adaptive Methods in the Heating Oil Industry

by

Joel Tay

B.A., University of Cambridge (2011)

Submitted to the
Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2019

Signature redacted

Author ....
Sloan School of Management

Signature redacted

Sep 1, 2018

Certified by......... ....
Dimitris Bertsimas
Boeing Leaders for Global Operations
Sloan School of Management
Co-director, Operations Research Center
Thesis Supervisor

Signature redacted

Accepted by....... ....
Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-director, Operations Research Center

Integrated Robust and Adaptive Methods in the Heating Oil Industry

by

Joel Tay

Submitted to the Sloan School of Management
on Sep 1, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Almost six million households in the United States alone use heating oil as their main fuel, the vast majority of these in the Northeastern US. In this thesis, we examine some problems faced by a planner who is contracted to resupply customers with heating oil through the winter season, and use robust and adaptive optimization and machine learning to develop models that allow the planner to address these problems under uncertainty at a realistic scale.

In the first part of the thesis (Chapter 2), we consider the problem of resupplying customers spread over a geographical area. Due to the presence of uncertainty in demand, the planner has to choose an appropriate fleet size, decide on the most cost-effective routes and schedules, and on how much to resupply each customer. We develop novel scalable and adaptive algorithms to address this problem, demonstrating the potential for significant cost savings in simulations while being able to address problem sizes in the thousands.

In the second part of the thesis (Chapter 3), we consider the problem of executing the purchase of a commodity. In addition to price uncertainty on the daily commodity market, we model two kinds of discounts offered by commodity sellers vying for the planner's business. We develop a tractable model to formulate a purchasing strategy for a desired quantity, and use recently-developed machine learning techniques to find optimal decision trees that the planner can apply to different problem parameters to yield readily interpretable purchasing strategies, without having to re-solve the optimization models. We demonstrate experimentally that these strategies perform almost as well as those given by the actual optimization models.

Finally, in the third part of the thesis (Chapter 4), we demonstrate the possibility of solving the previous two problems as an integrated whole, allowing the planner to simultaneously optimize the routing, scheduling, and purchasing aspects of heating oil delivery. Although the integrated problem size may be too large to solve directly with realistic problem sizes, we use Lagrangean decomposition methods to make the problem tractable, and show experimentally that this allows us to get high-quality solutions that reduce the combined cost of the two subproblems.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Leaders for Global Operations
Sloan School of Management
Co-director, Operations Research Center

Acknowledgments

I would first like to acknowledge my deepest gratitude to Dimitris Bertsimas, my advisor, as someone who has taught me not only to do research, but also to aspire to change the world for the better. With unflagging optimism and energy, he has always encouraged me to aim higher, to work smarter, and to push harder in my research. Seeing through the years his passion for learning, for teaching, and for seeking to make a real impact through his work has motivated me not just to believe in myself as a researcher, but also to demand much more from myself. I deeply appreciate his generosity with his time and attention, above and beyond our research meetings.

I would also like to thank Patrick Jaillet and Carolina Osorio for serving on my thesis committee. Their guidance has been invaluable as my thesis took shape, and I am extremely grateful for their thoughtful feedback throughout my time at MIT. Both Patrick and Carolina were voices of encouragement in moments when I struggled, and for that too I am exceedingly thankful.

In understanding the research problems which I worked on in my thesis, I had the opportunity to ask questions and hear firsthand from experienced practitioners in the heating oil industry. I would like to thank Ted Lorden and Betsy Blanc for introducing this interesting domain to us, and for helping me ask the important questions.

Outside of research, one of the highlights of my time here was the chance to work with some of the best teachers, who have a passion for education and care deeply for their students. I am particularly grateful to Karen Zheng and Colin Fogarty for the time spent as their teaching assistant, from which I learned so much.

A special thank you also goes to the people who are responsible for the Operations Research Center, my home at MIT. While Dimitris and Patrick, the co-directors, set the tone of the Center, I would also like to thank Laura Rose and Andrew Carvalho, who helped me navigate the torrent of deadlines and paperwork which doth so easily beset us.

I value greatly the friendships that were forged here at the ORC. Swati, who worked with me on the class project which eventually became the paper on which

Chapter 2 of this thesis is based. Iain and Miles, who patiently helped me exterminate many bugs in my code. And of course the memories of hard work and hard play, for which I also thank Nataly and Jacopo, Yaron and Efrat, David and Rhea, John and Sarah, Leon, Paul, Peng, Wang Chi, He, Joline, Phebe, Zachary, Chaitanya, Vishal, Angie, Kris, Fernanda, Daniel and Shiyun, Velibor, Nikita, and my ORC neighbors, Michael, Rajan and Nishanth.

My journey to Operations Research was serendipitous and sesquipedelian. For even learning that the field existed, I am indebted to Melvyn Sim, and to my mentors in A*STAR, Tony Quek and Lee Eng Wah.

I am immensely grateful for the many friends we made at Christ the King Presbyterian, who were there when we needed them. We will miss you all very much.

Finally, I have always been greatly comforted to have the support of my family back home. Especially to my parents and in-laws, thank you for your love and prayers. To Amos, Soph and Wes, thanks for being there. And to my long-suffering and often-optimized wife Sarah, for moving to the other side of the world with me, and to our brave son Chesterton: my biggest and bestest love.

Contents

List of Figures

List of Tables

To Kong

Chapter 1

Introduction

Decision makers have access to more data now than ever before. From increasingly complex prediction models for everything from weather patterns to consumer behavior, to the vast arrays of cheap sensors driving the Internet of Things, there is a real need for models that scale well with the size of the real-world problems that they aim to solve.

At the same time, the increase in the power of computational hardware has added to the theoretical improvements in mixed-integer optimization solvers over the last few decades, giving planners a chance to solve models that were once far too large to even consider. This has allowed a paradigm shift in operations research from formulating models that have less dependence on a priori assumptions, towards data-driven models that combine the computational and data advantages of our time to create models with ever-greater predictive powers.

The framework of robust optimization, in particular, has proven to be a fruitful approach to the problem of modeling the uncertainty that is inherent in most of these domains. Contrasted to the traditional approach of stochastic optimization that considers uncertainty under some posited probability distribution, robust optimization considers a set-based deterministic model of the uncertainty, and determines the worst-case solution over all actualizations in that set.

Over many domains, models formulated by the robust optimization methodology have been able to yield high-quality solutions that offer an attractive tradeoff between a small increase in average cost and protection from the worst-case uncertainty. Practitioners have often been able to solve robust optimization models quickly with modern computational power, and these models scale well with the original problems. Robust optimization integrates well with linear optimization, mixed-integer optimization, and often does well with even more powerful models such as semidefinite optimization and convex optimization.

In this thesis, we apply the methodology of robust optimization to some of the central problems that might be faced by a planner in the heating oil industry.

Heating oil is a liquid petroleum distillate that is used as a fuel oil for furnaces or boilers in buildings. Known in the US as "No. 2 heating oil", it is most commonly used among residences in the Northeastern US. Almost 6 million households in the US rely on heating oil as their main space heating fuel [75], 84% of these in the Northeast. In 2016, for example, this corresponded to 3.1 billion gallons of heating oil sold to Northeast residential consumers [76].

A planner in the heating oil industry has to grapple with several forms of uncertainty, all of which will cut heavily into the bottom line if not properly addressed. Most important amongst these is the demand uncertainty, which varies between consumers not only at an individual level, but also in a highly correlated way through the seasonality of temperature, since the main purpose of heating oil is space heating. It is affected by trends in the weather throughout the heating season (October to March), and also by daily temperature variations.

In addition, the price of heating oil is volatile, exhibiting fluctuations that depend heavily on the price of crude oil in a complex way [100, 88]. Spatial price variations depending on local market structures and socio-economic factors have also been observed [147].

In this thesis, these two sources of uncertainty are the main drivers behind the planner's need to protect their operations against the worst-case actualizations. We next summarize the three chapters of this thesis in the context of protecting the purchasing and resupplying operations of a heating oil practitioner against uncertainty. Our contributions and a detailed literature review are introduced separately in each of the three chapters.

1.1 Scalable Robust and Adaptive Inventory Routing

In the first part of this thesis, we consider the problem of inventory routing where a planner is contracted to resupply customers with a commodity for which they have uncertain demand over time. We consider a model of temperature-dependent demand uncertainty that is particularly relevant for heating oil consumption over winter.

To maintain the customers' inventory, the planner has to determine routes and schedules to visit the customers, and the quantities that should be refueled. Making high-quality decisions is valuable because it allows the planner to maintain a smaller fleet of vehicles, an important factor in reducing operational cost. The planner is interested both in reducing the frequency of stockouts and minimizing the cost of operations.

Using robust optimization, a deterministic set-based approach to modeling uncertainty, we formulate the problem with demand uncertainty. Although the uncertainty set we obtain is non-convex, we present an algorithm that lets us maintain tractability by finding explicitly the worst-case actualization over the uncertainty set. We show that our model scales well with the problem size and yields high-quality solutions that reduce, in simulations over a variety of data sets, both the total operational cost and the stockouts experienced. We show that this corresponds also to a potential reduction in fleet size.

Finally, we demonstrate the use of a cutting-plane algorithm to generalize our results to adaptive optimization, leading to the possibility of real-time updates in the routing and scheduling decisions made by the planner. Our models are able to solve problems of sizes in the thousands, over a realistic time horizon of an entire winter season (150 days).

1.2 Robust Purchase Execution

In the second part of this thesis, we consider the problem of a planner who is required to purchase some quantity of a commodity. The planner has to decide not only when to execute the purchase over time, but also has a choice of sellers to choose between. Because of competition, these sellers might offer the commodity at a price somewhat lower than the daily spot price.

Using the robust optimization methodology, we model the fluctuations of the spot price of the commodity as an ellipsoidal uncertainty set, and also present a novel formulation of the sellers that incorporates discounts that they can offer to incentivize the planner to make larger and more frequent purchases.

We show that our model can be solved tractably for realistic problem sizes. We also show that the model remains tractable when extended to multiple purchases over the time period, and also for more general convex uncertainty sets. We also motivate the qualitative forms of the solutions for various parameters as being optima for simplified versions of the problem.

Finally, we use a modern machine learning approach to generate useful insights from our model, and address the need for online solutions. We show that the planner can generate optimal decision trees to investigate various qualitative features of the optimal solutions, without having to re-solve the model for a new set of parameters or data. We demonstrate in simulations that our decision trees can attain high accuracy in predicting the presence of these qualitative features, and also in predicting the cost of the model.

We finally demonstrate the use of a suite of decision trees to generate high-quality strategies for the robust purchase execution problem, and show that these strategies are competitive with those obtained from solving the optimization model directly.

1.3 Integration of Robust Purchase Execution and Robust Inventory Routing

In the third part of this thesis, we consider integrating the inventory routing problem in Chapter 2 and the commodity purchasing problem in Chapter 3. This is one example of a problem where the individual subproblems may be tractable for realistic problem sizes, but combining them causes the problem to become too large to solve exactly.

Using Lagrangean decomposition, a generalization of Lagrangean relaxation that creates copies of the variables that appear in both subproblems and relaxes the constraint enforcing their equality, we decompose the problem such that all the constraints in the original problem are preserved in one of the subproblems. We show experimentally that this allows us to find high-quality solutions to the integrated problem.

We are able to demonstrate in simulations that integrating the two previous problems leads to around 10% decrease in mean cost when the subproblems are small, and yields a smaller improvement for large subproblem sizes. We also observe a significant decrease in worst-case cost and cost variability over the solutions obtained from solving the subproblems separately.

Chapter 2

Scalable Robust and Adaptive Inventory Routing

2.1 Introduction

We consider the rich problem of inventory routing where a supplier has a contract with individual customers to monitor their inventory of a commodity that diminishes over time, and to resupply that commodity to maintain customer stocks above a certain threshold. Some sizeable industries concerned with inventory routing problems of this type are those supplying commodities such as soft drinks in vending machines, portable water in offices, or heating oil in residential areas. In many of these inventory routing applications, the presence of uncertainty in the customers' demand for the commodity (and other uncertainties in data, e.g. temperature in heating oil usage models) is a critical issue that must be addressed in order to provide solutions that are of practical value in the real world. In this chapter, we provide novel scalable and adaptive algorithms to address the inventory routing problem using a robust and adaptive optimization framework.

Given a network of customers spread over a geographic area, the supplier needs to make the following key operational decisions:

- **Fleet size**: ahead of the operational period, the supplier needs to decide the number of vehicles to be maintained and the crew size required. A larger crew size and more vehicles increase the cost of operation, whereas a reduction in these may impact the quality of service negatively, and require a larger emergency fleet to handle stockouts.

- **Routes and Schedules**: the supplier needs to determine which routes to utilize to visit customers, and when to schedule these routes, while minimizing their cost of operation (thus maximizing their profits).

- **Refuelling quantities**: when a customer is visited, the supplier needs to determine how much of the commodity to resupply. Attempting to resupply all customers to their full capacity might not be feasible for the vehicles' capacity, or it might limit the number of customers that a vehicle can resupply.

Having defined our key operational decisions, we now consider the key objectives that a supplier is concerned with, namely: (i) **reducing the frequency of stockouts** and (ii) **minimizing the cost of operations**. Reducing the frequency of stockouts is important, as, besides the obvious damage to brand image that results from customers' stocks being depleted, it is also highly undesirable for suppliers because they have to designate vehicles to make unplanned emergency replenishments of these customers, often at very short notice.

Regarding operational cost, much of the inventory routing literature (e.g. [104]) has focused on minimizing the routing cost while maintaining a desired level of service. However, an important reason that many approaches to this problem do not scale well is that they attempt to solve for the optimal routes. As this requires solving the Travelling Salesman Problem as a subproblem, it becomes difficult to use these approaches to solve problems of the sizes required in real-world applications.

In practice, vehicles in the fleet will be traveling or servicing customers for similar lengths of time, and so it is thus natural to seek solutions with reduced fleet sizes that are robust to uncertainties in the rate of customers' demand for the commodity, while minimizing the routing costs only as a second-order objective. This allows the planner both to reduce expensive stockout resupplies, and the capital, maintenance and labor cost of the vehicle fleet, which is usually particularly high in the peak season and has a greater cost savings potential than fuel cost (e.g. [99]). It is therefore sufficient for our purposes to use a fast heuristic for the routing component, which ensures feasibility for the routing for a given vehicle fleet size.

Current exact approaches in the literature [138, 3] solve only up to around a hundred customers and do not scale to problem sizes that arise in real life, while heuristic solutions usually decompose the problem into a series of problems with shorter time horizons because of concerns about tractability and uncertain data (e.g. [71, 131]. Our main application throughout the chapter is to companies that provide heating oil in residential areas. For example, a typical company of this nature in New England might have a customer base spanning north central Massachusetts and southern New Hampshire with around 10,000 customers. Our key contribution is a robust and adaptive mixed integer optimization (MIO) formulation that scales to large problem sizes, augmented with a demand uncertainty set that varies with temperature and heuristic route generation. Using data sets generated from real temperature data, we demonstrate both the effectiveness and scalability of our approach.

The rate of demand of the commodity has typically been considered in the literature (e.g. [56], or the survey of [87]) to be either (dynamically) deterministic or stochastic. A deterministic rate of demand, as with many optimization problems, leads to more tractable but less realistic models. A stochastic rate of demand, however, is less tractable for large instances and often leads to heuristic solutions which are sensitive to the assumptions made about the probability distribution of the demand. In contrast, a robust optimization approach combines the tractability of deterministic models with the realism of stochastic approaches by modeling uncertainty in a deterministic manner, and leads to solutions that are less sensitive to the probabilistic assumptions made about the underlying demand.

Our contributions in this section can be summarized as follows:

1) **Robustness.** We present a robust formulation of the uncertainty set for demand that captures, for the case of resupplying heating oil, the dependence of demand on temperature as well as individual customers' rates of consumption. This results in a

novel non-convex uncertainty set which we are able to tractably optimize over, thus generating the critical worst-case demand scenarios.

2) **Adaptability.** For the case where customer demand can be recorded remotely, we present an approach that allows us to adapt our operational decisions according to observed demand. We demonstrate computationally that the adaptive solutions outperform both the deterministic and robust formulations.

3) **Scalability.** By combining:

a) novel ways to generate the critical worst-case demand scenarios,

b) automated neighborhood route selection,

c) route generation heuristics,

d) generating constraints on the fly for the adaptive formulation,

we are able to solve problems with ~6000 customers over a time horizon of 150 days, within two hours for both the robust and adaptive formulations.

4) **Quality of solutions.**

a) We demonstrate that the robust solutions of our model materially decrease stockouts and are relatively insensitive to estimation noise in demand and temperature, achieving across a variety of data sets of sizes ranging from 51 to 5915, an average reduction in stockouts of over 94% from a deterministic model.

b) We show that both the robust and adaptive formulations can be used to reduce vehicle fleet size, while still outperforming the deterministic solution.

c) We demonstrate that the robust and adaptive solutions lead to a decrease in total operational cost for the supplier, when combining routing cost with vehicle fleet cost and the cost of resupplying customers who experience stockouts.

The remainder of this chapter is structured as follows: in Section 2.2, we survey some of the related literature and discuss why current approaches do not scale well. In Section 2.3, we introduce deterministic, robust and adaptive models for the capacitated inventory routing problem. We define our uncertainty set, and provide an algorithm that maximizes affine functions of demand over the uncertainty set. In Section 2.4, we discuss our techniques for route generation and some heuristics to further improve our routes. We detail our experiments and computational results in

Section 2.5. We finally conclude with overall discussion and some future directions in Section 2.6.

2.2 Related Work

Vehicle routing problems (VRPs) arise naturally from many problem contexts, and as such have been extensively studied in many flavors. Beginning with "The Truck Dispatching Problem", proposed by [65], the difficulty of these problems and their relevance to many industries have generated much research over the past few decades.

One of the best-studied formulations of vehicle routing problems is the capacitated vehicle routing problem (CVRP), which in its most basic form describes the problem of determining a minimum-cost set of routes by which a fleet of delivery vehicles with limited capacity delivers quantities of a product or commodity to customers at various locations. When the costs of potential routes and the customer demands are assumed to be fixed and known, this is a deterministic problem. Early approaches for getting exact solutions of the CVRP were for decades dominated by branch-and-bound algorithms (e.g. [57, 58, 112]); in addition, branch-and-cut algorithms were later developed with many different families of cuts [113, 14, 134, 118, 16], often building on research on the Travelling Salesman Problem. More recently, another popular approach is to solve the problem using column generation alongside cut generation (e.g. [85, 15, 129]). We refer the reader to [61, 90, 111, 18, 143] for detailed literature surveys about the CVRP and related vehicle routing problems.

However, the solutions to deterministic VRPs can be sensitive to errors or uncertainties in the parameters of the problem, becoming suboptimal or even infeasible for real-world actualizations. This has typically been addressed by taking the uncertain parameters as random variables, and utilizing stochastic programming to formulate the model. Assuming a known probability distribution for the uncertain parameters, probabilistic guarantees can then be made (e.g. a chance-constrained VRP). (More generally, the field of stochastic programming is described in much greater detail in [50] and [137], just to give two examples.) However, stochastic VRPs are much harder to solve than their deterministic counterparts [73]. Developing exact algorithms that solve these problems to optimality has been challenging for problems of any realistic size, and much work has been done on heuristics (see [142] for a detailed survey), and recently, metaheuristics [143, 9] that work well on VRPs.

Often, in addition to finding suitable routes, the planner has to manage levels of inventory between a number of customers or retailers, i.e. solve an inventory routing problem (IRP) [81]. An important version of the IRP that we address is that of Vendor-Managed Inventory (VMI). A business practice that was popularized in the 1980s by Walmart and Procter & Gamble, in VMI models the suppliers are responsible for monitoring the inventory levels of their customers, and deciding on replenishment schedules and quantities accordingly. This can result in benefits such as lower inventory required [148], cost reductions [135], and a smaller bullwhip effect in supply chains [68].

One key difference here is that in the problem we address, the planner does not have real-time telemetry measurements of the customer's inventory, which induces uncertainty in the modeling of the demand, not just in the future consumption of the customers, but also in the time period that has elapsed since the last replenishment. We note that in VMI, it is usually assumed that the planner has access to these real-time measurements, which can make the model more responsive to changing conditions, at the cost of some tractability.

Due to the increased difficulty of simultaneously solving for schedules and quantities, the problem is in practice simplified in a variety of ways. For example, the quantities could be decided by a deterministic order-up-to-level policy [28, 7] where the customer is always replenished to maximum capacity. Alternatively, sample-based methods can be used to extend methods for deterministic demand to work with stochastic demand [103].

An approach commonly used in practice is to model the resupplying problem as a series of one-day problems, where forecasting models based on the historical consumption of the customers and temperature data indicate which customers are likely to need replenishment within the next day. The planner then optimizes a capacitated routing problem to determine routes that will cover these customers, along with customers who will need replenishment in the following days. [131] introduces a tabu search metaheuristic and two large neighborhood search metaheuristics for this problem, while [72] and [71] developed a two-stage approach for propane delivery where the customers are first assigned to specific days, and then routes are constructed daily.

A different approach to modeling IRPs with stochastic demands has been to handle the demands' dependence on uncertain temperatures by using Markov decision process models, (e.g. [73, 109, 2], among many more). These approaches tend to have higher computational times, making them less useful for more complex problems of realistic sizes. For more comprehensive reviews of the general inventory routing problem (IRP) and various solution approaches, we refer the reader to [80, 54, 109, 61, 29, 59], and a recent comparison of different IRP formulations [8].

A problem with a similar flavor is that of IRPs with transportation procurement, where the planner outsources the deliveries to the customers. In contexts where this is possible, it can lead to more flexibility as the planner does not have a fixed fleet size constraint, and requires optimization of the purchase of transport capacity in each time period, rather than routing each vehicle. We direct interested readers to the recent works of [26, 27] for these.

A paradigm that has proven useful in approaching problems modeling optimization under uncertainty is Robust Optimization (RO) (for instance [47, 21, 67, 31]). This approach leads to solutions that are guaranteed to satisfy the constraints for all uncertain parameters in a chosen uncertainty set, and often leads to tractable models requiring weaker assumptions on the uncertain parameters than stochastic formulations. RO formulations have been found in practice to yield solutions that are competitive with the optimal deterministic solution, and perform significantly better in worst-case scenarios. They also tend to be less affected by errors in parameter estimation or structural assumptions [91, 48].

While demand uncertainty has long been considered in its stochastic form [30, 49, 87], recent works have proven the usefulness of RO in formulating certain varieties of VRPs [126]. For instance, [140] consider a formulation of the single-stage

Robust Capacitated VRP (RCVRP) under demand uncertainty that can be solved deterministically, using the budget-of-uncertainty approach first developed in [47], and [93] consider the RCVRP with more general demand uncertainty that can be reformulated to yield numerical solutions.

[138] and [3] have previously addressed the inventory routing problem within a RO framework. [138] report solving instances with a branch-and-cut algorithm, solving a Travelling Salesman problem as a subproblem exactly, for up to 30 customers and a time horizon of seven periods. [3] uses a heuristic approach to generate routes, proposing a nonlinear MIO problem, and reports solving for cyclic distribution routes for 50 customers. In contrast, our methods allow us to solve problems with the number of customers two orders of magnitude larger than both of these, over a time horizon which is an order of magnitude larger than [138], by solving a deterministic MIO to generate robust solutions, and using a cutting-plane algorithm to generate adaptive solutions.

Finally, we consider the problem of formulating an adaptive multistage robust optimization model. While the fully adaptive robust optimization problem is intractable via a dynamic programming approach, affinely-adaptive robust optimization solutions have been found to perform almost as well, while retaining the tractability of single-stage robust optimization problems [22, 40]. This approach has recently been applied to the unit commitment problem in power generation [44, 117]. Finite adaptability is a different approach that works well for some multistage robust optimization models [32, 37], but we chose affine adaptability due to its stronger scalability characteristics.

2.3 Problem Formulation

2.3.1 Context

To view the problem we address in a concrete context, consider the following inventory routing problem over a finite horizon: A company has customers who consume a homogeneous commodity over time, and a fleet of vehicles that is used to resupply them. We would like to generate a feasible schedule of routes for the vehicles that satisfies capacity constraints for users and vehicles, and leads to a low likelihood of stockouts for the customers.

A key insight that helps us achieve this is the observation that in practice, customers are often located in small neighborhoods, and that most of the variable cost (i.e., travelling distance) of the routing problem is derived from travel between depots and these small neighborhoods of customers. Within these neighborhoods, then, routes can be optimized sufficiently for industrial purposes by local search algorithms such as 2-opt [64]. Therefore, our approach is to think of routes not as a list of customers, but as a neighborhood which a vehicle might travel between in a given time period. Upon selecting a route for a vehicle, a feasible schedule is then one which assigns customers to that vehicle that are on that route, i.e., in the associated selected neighborhood. Correspondingly, we assign costs to routes based on travel between the depot and the customers in the selected neighborhood, bearing in mind that the costs are to be taken as accurate only to the first order. The realized cost will depend on the customers we assign to the vehicle servicing a route.

This has a few key advantages. Firstly, it leverages the current knowledge of the company in the form of extant routes and neighborhoods, driver experience and other geographic and network information. In other words, it allows us to warm start our model with a set of routes that are already known to be feasible, and gradually introduce routes to improve the solution quality of our model as needed. Furthermore, it significantly reduces the solution space of feasible routes, which helps the model to scale to large problem sizes more easily. By varying the sizes and coverage of the set of routes that we optimize over, we can exercise control over the tradeoff between scalability and solution quality, as needed.

Naturally, this observation does not hold true for all problem domains. Where the routing cost is dominated by the cost of individual links of the route, e.g. in a problem where a driver has to visit all customers on a small route, then more sophisticated vehicle routing algorithms will be necessary. Often, though, heuristic algorithms are sufficient for route optimization at a local level, and indeed planners in many industries will be best served to use commercially available routing software within small neighborhoods.

For the vehicle routing problem under consideration, our decision-making has to take into account two sources of uncertainty in the demand for the commodity. The more important of these is the uncertainty associated with changes in temperature, which is correlated across all the customers. To a smaller extent, there is also an uncertainty in demand specific to each customer, which we assume is uncorrelated across customers. Using the well-established RO methodology, we define appropriate uncertainty sets (see (2.12) below) that capture these phenomena. In Section 2.3.4, we discuss ways to initialize the parameters of this uncertainty set from observations or simulations of the uncertain data.

2.3.2 Nominal Formulation

We begin by defining the nominal formulation of the inventory routing problem - in other words, we solve the problem for the case where demand is fixed rather than uncertain. Consider N customers who need to be resupplied over a time horizon T, who we index as customers $i \in [N] = \{1, \dots, N\}$. The customers are to be resupplied with a fleet of M vehicles, each of capacity S. In a single time period, the vehicles can be assigned to a tour $\theta \in [\Theta]$, each which has associated cost c_θ. Each customer i has a maximum capacity of Q_i, and we suppose that customer i begins the season with Z_i of the commodity remaining. For the nominal formulation, we assume that demand d_i^t is known for all customers and time periods.

We consider the following decision variables:

- $g_{i,\theta}^t$, the amount of fuel that customer i will be resupplied via route θ at time t,
- u_i^t, the total amount of fuel that customer i will be supplied at time t,
- Binary variable v_θ^t which is 1 if and only if tour θ is selected at time t.

Then, the nominal formulation is:

$$
\begin{aligned}
\min_{\mathbf{u},\mathbf{v},\mathbf{g}} \quad & \sum_{t=1}^{T}\sum_{\theta=1}^{\Theta} c_\theta v_\theta^t && (2.1)\\
\text{s.t.} \quad & 0 \le Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t} d_i^\tau, \quad \forall i \in [N], \quad \forall t \in [T], && (2.2)\\
& Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t-1} d_i^\tau \le Q_i, \quad \forall i \in [N], \quad \forall t \in [T], && (2.3)\\
& \sum_{\theta=1}^{\Theta} v_\theta^t \le M, \quad \forall t \in [T], && (2.4)\\
& u_i^t \le \sum_{\theta=1}^{\Theta} g_{i,\theta}^t, \quad \forall i \in [N], \quad \forall t \in [T], && (2.5)\\
& \sum_{i=1}^{N} g_{i,\theta}^t \le S v_\theta^t, \quad \forall \theta \in [\Theta], \quad \forall t \in [T], && (2.6)\\
& g_{i,\theta}^t = 0, \quad \forall i \in [N], \quad \forall \theta : i \notin \theta, \quad \forall t \in [T], && (2.7)\\
& g_{i,\theta}^t \ge 0, \quad \forall i \in [N], \quad \forall \theta \in [\Theta], \quad \forall t \in [T], &&\\
& u_i^t \ge 0, \quad \forall i \in [N], \quad \forall t \in [T], &&\\
& v_\theta^t \in \{0,1\}, \quad \forall \theta \in [\Theta], \quad \forall t \in [T]. &&
\end{aligned}
$$

Eq. (2.1) expresses the cost minimization objective. Eq. (2.2) guarantees that each customer is resupplied so that their supply of the commodity is never depleted, while Eq. (2.3) enforces customer capacity constraints. Eq. (2.4) respects the fleet size. Eq. (2.5) ensures that the amount of fuel assigned to refuel a customer is also assigned to some route in the same time period. Eq. (2.6) both allows us to assign fuel to a route only if the route is actually selected, and if so, also enforces vehicle capacity limits. Eq. (2.7) ensures that assignments are only made for customers that are on a given route.

2.3.3 Robust Formulation

Now we move to the robust formulation of the inventory routing problem. Here, rather than assume we know what the demand $\mathbf{d}$ is, we assume rather that it lies within an uncertainty set $\mathcal{U}$ which we have constructed beforehand. We discuss the construction of $\mathcal{U}$ in more detail in the next subsection. We also assume that the amounts of fuel that customers start with, Z_i, take values in the interval $[\underline{Z}_i, \bar{Z}_i]$.

As before, we consider the same variables $g^t_{i,\theta}$, u^t_i and v^t_θ. Then the robust formulation is the same as before, except that now constraints (2.2) and (2.3) become:

$$0 \leq Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t} d_i^\tau, \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall \boldsymbol{d} \in \mathcal{U}, \ \ \forall Z_i \in [\underline{Z}_i, \bar{Z}_i], \tag{2.8}$$

$$Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t-1} d_i^\tau \leq Q_i, \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall \boldsymbol{d} \in \mathcal{U}, \ \ \forall Z_i \in [\underline{Z}_i, \bar{Z}_i], \tag{2.9}$$

with the same interpretations.

2.3.4 Constructing $\mathcal{U}$

We describe here one method of constructing $\mathcal{U}$ based on insights from the Central Limit Theorem [19], particularly applicable to the scenario of supplying heating oil to residences during winter. To do this, we assume that for any given customer, expected demand is constant above a certain temperature and increases linearly as the temperature decreases below that point. Specifically, for customer i, we assume that there exists a breakpoint Ψ_i above which expected demand is B_i^0, and that if the temperature decreases below Ψ_i, the expected demand increases with a slope (against temperature) of B_i^1. We operate with the supposition that Ψ_i, B_i^0 and B_i^1 have been estimated for each customer from historical data.

We now assume that for each time period t, the temperature τ_t is subject to i.i.d. variation, and thus construct a CLT-style uncertainty set $\mathcal{U}_\tau$ for the temperature,

$$\mathcal{U}_{\tau} = \left\{ \boldsymbol{\tau} : \left| \frac{\sum_{t=1}^{T} (\tau_t - \bar{\tau}_t)}{\sigma_{\tau}\sqrt{T}} \right| \leq \Gamma_{\tau}, \;\; \bar{\tau}_t - 3\sigma_{\tau} \leq \tau_t \leq \bar{\tau}_t + 3\sigma_{\tau} \;\; \forall t \in [T] \right\}. \tag{2.10}$$

Here $\bar{\tau}_t$ and σ_{τ} are the mean and standard deviation of the temperatures respectively, and Γ_{τ} is a robust parameter that we are free to select, which we discuss below. We refer to the value $\sqrt{T}\sigma_{\tau}\Gamma_{\tau}$ as the budget of variation in temperature, i.e., the net amount our temperatures are allowed to vary from their means.

We next consider the additional noise in the demand. For simplicity, we assume the demand is subject to additional zero-mean noise that has the same distribution for each time period, but is i.i.d. across customers, and thus construct a CLT-style uncertainty set $\mathcal{U}_{\epsilon}$ for the noise in demand,

$$\mathcal{U}_{\epsilon} = \left\{ \boldsymbol{\epsilon} : \left| \frac{\sum_{i=1}^{N} \epsilon_i}{\sigma_{\epsilon}\sqrt{N}} \right| \leq \Gamma_{\epsilon}, \;\; -3\sigma_{\epsilon} \leq \epsilon_i \leq 3\sigma_{\epsilon} \;\; \forall i \in [N] \right\}, \tag{2.11}$$

where σ_{ϵ} is the standard deviation of the demand noise.

This gives us our uncertainty set for demand, $\mathcal{U}$, which, as described above, consists of all demand vectors for which the corresponding temperature and demand noise simultaneously lie within the uncertainty sets $\mathcal{U}_{\tau}$ and $\mathcal{U}_{\epsilon}$, respectively.

$$\mathcal{U} = \left\{ \boldsymbol{d} : d_i^t = B_i^0 + B_i^1 \max(0, \Psi_i - \tau_t) + \epsilon_i, \;\; \boldsymbol{\tau} \in \mathcal{U}_{\tau}, \;\; \boldsymbol{\epsilon} \in \mathcal{U}_{\epsilon} \right\}, \tag{2.12}$$

where B_i^0, B_i^1 and and Ψ_i are all parameters estimated from data.

Selecting robust parameters The uncertainty sets $\mathcal{U}_{\tau}$ and $\mathcal{U}_{\epsilon}$ involve the parameters Γ_{τ} and Γ_{ϵ} that represent the planner's desired balance between optimality and robustness. We next outline our approach for selecting these parameters. Assuming that temperatures τ_t are independent for each time period t, with mean $\bar{\tau}_t$ and variance σ_τ^2 from an otherwise unknown distribution, we select Γ_τ such that $\mathcal{U}_\tau$ contains the realized temperature with probability 99% for large T. Specifically, from the Central Limit Theorem,

$$\lim_{T \leftarrow \infty} \mathbb{P}\left(\left|\sum_{t=1}^{T}(\tau_t - \bar{\tau}_t)\right| \leq \Phi^{-1}(0.99)\,\sigma_\tau\sqrt{T}\right) = 0.99, \tag{2.13}$$

where Φ is the cdf of the standard normal distribution, and so we select $\Gamma_\tau = \Phi^{-1}(0.99)$. A similar approach is used for selecting Γ_ϵ. For other possible approaches to selecting the robust parameters, see [23, 48, 21, 38].

For a given planning horizon T and N customers, let the demands $\boldsymbol{d} \in \mathbb{R}^{N\times T}$ lie in the uncertainty set $\mathcal{U}$ given in (2.12), which is non-convex, necessitating a novel approach to generate critical worst-case scenarios.

Note that the only robust constraint in our formulation is constraint (2.8), which requires us to protect against the maximum and minimum values of $\sum_{\tau=1}^{t} d_i^\tau$ over $\mathcal{U}$ for each customer i in $[N]$ and each day t in $[T]$. We next give algorithm Opt-Temp that allows us to optimize over $\mathcal{U}$ an affine combination of convex non-increasing functions of temperature. Note that demand without customer-specific noise is a convex non-increasing function of temperature in our model. In addition, as each robust constraint only involves one customer, the worst-case ϵ_i can always be taken to be $3\sigma_\epsilon$ for maxima, and $-3\sigma_\epsilon$ for minima. Given a customer i and day t, we can use these to construct a demand vector $d_i \in \mathbb{R}^T$ that maximizes the sum $\sum_{\tau=1}^{t} d_i^\tau$. This enables us to solve the robust formulation as a deterministic problem, vastly improving computational performance. For notational convenience, we refer to the natural projection of $\mathcal{U}$ onto the set of demand vectors for customer i as $\mathcal{U}_{[i]}$.

Summary of algorithm: To maximize the sum $\sum_{t=1}^{T} a_t d_t(\tau_t)$ for $\tau \in \mathcal{U}_\tau$ as defined by Eq. 2.10, where $d_t(\tau_t)$, for each t, is a convex non-increasing function of τ, we let the set of days with non-negative affine coefficients, i.e., $a_t \geq 0$, be T_1, and those with negative affine coefficients, i.e., $a_t < 0$, be T_2. In algorithm Opt-Temp, we consider two cases: (i) $|T_1| \geq |T_2|$ and (ii) $|T_1| < |T_2|$. For the first case, we set all temperatures to be at their upper bounds, i.e., $\tau_t = \bar{\tau}_t + 3\sigma_\tau$. We then greedily choose the days in T_1 and for each such t decrease its corresponding temperature as far as possible. In the second case, we set all the temperatures to be at their lower bounds, i.e., $\tau_t = \bar{\tau}_t - 3\sigma_\tau$. We then optimize the restricted objective function over the days T_2 using standard convex optimization techniques. In both cases, we ensure that the temperatures selected respect the bound $|\sum_{t=1}^{T} \tau_t - \bar{\tau}_t| \leq \Gamma_\tau \sqrt{T} \sigma_\tau$, where Γ_τ is a robust parameter. To prove optimality, we show that there exists an optimal solution with at most one temperature not attaining one of its bounds, and that our algorithm finds such a solution.

Formally, we present in Algorithm 1 an algorithm OPT-TEMP for maximizing an affine combination of convex non-increasing functions over $\mathcal{U}_\tau$. The algorithm finds, for convex non-increasing functions $d_t(\tau)$ and coefficients a_t, a temperature vector yielding $\max_{\tau \in U_\tau} \sum_{t=1}^{T} a_t d(\tau_t)$. In our presentation of the algorithm we use a sorting function SORT(R), which sorts the set of days R in descending order of the difference in the objective function when the temperature is changed from $\bar{\tau}_t + 3\sigma_\tau$ to $\bar{\tau}_t - 3\sigma_\tau$, i.e., SORT$(R) = \{t_1, t_2, \ldots, t_{|R|}\}$ such that $\Delta(t_x) \geq \Delta(t_y)$ whenever $x < y$, where:

$$\Delta(t) = a_t(d_t(\bar{\tau}_t - 3\sigma_\tau) - d_t(\bar{\tau}_t + 3\sigma_\tau)).$$

$D(q, k, F)$ calculates the increase in objective value that we could get, for fixed k and F, of allowing q to be the single time period that does not achieve either of its temperature bounds. Figure 2-1 explains the logic of the algorithm graphically.

Theorem 1. *The temperature vector $\boldsymbol{\tau}^* \in \mathbb{R}^T$ output by the Algorithm 1 maximizes $\sum_{t=1}^{T} a_t d_t(\tau_t)$ over $\mathcal{U}_\tau$.*

Proof. We first show that $\boldsymbol{\tau}^*$ is feasible. For the case where $|T_1| < |T_2|$, the temperatures are guaranteed to be feasible by definition of the optimization subproblem that we solve (Note that as we only optimize for T_2, this is a convex optimization problem and so tractable). To show feasibility for the case where $|T_1| \geq |T_2|$, we

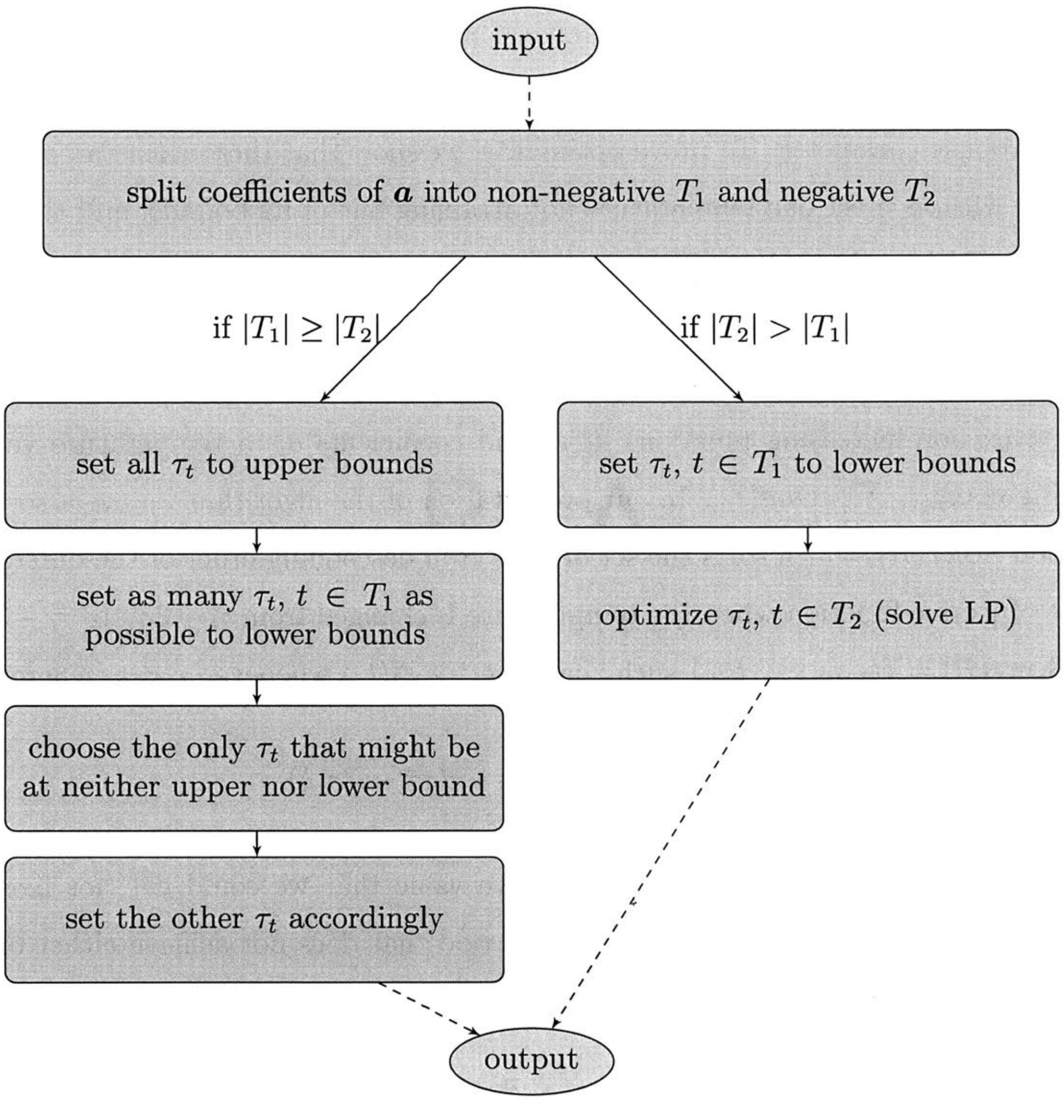

Figure 2-1: The logic of Algorithm 1.

Algorithm 1: OPT-TEMP

Input: $\Gamma > 0$, σ, $\bar{\boldsymbol{\tau}} \in \mathbb{R}^T$, $\boldsymbol{a} \in \mathbb{R}^T, d_t : \mathbb{R} \to \mathbb{R}\ \forall t \in [T]$

Output: $\tau \in \arg\max_{\tau \in \mathcal{U}_\tau} \sum_{t=1}^T a_t d_t(\tau_t)$

$T_1 = \{t \in T : a_t \geq 0\},\ T_2 = T \setminus T_1, k = l = m = 1;$

if $|T_1| \geq |T_2|$ **then**

  $\tau_t = \bar{\tau}_t + 3\sigma\ \forall t \in [T],\ F = 3T\sigma;$

  $\{t_1, t_2, \ldots, t_{|T_1|}\} = \text{SORT}(T_1);$

  while $F \geq 6\sigma - \Gamma\sqrt{T}\sigma$ **and** $k < |T_1|$ **do**

    $(\tau_{t_k}, F) \leftarrow (\tau_{t_k} - 6\sigma, F - 6\sigma);$

    $k \leftarrow k + 1;$

  end

  if $F > -\Gamma\sqrt{T}\sigma$ **and** $k < |T_1|$ **then**

    $q^* = \arg\max_{q \in T_1} D(q, k, F);$

    if$(q^* \leq k - 1)$ **then**

     $(\tau_{t_k}, \tau_{t_{q^*}}, F) \leftarrow (\tau_{t_k} - 6\sigma, \tau_{t_{q^*}} + 6\sigma - F - \Gamma\sqrt{T}\sigma, -\Gamma\sqrt{T}\sigma);$

    else $(\tau_{t_{q^*}}, F) \leftarrow (\tau_{t_{q^*}} - F - \Gamma\sqrt{T}\sigma, -\Gamma\sqrt{T}\sigma);$

  end

end

else $\tau = \arg\max_{\tau \in \mathcal{U}'} \sum_{t=1}^T a_t d_t(\tau)$ for $\mathcal{U}' = \mathcal{U}_\tau \cap \{\tau : \tau_t = \bar{\tau}_t - 3\sigma_\tau\ \forall t \in T_1\};$

where SORT is the sorting function defined earlier, and

$$\text{where } D(q,k,F) = \begin{cases} a_q d_q(\bar{\tau}_q + 3\sigma - F) - a_q d_q(\bar{\tau}_k + 3\sigma) & \text{if } q > k, \\ a_q d_q(\bar{\tau}_q + 3\sigma - F) + a_{k+1} d_{k+1}(\bar{\tau}_{k+1} - 3\sigma) & \\ -a_q d_q(\bar{\tau}_q - 3\sigma) - a_{k+1} d_{k+1}(\bar{\tau}_{k+1} + 3\sigma) & \text{if } q \leq k. \end{cases}$$

consider the bookkeeping variable F, which tracks the value of $\sum_{t=1}^{T}(\tau_t - \bar{\tau}_t)$. Before we update a temperature, we check that F will not exceed the CLT-type bounds $-\Gamma\sqrt{T}\sigma \leq F \leq \Gamma\sqrt{T}\sigma$, and limit the magnitude of our updates accordingly. Similarly, the temperatures are initialized at their upper bounds and never decreased by more than 6σ, the width of the feasible interval for a single temperature. Also, note that we are assured of the existence of a feasible solution (e.g. setting the temperatures to their mean values). Thus, $\boldsymbol{\tau}^*$ is feasible.

Next we prove that $\boldsymbol{\tau}^*$ is optimal. Suppose we had a feasible temperature vector where for some $r \in T_1$, $\tau_r > \bar{\tau}_r - 3\sigma$, and for some $s \in T_2$, $\tau_s < \bar{\tau}_s + 3\sigma$. Then we could decrease τ_r and increase τ_s by some small ϵ, while not decreasing the objective function. This means that we can limit ourself to optimal solutions where either the temperatures in T_1 all attain their lower bounds $\bar{\tau}_r - 3\sigma_\tau$, or the temperatures in T_2 all attain their upper bounds $\bar{\tau}_s + 3\sigma_\tau$. (We will show that the smaller set attains its bounds.)

Case (i) $|T_1| < |T_2|$: We show that in this case, there exists at least one optimal temperature vector τ^* such that $\tau_t^* = \bar{\tau}_t - 3\sigma$ for all days in T_1. (Note that such an optimal temperature vector is easy to find: for days in T_1, all the temperatures are at their lower bounds, and temperatures for days in T_2 can be found using linear optimization). Consider any optimal temperature vector τ^{opt} that maximizes $\sum_{t=1}^{T} a_t d_t(\tau_t)$ such that all the temperatures in T_2 attain their upper bounds, i.e., $\tau_t^{opt} = \bar{\tau}_t + 3\sigma$ for $t \in T_2$ (if not, as argued above, all the temperatures in T_1 must be at their lower bounds, thus proving our claim). Let $F^{opt} = \sum_{t=1}^{T}(\tau_t^{opt} - \bar{\tau}_t) = \sum_{t\in T_1}(\tau_t^{opt} - \bar{\tau}_t) + 3\sigma|T_2|$. Note that $F^{opt} \leq \Gamma\sqrt{T}\sigma$ since τ^{opt} is feasible. Now, consider a temperature vector τ' such that $\tau_t' = \bar{\tau}_t - 3\sigma$ for $t \in T_1$ and $\tau_t' = \bar{\tau}_t + 3\sigma$ for $t \in T_2$. Let $F' = \sum_{t\in T_1}(\tau_t' - \bar{\tau}_t) + \sum_{t\in T_2}(\tau_t' - \bar{\tau}_t) = 3(|T_2| - |T_1|) \geq 0$. Also, note that $F' \leq F^{opt} \leq \Gamma\sqrt{T}\sigma$. Thus, τ' is feasible and its function value is no worse than τ^{opt}. Hence, we have proved that there exists an optimal temperature vector which attains the lower bounds for temperatures in T_1.

In this case, optimality follows from the definition of the optimization subproblem that we solve, restricted to T_2.

Case (ii) $|T_1| \geq |T_2|$: Similar to the previous case, we can assume that the temperatures in T_2 all attain their upper bounds, i.e., for all $t \in T_2$, we have $\tau_t = \bar{\tau}_t + 3\sigma$. We next show that there exists such an optimal solution where at most one temperature τ_t for a $t \in T_1$ is neither at $\bar{\tau}_t - 3\sigma$ nor $\bar{\tau}_t + 3\sigma$.

Suppose we had some feasible solution with $r, s \in T_1$, $\tau_r \neq \bar{\tau}_r \pm 3\sigma$, $\tau_s \neq \bar{\tau}_s \pm 3\sigma$. We want to adjust these temperatures so that one attains its bound, without decreasing the objective function. Let $a = \min(\tau_r - (\bar{\tau}_r - 3\sigma), \bar{\tau}_s + 3\sigma - \tau_s)$, $b = \min(\bar{\tau}_r + 3\sigma - \tau_r, \tau_s - (\bar{\tau}_s - 3\sigma))$. By the convexity of d_r and d_s, we use Jensen's inequality to get:

$$\frac{b}{a+b} d_r(\tau_r - a) + \frac{a}{a+b} d_r(\tau_r + b) \geq d_r(\tau_r), \tag{2.14}$$

$$\frac{a}{a+b} d_s(\tau_s - b) + \frac{b}{a+b} d_s(\tau_s + a) \geq d_s(\tau_s). \tag{2.15}$$

Adding these inequalities implies that either $d_r(\tau_r - a) + d_s(\tau_s + a)$ or $d_r(\tau_r + b) + d_s(\tau_s - b)$ must be at least $d_r(\tau_r) + d_s(\tau_s)$, and so we can adjust τ_r and τ_s as desired. We thus can limit ourselves to considering temperature vectors with at most one temperature not attaining either of its 3-sigma bounds.

Finally, suppose we knew that τ_t was the temperature not attaining its bounds. Then, a simple greedy algorithm for the temperature values at lower and upper bounds would give the optimal temperature vector.

In our algorithm, we iterate over all the choices for the day with the temperature not attaining its bounds, and select the one with the best objective value. The remaining temperatures are set to their upper or lower bounds, sorted so that they have the same output a greedy algorithm would have. Therefore, we obtain a temperature vector that maximizes the objective function over both sets of days, T_1 and T_2.

□

We can now explicitly find the minima and maxima over $\mathcal{U}$ for the sums of demand seen in the robust constraints. For the maximum demand, we construct a worst-case temperature vector for $\sum_{\tau=1}^{t} d_i^t$ using the above algorithm. As mentioned above, the robust constraints each involve just a single customer and so ϵ_i can be taken to be $3\sigma_\epsilon$.

For the minimum demand, the algorithm requires us to solve a convex optimization subproblem. In fact, for each $s \in [T]$ and $i \in [N]$, we can compute the minimum value of $\sum_{t=1}^{s} d_i^t$ by solving the following linear optimization:

$$\min \sum_{t=1}^{s} d_i^t \tag{2.16}$$

$$\text{s.t.} \quad -\Gamma_\tau \sqrt{T}\sigma_\tau \leq \sum_{t=1}^{T} (\bar{\tau}_t - \tau_t) \leq \Gamma_\tau \sqrt{T}\sigma_\tau, \tag{2.17}$$

$$\bar{\tau}_t - 3\sigma_\tau \leq \tau_t \leq \bar{\tau}_t + 3\sigma_\tau, \quad \forall t \in [T], \tag{2.18}$$

$$d_i^t \geq B_i^0 + B_i^1 x_t - 3\sigma_\epsilon, \quad \forall i \in [N], \quad \forall t \in [T], \tag{2.19}$$

$$x_t \geq \Psi_i - \tau_t, \quad \forall t \in [T], \tag{2.20}$$

$$\boldsymbol{x} \geq \boldsymbol{0}, \quad \boldsymbol{d}_i \geq \boldsymbol{0}. \tag{2.21}$$

Similar to before, ϵ_i can be taken to be $-3\sigma_\epsilon$. This allows us to replace our robust constraints with $2NT$ deterministic constraints, in each case picking the appropriate endpoint of the interval $[\underline{Z}_i, \bar{Z}_i]$ to robustify against (i.e. $\underline{Z}_i$ for lower bounds and $\bar{Z}_i$ for upper bounds).

In practice, we observed that as the robust constraints for time t do not involve customer demands for time periods beyond that, it improved the performance of our algorithm to project $\mathcal{U}$ onto the first t time periods and find the worst-case vector corresponding to $\Gamma_\tau\sqrt{t/T}$. This weakens the theoretical probabilistic guarantees that we can make, because the Central Limit Theorem might not apply in small cases. However, in our experiments this adaptation did not result in a significant increase in stockouts, but it did produce a significant decrease in the cost (and conservativeness) of the models. Note that the protection against stockouts is weakest against the earlier time periods at the very start of the heating season, when a customer is less likely to stockout anyway.

2.3.5 Affine Adaptive Robust Formulation

As technology develops, it is becoming increasingly feasible for companies to install sensors in customers' buildings. This might allow them, for instance, to track the daily consumption of their customers, improving the solution quality of their planning models. While it may be impractical to alter the fleet and crew schedule on short notice, we adapt our formulation so that the quantity of fuel resupplied will now be partially responsive to the actual demand observed. Without this new information from sensors, a company is limited to observations made during scheduled deliveries, i.e., the aggregated demand between refuelling decisions, which is much less informative.

We now define an affine adaptive robust formulation that applies to the scenario where we have additional real-time information about customers' demands. Instead of having the model decide on exact amounts to refuel each customer daily, we set the quantities refuelled to be affine functions of the demand in the previous days, and solve for the coefficients of these affine functions.

To make the formulation adaptive, we substitute each u_i^t with an affine function of previous days' demands: $u_i^t = b_i^{0,t} + \sum_{j=1}^{t-1} b_i^{j,t} d_i^j$ (remember that consumption for a day occurs after any refuelling on that day), where the various $b_i^{j,t}$ are now variables we are solving for. Similarly, we substitute each $g_{i,\theta}^t$ with $g_{i,\theta}^t = a_{i,\theta}^{0,t} + \sum_{j=1}^{t-1} a_{i,\theta}^{j,t} d_i^j$, where $a_{i,\theta}^{j,t}$ are variables.

This leads to the following formulation:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{v},\mathbf{g}} \quad \sum_{t=1}^{T}\sum_{\theta=1}^{\Theta} c_\theta v_\theta^t \tag{2.22}$$

$$\text{s.t.} \quad 0 \le Z_i + \sum_{\tau=1}^{t}(b_i^{0,\tau} + \sum_{j=1}^{\tau-1} b_i^{j,\tau} d_i^j) - \sum_{\tau=1}^{t} d_i^\tau, \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall \mathbf{d} \in \mathcal{U}, \tag{2.23}$$

$$Z_i + \sum_{\tau=1}^{t}(b_i^{0,\tau} + \sum_{j=1}^{\tau-1} b_i^{j,\tau} d_i^j) - \sum_{\tau=1}^{t-1} d_i^\tau \le Q_i, \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall \mathbf{d} \in \mathcal{U}, \tag{2.24}$$

$$\sum_{\theta=1}^{\Theta} v_\theta^t \le M, \ \ \forall t \in [T], \tag{2.25}$$

$$b_i^{0,t} + \sum_{j=1}^{t-1} b_i^{j,t} d_i^j \le \sum_{\theta=1}^{\Theta}(a_{i,\theta}^{0,t} + \sum_{j=1}^{t-1} a_{i,\theta}^{j,t} d_i^j), \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall \mathbf{d} \in \mathcal{U}, \tag{2.26}$$

$$\sum_{i=1}^{N}(a_{i,\theta}^{0,t} + \sum_{j=1}^{t-1} a_{i,\theta}^{j,t} d_i^j) \le S v_\theta^t, \ \ \forall \theta \in [\Theta], \ \ \forall t \in [T], \tag{2.27}$$

$$a_{i,\theta}^{j,t} = 0, \ \ \forall i \in [N], \ \ \forall \theta : i \notin \theta, \ \ \forall t \in [T], \ \ \forall j \in \{0, \ldots, t-1\}, \tag{2.28}$$

$$a_{i,\theta}^{j,t} \ge 0, \ \ \forall i \in [N], \ \ \forall \theta \in [\Theta], \ \ \forall t \in [T], \ \ \forall j \in \{0, \ldots, t-1\},$$

$$b_i^{j,t} \ge 0, \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall j \in \{0, \ldots, t-1\},$$

$$v_\theta^t \in \{0,1\}, \ \ \forall \theta \in [\Theta], \ \ \forall t \in [T].$$

Note that all the constraints in the adaptive robust formulation have the same interpretation as their counterparts in the robust formulation, although fuel supplied is now adaptive in that it is an affine function of demand. Furthermore, the starting quantities, Z_i, are no longer taken to be uncertain, as we would expect real-time measurements of demand to also yield exact information about the customers' remaining fuel.

The number of variables in the adaptive formulation is an order of magnitude greater than the nominal or robust case. Thus it is impractical to solve it using a deterministic linear MIP, as we did for the nominal formulation. In addition, the constraints (2.23), (2.26) and (2.27) involve products of our decision variables and the uncertain demand. This means that to separate over these constraints one would need to solve a quadratic optimization problem over a non-convex set.

We instead use a cutting-plane algorithm that exploits the structure of the uncertainty set $\mathcal{U}$, to tractably solve the adaptive formulation. Given a candidate solution, we can, for each of the constraints (2.23) or (2.26), use OPT-TEMP to give us the worst-case demand corresponding to that particular constraint and candidate solution, i.e., if the constraint is violated, we can find a demand vector in $\mathcal{U}$ that shows the violation, giving us a feasible cutting plane. Specifically, as noise for each customer is constant across time periods, the noise ϵ_i for a worst-case demand vector for that constraint-candidate pair is given by a greedy algorithm sorting on its coefficient, $\sum_{i=1}^{N} a_{i,\theta}^{0,t} \sum_{j=1}^{t-1} a_{i,\theta}^{j,t}$. If $\mathbf{d}^*$ is such a demand vector that causes a violation, we can add new deterministic constraints that check the violated constraints against $\mathbf{d}^*$. We then reoptimize the model, each time enforcing a check against all the previously-violated constraints with their associated demand vectors, and generate a new candidate solution. We repeat the process until the candidate solution we have does not violate any of the constraints.

2.4 Route Generation

Route generation is a widely studied problem, especially given its importance in various vehicle and inventory routing problems (for example, see [110, 84, 90]), applied to a plethora of real-world applications such as routing for bakery companies [127], blood product distribution [102], grocery industry [136], ship-routing [96]. The literature is ripe with a number of exact [17, 110, 112] and heuristic [101, 145, 116, 136] approaches for route generation. Since tractability is a major concern with exact approaches, we employ heuristic methods to generate a feasible set of routes. We would however like to emphasize that exploring route generation techniques is not the main focus of this work.

In this work, we consider routes to be not just feasible tours, but potential neighborhoods of customers, where a schedule will specify which subset of customers is actually served on a given trip. Our formulation operates under the approximation that there is a fixed cost to 'visit' a neighborhood, irrespective of how many houses are actually resupplied with the commodity. This is a reasonable approximation in our problem context since the cost of routing is a secord order cost, compared to the cost of customer stockouts (affecting customer satisfaction and reputation of the firm) and the cost of maintaining a fleet of vehicles.

We generate an initial set of feasible neighborhoods for our datasets in two ways: (i.) using a user-operated GUI where the supplier can manually select neighborhoods that are typically served together, (ii.) using an automated sweep of the customer locations. The first approach is preferred when the supplier would like to utilize prior knowledge and accumulated expertise about different neighborhoods. Our second approach is an automated sweep of the geographic area under consideration. Our algorithm creates a *cover* of the entire space with 'neighborhoods' or boxes so that the number of customers in each box lies in an interval. This interval is selected so that a vehicle is able to resupply about half the customers in the neighborhood to maximum capacity. Experimentally, these sweeps generate a good first set of feasible routes that make the problem scalable.

We further improve the quality of the routes using a set-cover formulation inspired by the work of [53] augmented with well-studied tabu-search heuristics for improving routes (for e.g. in [60, 86]). We consider a set of possible schedules for the customers, covered using feasible routes such that on any day at most M vehicles are used. However, we deviate from [53] by assuming that the cost, c_θ, of a route θ is given by the Euclidean distance of the tour suggested by the 2-opt (TSP) heuristic from the depot that a customer is served from (which is usually good enough in practice). We construct the following input from a pre-computed solution of the nominal problem.

- T, the total number of days in the planning horizon,
- M, the maximum number of vehicles in the fleet,
- N, the number of customers,

- S_i, the set of valid schedules that a customer i could be visited at. In order to construct this set, we consider the service schedule suggested by the nominal solution, and shift it by allowing each customer to be visited up to three days before or after the scheduled delivery.

- Ω, the set of feasible routes. We initialize Ω with the set of routes obtained by either neighborhood selection or automatic sweep. We will now describe how we use the following formulation to improve the quality of the set of routes.

We next select routes, using the following set-covering-like formulation. Let a_θ^i be a constant equal to 1 if customer i is on route θ, and 0, otherwise, for all $i \in [N], \theta \in \Omega$. Let b_p^t be equal to 1 if service schedule p is feasible for customer i, i.e., $p \in S_i$ and b_p^t is 0, otherwise. We use two sets of binary variables: x_θ^t and y_p^i. Here x_θ^t is 1 if and only if route θ is selected on day t and 0, otherwise. Finally, y_p^i is 1 if and only if schedule p is selected for customer i and 0, otherwise. We now formulate an binary linear program as follows:

$$\min \sum_{\theta \in \Omega} \sum_{t \in [T]} x_\theta^t c_\theta \tag{2.29}$$

$$\text{s.t.} \sum_{p \in S_i} y_p^i = 1, \quad i \in [N], \tag{2.30}$$

$$\sum_{\theta \in \Omega} x_\theta^t a_\theta^i - \sum_{p \in S_i} y_p^i b_p^t \geq 0, \quad i \in [N], \;\; t \in [T], \tag{2.31}$$

$$\sum_{\theta \in \Omega} x_\theta^t \leq M, \quad \forall t \in [T], \tag{2.32}$$

$$x_\theta^t \in \{0,1\}, \quad r \in \Omega, \;\; t \in [T], \tag{2.33}$$

$$y_p^i \in \{0,1\}, \quad p \in S_i, \;\; i \in [N]. \tag{2.34}$$

The objective function aims at minimizing the cost of the routes selected. Constraints (2.30) guarantee that exactly one feasible service schedule is selected for a customer. Constraints (2.31) guarantee that if the selected service schedule for customer i requires service on day t, then there must be a route selected on day t with customer i on the route. Constraints (2.32) ensure that at most M vehicles are used on any day, thereby respecting the fleet size.

We relax the above integer optimization problem and do column generation on the resulting optimization relaxation. We generate a set of candidate routes using the following heuristic operations on all the existing routes:

- **Insert** a customer into an existing neighbouring route,
- **Swap** two customers from neighboring routes,
- **Remove** customers from an existing route,
- **Construct** new routes for each day t by considering customers i such that their dual variables $p_{i,t}$ take large values.

For each candidate route θ for each day t, we compute its reduced cost as follows: $\bar{c}_\theta = c_\theta - \sum_{i\in\theta} p_{i,t} - p_M$ where $p_{i,t}$ is the optimal dual variable corresponding to constaints (2.31) and p_M is the dual variable corresponding to the constraint (2.32). We add the route that has the most negative reduced cost out of our set of candidate solutions, until a set with the required number of routes covering each customer is generated.

2.5 Computational Experiments

To test the scalability of our problem formulations and the quality of our solutions, we generated a number of datasets based on real-world problems. We imported customer locations from a few instances in the TSPLIB, the standard test bed of the Traveling Salesman Problem, with the size of these instances (i.e. the number of customers) ranging from 51 to 5915 (data instances `eil51, rat99, kroB200, rat575, pcb1173, d2103, rl5915`). For simplicity, we assumed a homogenous fleet of vehicles (in particular, with identical maximum capacity), operating from a single depot located at the centroid of the users.

We assumed all the customers to have homogenous heating oil tanks with identical maximum capacity. For each customer, we generated a base temperature above which their expected demand was near-zero and constant, and below which it increased linearly as temperature decreased. We randomly generated family sizes for each customer ranging from 2 to 4, and scaled the mean demand accordingly, adding noise in both the temperature and for each user's demand as described in Section 2.3.3. To tune our uncertainty parameters for temperatures, we used actual data for Boston for the months of November 2013 to March 2014, representing a full season of heating oil consumption [149].

We further generated estimated initial amounts for each customer, and subjected these to further zero-mean uncertainty proportional to the difference of these amounts to the full customer capacity, encapsulating the principle that a customer with higher usage or a customer who was serviced a longer time ago should have more uncertainty in their starting amounts. To be precise, if the estimated initial amount for a customer was z_i^{est}, the noisy initial amount was

$$z_i = z_i^{est} + (Q - z_i^{est}) \times U_i, \quad U_i \sim U(-1/2, 1/2).$$

(These are all decisions consonant with the size of actual companies (e.g. as described in [70] for propane delivery), and in practice companies already use estimates based on similar parameters.)

Family sizes and uncertain initial amounts were randomized separately to get *training* and *testing* datasets. We used the training set to tune our robust parameters Γ_τ and Γ_ϵ, and set these parameters correspondingly in the testing set to test the performance of our approach in terms of running time and effectiveness. We assumed in our experiments that a centralized depot serves all the customers, although the formulation generalizes easily to applications with multiple depots, each with their own fleet of vehicles. We give exact details of the parameters in Appendix A.

For our computational experiments, we let the nominal, robust and adaptive formulations solve for two hours each, using the nominal solution as a warm start (though infeasible) to the robust model, and the robust solution as a warm start to the adaptive model.

While the adaptive model we presented in Section 2.3.5 schedules a customer with refuelling quantities that are affine in all of that customer's observed demand, we improved the tractability of our implementation of the adaptive model by relaxing the number of terms of the adaptability. Specifically, we limited the refuelling quantity for a customer for time period t to be a base amount $b_i^{0,t}$, plus a term linear in that customer's demand during time period $t-1$ (i.e., d_i^{t-1}), a term linear in the total demand of the customer during time periods $t-3$ and $t-2$ (i.e., $d_i^{t-3} + d_i^{t-2}$), and a term linear in the total demand of the customer during time periods $t-7$ to $t-4$ (i.e., $d_i^{t-7} + \cdots + d_i^{t-4}$). We also solved for non-adaptive quantities across different vehicles, i.e. for $g_{i,\theta}^t$. We used the robust solution as a warm start to the base amount, and initialized the other affine coefficients as zero. Note that this is, by design, already a feasible solution to the adaptive model, ensuring that the adaptive model always gave feasible output.

Each dataset (both training and testing) contained fifty generated scenarios for any computational experiment. All our instances were solved with Gurobi 6.0.0 on a Intel Xeon E5687W (3.1 GHz) processor with 16 cores and 128 GB of RAM.

To investigate the quality of the solutions resulting from our models, we ask the questions of a) whether our robust and adaptive models lead to fewer stockouts, b) what effect this has on service cost, and whether this suggests a possible reduction of the vehicular fleet size.

2.5.1 Stockouts

We first investigate whether our robust and adaptive inventory routing models lead to a reduction in stockouts. As mentioned above, the three formulations were solved sequentially (nominal-robust-adaptive) on problems of a fixed vehicle fleet size and a decision horizon of 151 days. Figures 2-2, 2-3 and 2-4 show the average percentage of customers who experienced stockouts for the nominal, robust and adaptive formulations respectively. The standard variation in the (temperature and demand) noise was scaled appropriately for each dataset so that the models were trained on a base value of 5.

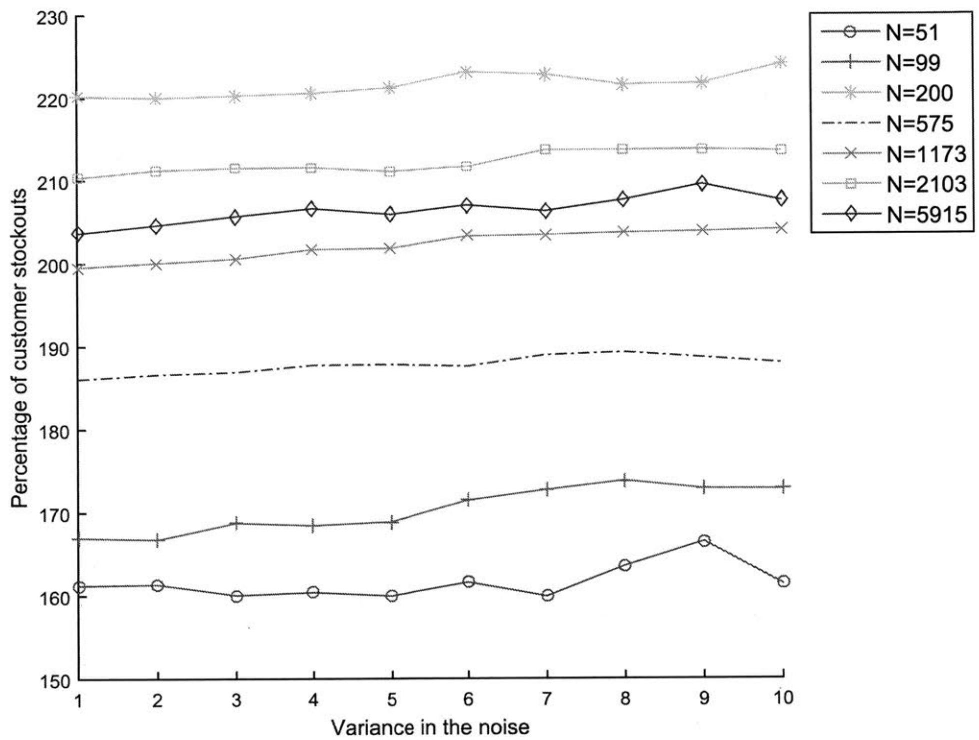

Figure 2-2: Average stockout percentages for the nominal solutions for data sets of different sizes.

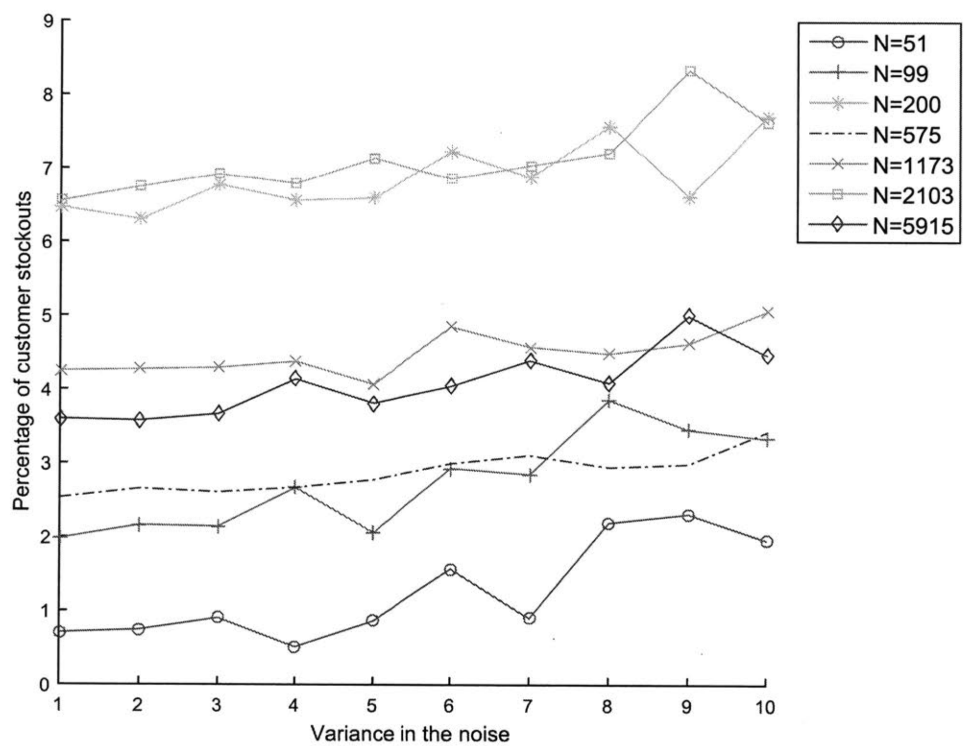

Figure 2-3: Average stockout percentages for the robust solutions for data sets of different sizes.

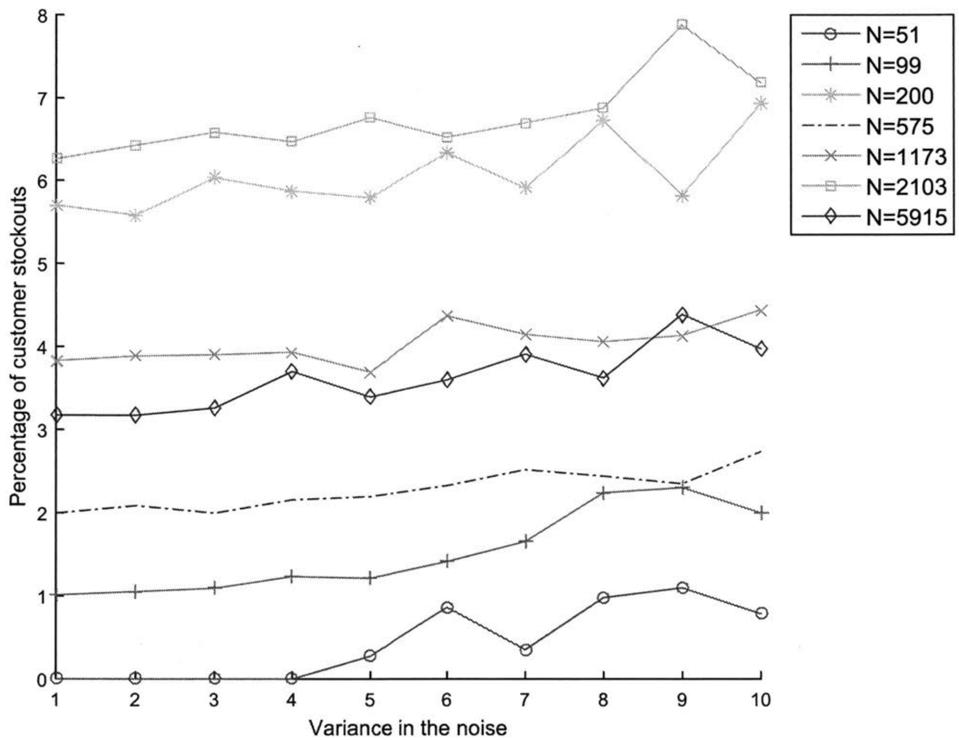

Figure 2-4: Average stockout percentages for the adaptive solutions for data sets of different sizes.

We observe that across data sets, while the nominal formulation had between 160%-225% of customers stocking out (some customers experienced stockouts multiple times), the robust formulation decreased this to below 9% of all customers, and in most cases half of that or even less. Stockouts decreased further by 0.5%-1% of all customers from the robust to the adaptive formulation for all the data sets, i.e. a decrease of over 5% in stockouts from the robust formulation. We also notice that the robust and adaptive formulations were less sensitive than the nominal formulation to increasing variance in the noise (i.e., errors in tuning the robust parameters).

We explore how the reduction in stockouts was distributed across the fifty scenarios for each data point in the above experiment. Figure 2-5 shows the standard box plot for the reduction in robust model stockouts as a fraction of nominal model stockouts, for the uncertainty regime of the base level of variance in the noise. We observe that every scenario generated had at least 94% relative reduction in stockouts from the nominal to robust models, with an average of over 96% relative reduction for each dataset.

Finally, we ran experiments to determine whether we had chosen sufficiently many routes or neighborhoods to cover each customer. We found that as long as customers were included in 2-3 neighborhoods, any additional coverage was superfluous and did not result in further cost reductions. This comports with our view of a route as a superset of the true route that the vehicle actually takes, because it is unlikely that it improves our situation for a customer to be assigned to a vehicle that is not going to resupply other customers in their immediate neighborhood.

2.5.2 Service Cost

We next consider the effect on the cost of servicing the customers with the different formulations. As before, we solve problems of a fixed vehicle fleet size. To get the combined cost of the problem, we consider costs from two sources, namely: 1) the variable cost from the routes, which is the objective function of the optimization model, and 2) the cost of refuelling a customer who experiences a stockout. Because these stockouts occur randomly throughout the course of a time period, and must be

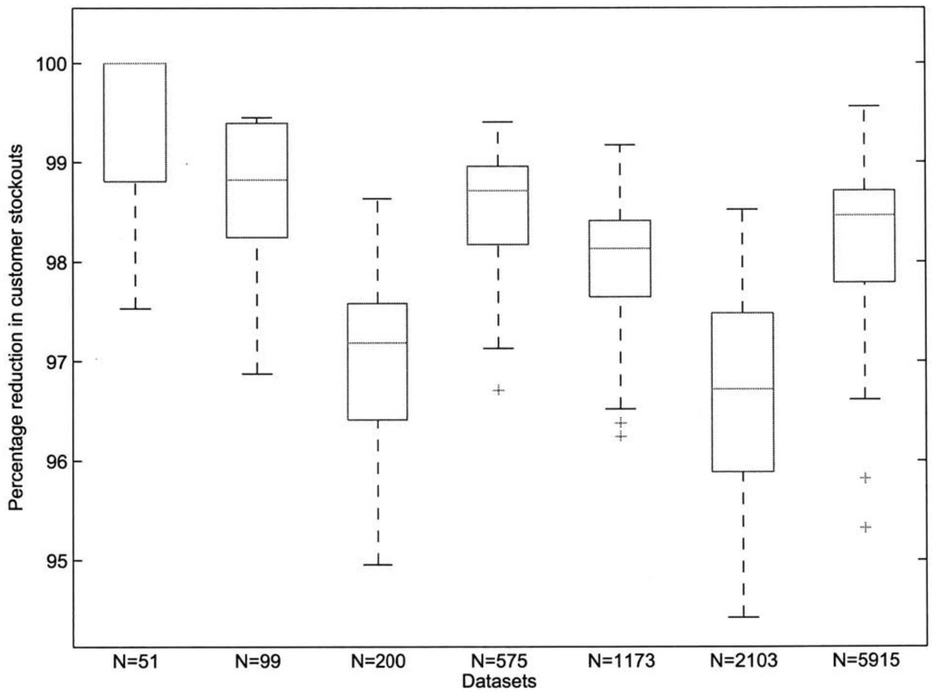

Figure 2-5: Interquartile and extreme values for the reduction in robust model stockouts as a fraction of nominal model stockouts.

addressed urgently, the planner must send an emergency refuelling vehicle out each time a customer stocks out. We assume that due to the reduced efficiency of the smaller emergency refuelling vehicle, its cost per unit distance is twice that of the usual refuelling vehicle fleet.

Table 2.1 compares the combined cost for the respective models, along with the percentage gap compared to the best lower bound the solver could find within the time limits we set. C_N, C_R and C_A are the combined costs of the nominal, robust and adaptive models respectively, while G_N, G_R and G_A are the respective provable duality gaps output by the Gurobi solver.

In all cases, the robust model had a combined cost no higher than 86% of the nominal model's. With larger data sets of over a hundred customers, the cost savings were 44% or more of the combined cost of the nominal model. The adaptive model had a combined cost that was a further 0.2%-0.3% lower than that of the robust model, i.e. an additional 0.1%-0.3% decrease to the combined cost of the nominal model beyond the improvements from going to the robust model.

To ensure that the solver gaps were not indicative of any major problems, we allowed the two smallest nominal models to run for several hours. At that point, the optimality gaps were below 5%, with no further change to the solutions. This suggests that any further improvement in these two cases, at least, would accrue to the robust and adaptive models. Similarly, no improvements were made to the larger cases after several more hours of running time, though in these cases the optimality gaps did not decrease sufficiently to draw the same conclusion.

Customers	C_N	G_N	C_R	G_R	C_A	G_A	C_R/C_N	C_A/C_N
51	16032	26.9	13791	37.2	13768	56.6	0.860	0.859
99	83527	45.7	66894	65.1	66684	66.5	0.801	0.798
200	5.295e6	77.3	2.966e6	74.3	2.959e6	75.5	0.560	0.559
575	1.001e6	1.61	459040	16.4	457547	18.9	0.459	0.457
1173	1.474e7	36.2	8.003e6	46.1	7.986e6	47.9	0.543	0.542
2103	3.190e7	31.8	1.373e7	38.0	1.370e7	39.7	0.431	0.429
5915	3.946e8	35.6	1.676e8	43.1	1.670e8	44.7	0.425	0.423

Table 2.1: Costs and solver gaps for data sets of different sizes.

2.5.3 Fleet Reduction

Finally, whereas in the previous subsections, the vehicle fleet size was constant for each data set, here we investigate the tradeoffs of reducing the vehicle fleet size. We focus on a single data set with $N = 575$, for which our previous experiments used a fleet of 11 vehicles.

To allow the models to output a solution even with an infeasibly small fleet size, we introduce slack variables into our model that allow the demand constraints to be relaxed for a steep penalty (we took this to be 10^7 times the amount of violation). Taking the combined cost introduced in Section 2.5.2, we now further add to this the fixed cost of a vehicle fleet of a given size, taken to be 10,000 per vehicle. Table 2.2 compares the new combined cost for the best solutions for vehicle fleets of different sizes. C_N, C_R and C_A are the combined costs of the nominal, robust and adaptive models respectively, while S_N, S_R and S_A are the average number of customers who stock out.

Vehicles	C_N	S_N	C_R	S_R	C_A	S_A
11	959490	852.9	529377	15.12	527747	12.38
10	1013552	932.54	519377	15.12	517747	12.38
9	1003552	932.54	509377	15.12	507747	12.38
8	993552	932.54	499377	15.12	497747	12.38
7	983552	932.54	556791	110.00	548587	95.04
6	991370	976.52	546791	110.00	538587	95.04
5	981370	976.52	1493560	1888.88	1493049	1860.24
4	1458104	1823.82	1483560	1888.88	1482509	1885.62
3	2157939	3233.10	2040526	2998.22	2038394	2994.58

Table 2.2: Stockout percentages for $N = 575$ with different fleet sizes.

We observe that the robust and adaptive solutions allow us to decrease the fleet size to 8 without increasing stockouts, and so decrease the combined cost. Decreasing the fleet size below 8 leads to an increase in combined cost for the robust and adaptive models, as demand is shifted from scheduled refuellings to emergency refuellings. On the other hand, with the nominal model, removing even one vehicle leads to an increased combined cost, as the savings from the smaller fleet size are lost to increased refuelling costs from the increased numbers of customers who stock out.

With five vehicles or fewer, the robust and adaptive solutions are unable to find high-quality solutions; because our penalty is applied to the total unmet demand, these models minimize this by spreading the shortfall out over a large number of customers, suggesting that increasing the fleet size is crucial to reduce the emergency refuellings - in the worst case, we are experiencing over five times the number of stockouts as we have customers. At this point, our fleet sizes are highly infeasible for the models, and a significant part of the "cost" is from the penalty from the slack variables.

However, with at least six vehicles, we get not only a significant cost decrease, but we also observe that the adaptive solution has 81-86% of the number of stockouts that the robust solution has.

2.6 Discussion

We have presented robust and adaptive formulations for the finite horizon inventory routing problem that are tractable for ~6000 customers. For an uncertainty set where customers' demands demonstrate limited dependence, where the usual methods of robust optimization are insufficient, we have constructed an algorithm that allows us to find worst-case scenarios deterministically (robust formulation) or relative to a candidate solution (adaptive formulation). We have shown a significant decrease in stockouts (over 94% in all test cases) for our models, translating to a 14% decrease in cost for the supplier. In addition, we have shown that our models, with slack variables, are capable of providing further cost savings through a reduction in the vehicle fleet size.

While our work here has been in the context of a heating oil problem, it is applicable more broadly to other problems where the customer demand satisfied by a Vendor-Managed Inventory paradigm can be modeled by a tractable uncertainty set. Such problems might include beverages in vending machines, or more recently, bike-sharing in cities, where demand is dependent on temperature.

We would also like to explore possible improvements in the provable lower bounds on our solutions, for example along the lines of [33]. Other improvements include more sophisticated ways of modelling emergency refuelling routing decisions, and smarter ways of managing fuel quantities dynamically.

Chapter 3

Robust Purchase Execution

3.1 Introduction

We consider the problem of a planner who has to meet the demand of customers who have an uncertain rate of consumption of some commodity. The planner has to devise a strategy to meet this demand while minimizing cost, which involves decisions not only on when to make the purchases, but also from which source to purchase from.

The planner has a choice of suppliers from whom to purchase this commodity, but is constrained both by limited storage capacity and by limited bandwidth to transport the commodity from suppliers to storage, and from storage to customers. The planner thus has to devise a strategy to minimize cost that is feasible for these constraints, and meets the customers' demand.

As the planner faces fluctuating prices, which might incorporate seasonal trends, exogeneous price shocks, or just random variation, there is uncertainty in the prices that will be available to him/her over the course of the time horizon, and the chosen strategy should also model the uncertainty in the underlying purchase price of the commodity.

These changes might depend not only on an exogeneous price, but also on the quantities that the planner wishes to purchase in each time period. This implies that when executing the purchase of a large quantity of some commodity, a planner who wants to minimize the total cost must consider the effects that the way the purchase is structured and carried out have on the price. For instance, a supplier might be willing to offer discounts on a larger order, or to a customer who has purchased greater quantities of the commodity in recent periods. This adds an additional layer of opportunity for planners to optimize their purchases, but also requires them to take into account the additional uncertainty in how the actual price paid differs from the market price, which is itself already an uncertain quantity.

There are several time horizons from which this problem may be approached, ranging from initial planning decisions for an entire planning season to tactical and near-real-time purchasing decisions. In this chapter, we focus on a tactical buying problem, where the planner has a single deterministic forecast of demand and wants to spread the purchases of the commodity across the time horizon to minimize worst-case cost.

To contextualize the above discussion throughout this chapter, the main application that will be considered is in the domain of heating oil. Take, for example, a typical company that provides heating oil in residential areas. A company of this nature in New England might have a customer base spanning north central Massachusetts and southern New Hampshire, with around 10,000 customers. The total demand of these customers can be estimated over various time scales, ranging from the next few days to the entire winter season (November to March), with correspondingly increasing uncertainty.

For simplicity, in most of our discussion we will assume that the planner in this company is trying to purchase a total quantity of heating oil over some time horizon - a month, maybe, and is only concerned that at the end of the time horizon, this quantity is acquired. In Section 3.4.1, we discuss the generalized case where the planner has demand constraints that have to be satisfied periodically.

Additionally, the planner has a few regular suppliers from which heating oil can be purchased. In each time period, the suppliers are willing to offer a discount below that day's commodity market price, in order to incentivize larger business from the heating oil company. The magnitude of this discount depends both on the size of the order being made, and on the previous purchasing history of the company. The planner has thus to consider not only when to make a purchase, but which supplier to make it from.

Finally, the planner would still like to be able to apply the insights from the optimization models formulated here, in the event where there is a change in some of the parameters. For example, suppose that the planner wants to increase the total quantity purchased, or to decrease the worst-case risk. Would this lead to purchases being made at different times, or from different suppliers? How much would the impact on the total cost be?

Instead of re-solving the optimization models, we use a modern machine learning approach, and so generate useful insights, allowing the planner to make intelligent predictions about the optimal solution and objective value, and qualitative features of the solution. In particular, we use recent work in optimal decision tree methods to predict features of the model and solution.

Our contributions in this work can be summarized as follows:

1) **Robustness.** We present a robust model of the purchase execution problem that captures the dependence of cost both on the underlying market price, which we assume to experience fluctuations captured in an ellipsoidal uncertainty set, and daily discounts offered by the sellers that the planner has to choose between.

2) **Quality of solutions.** We show that the solutions of our model lead to substantially decreased worst-case cost and cost variance, with the tradeoff of a very small increase in average cost.

3) **Learning.** We use optimal decision tree methods to gain insight on the optimal solutions for our model. We show that after generating these decision trees, they can be used on new parameters to yield a high-quality purchasing strategy without having to solve the full optimization model again. In our computational experiments, the decision tree strategy always gave a worst-case cost within 2% of that from the optimization model, and on average had a worst-case cost within 0.2%.

The remainder of this chapter is structured as follows: in Section 3.2, we survey the related literature and discuss why a robust formulation is needed. In Section 3.3, we formulate a single-seller version of our problem and extend our discussion to the multi-seller problem in Section 3.4. We present our computational experiments and analysis in Section 3.5. In Section 3.6, we use regression trees to learn and predict useful features of the solutions to our model, and in Section 3.7, we show that these decision trees can be used to generate high-quality solutions directly. We finally conclude in Section 3.8 with overall discussion and some future research directions.

3.2 Related Work

3.2.1 Purchase Execution

The problem of how best to execute a purchase under price uncertainty has been investigated over the last sixty years. In the area of raw material or commodity purchases, [78] first studied a procurement problem with probabilistic prices, assuming known forecasts for demand and price. [106] proposed a dynamic programming model that yields decision pricebreaks, and quantities to purchase if the price falls below the pricebreaks. While the original model assumed a known and stationary price distribution function, it could be extended to non-stationary distributions, with the caveat that it requires the independence of the price density functions in successive time periods, and thus is better used as a heuristic for planning, when the density functions are outputs of a forecasting model where that assumption applies. [89] later showed the optimality of the price-dependent basestock policy for deterministic demands, but non-stationary price probability distibutions. [107] explicated the inadequacy of classical inventory models in this area of purchasing raw materials, and since then more work has been done on expanding the models with stochastic prices, e.g. [11, 98]. The interested reader is directed to [108] for more recent summaries of the work in this area, and [10] for the additional consideration of financial instruments.

A related problem, from the field of finance, is that of executing a trading strategy, whether for a single risky asset or a portfolio. The rapid growth of institutional investors such as banks, hedge funds, and mutual funds over the past decades has increased the importance to planners of modeling the market response to the way their intended strategy is carried out. [130], for instance, observes that "implementation shortfall" - the performance loss due to execution costs - was largely responsible for the surprising phenomenon of the paper portfolio based on the Value Line rankings outperforming the market by almost 20% per year from 1965 to 1986, while in reality the Value Line fund was only able to outperform the market by 2.5% a year. Trying to minimize execution costs has led to much fruitful research on the *price impact* of portfolio executions, which describes the effect that making a sale or purchase can have on the underlying price, not just at the time of the transaction, but also for future orders of the strategy.

One approach to handling risk that is common to both problems is the mean-variance approach, where a linear combination of the average utility and its variance is used to account for the planner's risk preferences. In the portfolio execution problem, this utility would be the (minimized) execution cost of selling the portfolio, with the additional constraint that the planner should have completely exited their position by the end of the time horizon.

Strategies to address this problem can be either static [4] or dynamic [45, 39], and more recent work has shown both that the efficacy of these strategies can be sensitive to estimation errors in the impact matrices and other parameters [119], but that regularized robust optimization can be a promising approach to reduce the effect of these estimation errors on the optimal strategy [120].

One important difference between our problem and the portfolio execution problem, as commonly formulated, is that the linear price impact term here does not result from the market price dynamics, but rather the individual sellers rewarding frequent customers. This has the effect of rewarding a large purchase, rather than penalizing it. In other words, while the typical portfolio execution strategy experiences a negative temporary price impact, here we have a positive temporary price impact that encourages larger purchases.

In addition, much of the prior research considers purchases that are sufficiently large that they affect the market price, such as those made by large financial institutions holding diverse portfolios. In contrast, we do not consider this to be a realistic assumption for our problem domain, and so assume that the underlying exchange price is not affected over time by the purchases of the planner. Nevertheless, past purchases do have an impact on the future price that the planner actually pays. The mechanism by which the quantity of commodity purchased affects the per-unit price, in our model, is through the discounts offered by the seller(s).

The previous literature in commodity purchasing has historically dealt with the uncertainty in the underlying market price by treating it as a stochastic variable or Markov process, or in some cases discretizing the uncertainty through sample-based worst-case modeling. A more recent paradigm that has been shown to be useful in formulating optimization models to make decisions under uncertainty is Robust Optimization (RO) (for instance [47, 21, 67, 31]). This set-based approach leads to solutions that are guaranteed to yield feasible solutions for all uncertain parameters in a carefully chosen *uncertainty set*, and often leads to tractable models which require weaker assumptions about the uncertain parameters to be made than when using stochastic formulations. In practice, RO formulations have been found to yield worst-case solutions that are competitive with the optimal deterministic solution, and perform significantly better in worst-case scenarios than solutions found without considering uncertainty. They also tend to be less susceptible to errors in parameter estimation or structural misspecifications [91, 48].

Although robust optimization has been applied in many cases to the portfolio execution problem with much success (E.g., [31, 79, 91, 114, 122]), there are some important structural differences between that problem and the purchase execution problem we consider, as we have mentioned above, and there do not appear to have been any attempts to apply robust optimization to the purchase execution problem in the current literature.

3.2.2 Machine Learning

Decision trees are currently one of the most frequently used predictive modeling approaches in solving classification problems. Given various attributes of the training data, a tree-like structure can be built, where each node represents a recursive partitioning of the feature space, and each of the leaves (the smallest partitions) is assigned a label, or classifier. The tree can then be used predictively on future data points, classifying them according to the decisions implied by following the tree from its root node.

The main advantage of decision trees over other predictive models in classification problems is their interpretability. In many areas such as healthcare, the split-and-label structure allows decision trees to be easily communicated graphically to non-experts, and more closely mirrors the actual decision-making process that humans use [105].

It has long been known that finding the *optimal* decision tree for training data, i.e., the tree that takes the absolute best decision at each split, is a NP-hard problem [115]. Although there have been several methods developed to construct optimal univariate decision trees, none of them have been able to solve for certifiably optimal decision trees in a reasonable length of time. These methods include linear optimization [24], continuous optimization [25], dynamic programming [62, 128], genetic algorithms [139], and more recently, stochastic gradient descent on an upper bound for the tree's empirical loss [124].

Another family of efficient enumeration approaches that have been proposed are T2 [13], T3 [141], and T3C [144], which create optimal non-binary decision trees up to a depth of 3. However, these trees lack the interpretability of binary decision trees, and their performances do not significantly improve on those of other heuristic approaches [141, 144].

Because of these practical limitations, the leading approach to constructing decision trees has been to determine the splits by utilizing a top-down approach. Classification and regression trees (CART) [52] begin at the root node, and determine a partition there (typically by minimizing an impurity measure), after which the procedure is repeated recursively at both child nodes. Other popular decision tree methods like ID3 [132] and subsequently C4.5 [133] also operate by similar recursive splits. However, as each split is made without consideration as to its impact on the future nodes in the tree, these *greedy approaches* can lead to trees that do not perform well in classifying out-of-sample points because they do not capture well the true underlying characteristics of the dataset.

In addition, top-down approaches face the problem of strong splits being hidden behind weaker splits. A tree that is too complex risks being overfitted to the training data, and so complexity has to be penalized in some way when generating a decision tree. If the penalties on complexity are too high while growing the tree, the first weaker split might not be selected and so the best tree might not be discovered. One approach to resolve this problem is *pruning*, where the decision tree is trained in two phases. First, it is allowed to grow as deep as computational resources allow, before the complexity penalty is applied to reduce, or prune, the branches of the decision tree. Other heuristics that have been proposed are *lookahead* heuristics (e.g., IDX [125], LSID3 and ID3-k [77]) that optimize the split at each node based on slightly deeper trees rooted at that node. However, it is unclear whether these methods actually lead to trees that avoid the "pathology of decision tree induction" [121] and are more generalizable.

More recently, the belief that the optimal decision trees cannot be tractably found has been re-examined. Even though it has long been recognized that natural MIO formulations exist for many statistical problems [12], it was hitherto thought that these were intractable for even small or medium-sized instances of these problems. However, led by improvements in mixed-integer optimization (MIO) in the last few decades [51, 123], both in MIO solvers such as Gurobi [97] and CPLEX [63], and in the computational power available to practitioners, the performance improvement in solving MIO problems has been estimated at a speedup factor of approximately 800 billion, leading to remarkable success when applying the modern optimization lens to many of these statistical problems [46, 41, 43, 42]. Following this approach, [34] introduced *optimal classification trees*, a novel MIO formulation which yields the optimal decision trees for axes-aligned splits. This formulation is tractable over real-world datasets of sizes in the thousands, and demonstrates significant improvement over CART and other heuristics on benchmark examples. Finally, work has been done to expand the MIO methodology from classification problems to other prescriptive problems, in particular optimal regression trees [35], where instead of a single classification label for each leaf, the model generates a prediction, or even a regression model.

3.3 Single-seller Model

We first consider a heavily simplified model where there is only one seller. This allows us to exclude any effects of competition, and develop intuition about how our model behaves across time. By analyzing the uncertain price component separately from the discount components, we will show the role of the robust parameter in balancing two qualitatively different solutions.

Consider a planner who wishes to execute a purchase of $\bar{S}$ units over a fixed and discretized time interval $[1, T]$. The planner only has a single supplier to purchase from, and so the only decisions that have to be made are the quantities to be purchased in each time period, S_t. We assume that we can construct an uncertainty set capturing the fluctuations in the market price. To be precise, in time period t, the price $P_t = P_{t-1} + \epsilon_t$, and we have some set $\mathcal{U}$ for which we assume that $\boldsymbol{\epsilon} \in \mathcal{U}$. We discuss the construction of $\mathcal{U}$ in greater detail in Section 3.3.1. P_0 is the (known) market price of the last time period before the model is solved.

We assume that over the feasible region, the planner's purchase is not large enough to observably impact the market price of the commodity in general. However, the seller might wish to incentivize a regular customer by offering them discount pricing. When formulating our model, this discount should be specific to the seller, and depend both on the size of a given purchase, and on the size of the total quantity purchased up to that point. This captures two kinds of behavior that a seller might wish to incentivize, namely larger purchases and repeat purchases, respectively.

To make the model sensible, we set an upper bound on the quantity that can be purchased in a single time period, such that over the feasible region, these discounts can be represented as being linear in the quantities they depend on. For the moment, we assume that the coefficients of these discounts are fixed and known to the planner; later on we will consider the effects of errors in our estimates of these discounts. To better understand the qualititative properties of our model, we will not consider capacity constraints here, but we note that adding such constraints would not significantly decrease the tractability of the model.

With these assumptions, the actual price per unit the seller is offered at time period t, Q_t, is given by $P_t - \theta_t S_t - \omega_t \sum_{k=1}^{t-1} S_k$, where the discount coefficients θ_t, ω_t are positive. The full single-seller model with price uncertainty is then given by:

$$
\begin{aligned}
\min_{Q_t, S_t} \max_{\epsilon \in \mathcal{U}} \quad & \sum_{t=1}^{T} Q_t S_t && (3.1)\\
\text{s.t.} \quad & \sum_{t=1}^{T} S_t = \bar{S}, && (3.2)\\
& Q_t = P_t - \theta_t S_t - \omega_t \sum_{k=1}^{t-1} S_k, && (3.3)\\
& P_t = P_{t-1} + \epsilon_t, && (3.4)\\
& 0 \leq S_t \leq \hat{S}. && (3.5)
\end{aligned}
$$

3.3.1 Constructing $\mathcal{U}$

Following the well-established methodology of robust optimization, we formulate our uncertainty set to represent all the instances of uncertainty that our model should take into account. Depending on the assumptions made about the data, uncertainty sets can be derived in many ways, e.g., from probabilistic laws [19], or statistical hypothesis tests [38].

To better illustrate our approach, here we consider a simple uncertainty set, where the additive price uncertainty across time periods is bounded by a spherical uncertainty set, and the robust parameter r allows us to control the conservativeness of the model as per the planner's risk preferences. The resulting uncertainty set is:

$$\mathcal{U} = \left\{ \epsilon : \sum_{t=1}^{T} \epsilon_t^2 \leq r^2 \right\}.$$

In Section 3.3.2, we discuss solving models with more sophisticated uncertainty sets. Our approach remains computationally tractable for convex uncertainty sets in general.

3.3.2 Solving the Single-seller Model

To solve this model, we can explicitly calculate the worst-case uncertainty for a candidate solution. We can use (3.4) repeatedly to rewrite (3.3) as:

$$Q_t = P_0 - \theta_t S_t - \omega_t \sum_{k=1}^{t-1} S_k + \sum_{k=1}^{t} \epsilon_k.$$

The component of the objective function (3.1) that is affected by price uncertainty is then $\sum_{t=1}^{T} \sum_{k=1}^{t} \epsilon_k S_t$. Switching our indices, we conclude that the worst-case values of ϵ_t are given by the solution to the following optimization problem:

$$\max_{\epsilon \in \mathcal{U}} \quad \sum_{t=1}^{T} \sum_{k=t}^{T} S_k \epsilon_t. \tag{3.6}$$

It is easier to see that this is correct by observing that a change in ϵ_t will affect exactly the prices of that time period and beyond, i.e., only the coefficients S_k for $k \geq t$.

But for our uncertainty set, this is just the point on the r-radius sphere centered at the origin that also lies on the ray of the objective vector in (3.6), and so the worst-case uncertainty is given in closed form for all t as:

$$\epsilon_t^* = \frac{r \sum_{k=t}^{T} S_k}{\sqrt{\sum_{l=1}^{T} \left(\sum_{k=l}^{T} S_k \right)^2}}. \tag{3.7}$$

The single-seller problem with only price uncertainty, (3.1), then becomes equivalent to

$$
\begin{aligned}
\min_{S_t} \quad & \sum_{t=1}^{T}\left(P_0-\theta_t S_t-\omega_t\sum_{k=1}^{t-1}S_k+\sum_{k=1}^{t}\epsilon_k^*\right)S_t, && (3.8)\\
\text{s.t.} \quad & \epsilon_t^* = \frac{r\sum_{k=t}^{T}S_k}{\sqrt{\sum_{l=1}^{T}\left(\sum_{k=l}^{T}S_k\right)^2}}, && (3.9)\\
& \sum_{t=1}^{T}S_t=\bar{S}, && (3.10)\\
& 0\le S_t\le \hat{S}, && (3.11)
\end{aligned}
$$

which, although not being convex, does have a smooth objective function and constraints, and is thus in a form that we can use modern nonlinear optimization solvers to find solutions for in a reasonable period of time. We implemented the model in JuMP [74] and solved it with the solvers Ipopt [146] and MUMPS [5, 6]. In Section 3.5, we will provide empirical evidence about the computational tractability of this solution for realistic sizes.

To address the problem of the solver only finding local optima, we tested our model with 100 different warm starts, each time drawing them uniformly from the space of all S_t satisfying the equality constraint, and took the output with the lowest cost.

In general, as long as the uncertainty set is convex, we can show that there are known ways to take the robust counterpart to the inner problem, rewriting it as a minimization problem which remains tractable. To do this for a general convex uncertainty set, $\mathcal{U}_{con}$, we rewrite (3.1) instead as:

$$
\begin{aligned}
\min_{S_t} \quad & \left(\sum_{t=1}^{T}\left(P_0-\theta_t S_t-\omega_t\sum_{k=1}^{t-1}S_k\right)S_t+\max_{\epsilon\in\mathcal{U}_{con}}\sum_{t=1}^{T}\left(\sum_{k=1}^{t}\epsilon_k\right)S_t\right),\\
\text{s.t.} \quad & \sum_{t=1}^{T}S_t=\bar{S},\\
& 0\le S_t\le \hat{S},
\end{aligned}
$$

and observe that the inner problem is a robust linear optimization problem in S_t with a convex uncertainty set, for which it is known [20, 92] that the robust counterpart can be tractably reformulated using Fenchel duality. As before, the reformulated problem is now amenable to solving by modern nonlinear optimization solvers.

3.3.3 Analytic Solutions for Nominal and Non-discounted Cases

We derive here two analytic solutions to simplifications of the problem, both of which have simple descriptions. These descriptions are significant because in our computational experiments, we found that using them as warm starts led to solutions that were very close to optimal, and additionally had the benefit of being highly interpretable.

The first is a local optimal solution for the case where the robust parameter r is zero, assuming that the discounts are sufficiently small and constant over time.

Theorem 2. *Assume that for all t, $\theta_t = \theta$, $\omega_t = \omega$, and $P - \omega\bar{S} - (2\theta - \omega)\frac{\bar{S}}{T} \geq 0$. Assume that $\hat{S} \geq \bar{S}/T$. Let $r = 0$. Then a local optimal solution to (3.1) is $S_t = \bar{S}/T$ for all t. In fact, this is the only possible local optimum for which $S_t \in (0, \hat{S})$ for all t.*

Proof. The assumption that $\hat{S} \geq \bar{S}/T$ is necessary, or the problem is clearly infeasible as the planner cannot purchase enough to satisfy Constraint (3.2). On the other hand, $\hat{S} \geq \bar{S}/T$ implies immediately that $S_t = \bar{S}/T$ for all t is a primal feasible solution. Note also that our bounds on S_t and capacity constraint satisfy the regularity conditions of linearity constraint qualification for the Karush–Kuhn–Tucker (KKT) conditions.

Using the reformulation in (3.8), the nominal problem is:

$$\begin{aligned}
\min_{S_t} \quad & \sum_{t=1}^{T} \left(P_0 - \theta S_t - \omega \sum_{k=1}^{t-1} S_k \right) S_t, \\
\text{s.t.} \quad & \sum_{t=1}^{T} S_t = \bar{S}, \\
& 0 \leq S_t \leq \hat{S}.
\end{aligned}$$

We take the Lagrangean of this problem,

$$L(\boldsymbol{S}, \lambda) = \sum_{t=1}^{T} \left(P_0 - \theta S_t - \omega \sum_{k=1}^{t-1} S_k \right) S_t + \lambda \left(\bar{S} - \sum_{t=1}^{T} S_t \right) - \sum_{t=1}^{T} \alpha_t S_t + \sum_{t=1}^{T} \beta_t (S_t - \hat{S}),$$

where λ is the KKT multiplier for the equality constraint, and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the multipliers for the lower and upper bounds respectively. Then

$$\frac{\partial L}{\partial S_t} = P_0 - 2\theta S_t - \omega \sum_{k \neq t} S_k - \lambda - \alpha_t + \beta_t.$$

For any optimal $(\boldsymbol{S}^*, \lambda^*)$, the equality constraint must hold by primal feasibility, and so by stationarity we have:

$$\frac{\partial L}{\partial S_t}(\boldsymbol{S}^*, \lambda^*) = P_0 - 2\theta S_t^* - \omega(\bar{S} - S_t^*) - \lambda - \alpha_t + \beta_t = 0.$$

If we have $S_t \in (0, \hat{S})$ for all t, then by complementary slackness, any local optimum must have $\alpha_t = \beta_t = 0$ for all t. But then by symmetry over t, all S_t must be identical, and it follows immmediately from the equality constraint that $S_t = \bar{S}/T$ for all t.

This implies that if we have $\lambda = P - \omega \bar{S} - (2\theta - \omega)\frac{\bar{S}}{T} \geq 0$, we also have dual feasibility and our solution is in fact a local optimum by the KKT conditions; it also shows that if this assumption does not hold, then in fact we cannot have any local optima in the interior of our bounds for S_t. $\square$

For the various parameter settings that were tested in 3.5.1 and 3.5.2, we found that using this local optimum for the nominal problem as our warm start always yielded an objective value within a fraction of a percent of the best solution from 100 uniformly generated warm starts, and prevented the model from stopping at solutions that fluctuated greatly over time.

The other solution that we derive is the solution for the robust problem where there are no discounts, i.e., θ and ω are always zero.

Theorem 3. *Assume that for all t, $\theta_t = 0$, and $\omega_t = 0$. Assume that $\hat{S} \geq \bar{S}/T$. Then there exists an optimal solution to (3.1) where for some t, $S_k = \hat{S}$ for all $k \leq t$, and $S_k = 0$ for all $k \geq t$.*

Proof. We consider the problem of minimizing only the worst-case component given in (3.6):

$$\min_{S_t} \max_{\epsilon \in \mathcal{U}} \quad \sum_{t=1}^{T} \sum_{k=t}^{T} S_k \epsilon_t \tag{3.12}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} S_t = \bar{S}, \tag{3.13}$$

$$0 \leq S_t \leq \hat{S}. \tag{3.14}$$

Observe that (3.12) is the dot product of the vector $(\epsilon_1, \ldots, \epsilon_t)$ and the vector $(S_1 + \cdots + S_T, S_2 + \cdots + S_T, \ldots, S_T)$.

Now consider some candidate S^* with some $t_1 < t_2$, $S^*_{t_1} < \hat{S}$ and $S^*_{t_2} > 0$. Then for any $\boldsymbol{\epsilon} > 0$, we can do no worse by increasing $S^*_{t_1}$ and decreasing $S^*_{t_2}$ by some feasibly small δ, because each element of $(S_1 + \cdots + S_T, S_2 + \cdots + S_T, \ldots, S_T)$ is either decreased or unchanged. In particular, the worst-case value of the objective does not increase, and so we can progressively transform any candidate solution to the desired form in this way. $\square$

3.4 Multi-seller Model

We now formulate the model where the planner has to choose which suppliers to purchase from, and the quantities to purchase. To do this, we suppose that the planner has to choose not only the quantity to purchase in any given time period, but also which seller to make their purchase from, chosen from a (small) fixed set. We can then represent the quantities purchased in time period t by S_{jt}, where j indicates the choice of seller.

Extending the notation in the natural way, our formulation for the multi-seller model with price uncertainty is given by:

$$\min_{Q_{jt},S_{jt}} \max_{\epsilon_t \in \mathcal{U}} \quad \sum_{t=1}^{T} \sum_{j=1}^{J} Q_{jt} S_{jt} \tag{3.15}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} \sum_{j=1}^{J} S_{jt} = \bar{S}, \tag{3.16}$$

$$Q_{jt} = P_t - \theta_{jt} S_{jt} - \omega_{jt} \sum_{k=1}^{t-1} S_{jk}, \tag{3.17}$$

$$P_t = P_{t-1} + \epsilon_t, \tag{3.18}$$

$$0 \leq S_{jt} \leq \hat{S}, \tag{3.19}$$

which we will reformulate in a similar way to the single-seller case to solve tractably. As before, we use (3.18) repeatedly to rewrite (3.17) as:

$$Q_{jt} = P_0 - \theta_{jt} S_{jt} - \omega_{jt} \sum_{k=1}^{t-1} S_{jk} + \sum_{k=1}^{t} \epsilon_k,$$

whence the only component of (3.15) that is affected by price uncertainty is $\sum_{t=1}^{T} \sum_{k=1}^{t} \sum_{j=1}^{J} \epsilon_k S_{jt}$, and, switching indices, the worst-case values of ϵ_t are given by the solution to the following optimization problem:

$$\max_{\epsilon \in \mathcal{U}} \quad \sum_{t=1}^{T} \sum_{k=t}^{T} \sum_{j=1}^{J} S_{jk} \epsilon_t. \tag{3.20}$$

For our uncertainty set, this is the point on the r-radius sphere centered at the origin that also lies on the ray of the objective vector in (3.20), and so the worst-case uncertainty is given in closed form for all t as:

$$\epsilon_t^* = \frac{r \sum_{k=t}^{T} \sum_{j=1}^{J} S_{jk}}{\sqrt{\sum_{l=1}^{T} \left(\sum_{k=l}^{T} \sum_{j=1}^{J} S_{jk} \right)^2}}. \tag{3.21}$$

The multi-seller problem with only price uncertainty, (3.15), then becomes equivalent to

$$\begin{aligned}
\min_{S_{jt}} \quad & \sum_{t=1}^{T} \left(P_0 - \theta_{jt} S_{jt} - \omega_{jt} \sum_{k=1}^{t-1} S_{jk} + \sum_{k=1}^{t} \epsilon_k^* \right) S_{jt}, \\
\text{s.t.} \quad & \epsilon_t^* = \frac{r \sum_{k=t}^{T} \sum_{j=1}^{J} S_{jk}}{\sqrt{\sum_{l=1}^{T} \left(\sum_{k=l}^{T} \sum_{j=1}^{J} S_{jk} \right)^2}}, \\
& \sum_{t=1}^{T} \sum_{j=1}^{J} S_{jt} = \bar{S}, \\
& 0 \leq S_{jt} \leq \hat{S},
\end{aligned}$$

which again we can formulate and solve with JuMP and Ipopt. As before, tractable reformulations can be found with Fenchel duality for more general convex uncertainty sets.

Following the same approach as in the single-seller case, we can again obtain a local optimal solution for the nominal problem where the quantity purchased is constant over time. We can also obtain the same interpretation for the global optimal solution to the robust problem where θ and ω are always zero, which is to purchase the total quantity as quickly as possible.

3.4.1 Extending the Model to Multiple Periods

Up to now, we have presented the formulations for spreading out the purchase of a single quantity of the commodity. While this allows us to analyze the solutions of the models more clearly, it bears noting that lower execution cost is not the only reason that a planner might wish to spread out the purchase over time.

For example, a planner might not actually be able to execute the purchase at once because of capacity constraints, such as would occur if the model was being used to plan for demand over a longer period of time. Alternatively, the planner might not require the full quantity at once, and might wish to model this flexibility. We show briefly how to adapt the formulations to handle these kinds of constraints without significant impact on the models' tractability.

For the single-seller case, for instance, suppose that instead of requiring that a *total quantity* of $\bar{S}$ be purchased by the end of time T, the planner requires the *cumulative quantity* purchased up to each time period t to be within the interval $\left(\underline{D}_t, \overline{D}_t\right)$. To model these constraints, we replace the constraint (3.2) with:

$$\underline{D}_t \leq \sum_{k=1}^{t} S_k \leq \overline{D}_t, \qquad t = 1, \ldots, T,$$

which only adds $2T - 1$ constraints to the model. Note that this is a strict generalization of our previous formulation, as the total quantity can be modeled by setting $\underline{D}_T = \overline{D}_T = \bar{S}$ in the final pair of constraints.

Similarly, for the multi-seller case, suppose that the planner requires the cumulative quantity purchased up to each time period t, summed over all sellers, to be within the interval $\left(\underline{D}_t, \overline{D}_t\right)$. To model these constraints, we replace the constraint (3.16) with:

$$\underline{D}_t \leq \sum_{k=1}^{t} \sum_{j=1}^{J} S_{jk} \leq \overline{D}_t, \qquad t = 1, \ldots, T,$$

which again adds $2T - 1$ constraints to the model.

3.5 Computational Experiments

In this section, we investigate the performance of our formulation in a few different cases. We are particularly interested in the tradeoff of protecting against uncertainty, in terms of the impact on the average cost, worst-case cost, and cost variance. We would also like to understand how the models scale with instances of increasing size, both in the number of sellers, and the length of the problem horizon.

To simulate exposing our models to reality, we tested them against scenarios with similarly constructed uncertainty sets, but with the true uncertainty parameter r_* ranging up to ten times the maximum value of the uncertainty parameter we chose, $r_{max} = 2\sigma$, i.e., with time-independent normally-distributed noise up to ten times the magnitude of what we chose to protect against in our model. We generated 100 scenarios for each of these uncertainty parameters. We solved all our optimization models on a 256GB RAM, Intel E5-2660 v4 2.0 GHz CPU computer. Our models were written in JuMP [74], using Ipopt [146] for our nonlinear optimization solver, and we took the best solution found from 100 uniformly-drawn warm starts.

3.5.1 Single-seller Case

We first present computational results for the single-seller case with price uncertainty only. We first examine how the time taken to solve the model scales with T, the length of the time horizon. We let T vary from 1 to 50, set $\bar{S} = 10T/3$, $\hat{S} = 10$, $P_0 = 5$, and $\theta_t = 0.001$ and $\omega_t = 0.0002$ for all t.

The time taken by the solver to reach an optimal solution is shown in Figure 3-1. We observe that the model remains tractable, solving in under a minute for $T = 50$, and for shorter time horizons, it reaches an optimal solution in just a few seconds.

We next consider the actual cost and cost variance for different parameter settings, listed in Appendix B.

In the case where the robust parameter was correctly specified (i.e., it matched the actual level of the noise), we observed a small increase in average cost, no more than 1.8% on average for any of the other parameter settings. In general, we found that even for the other cases where the robust parameter has been misspecified, robustifying the problem did not significantly increase the average cost that was actually observed.

On the other hand, Figure 3-2 shows that the variance of the true observed cost

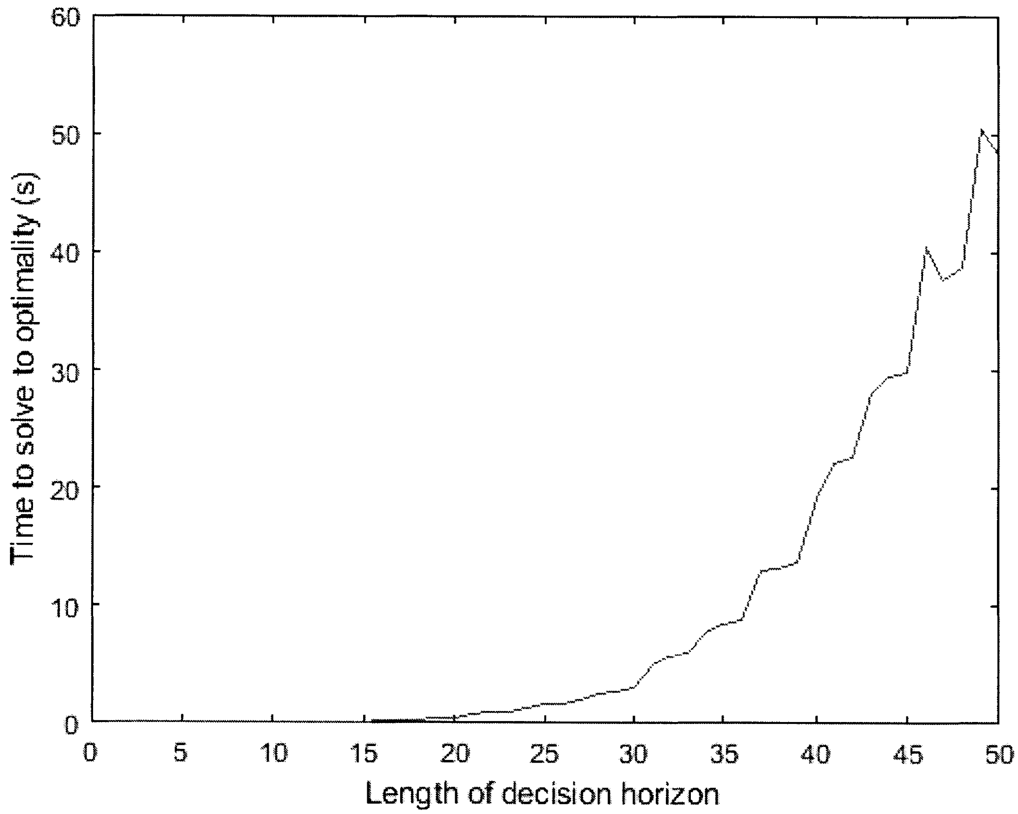

Figure 3-1: Solving time for different time horizons, one seller.

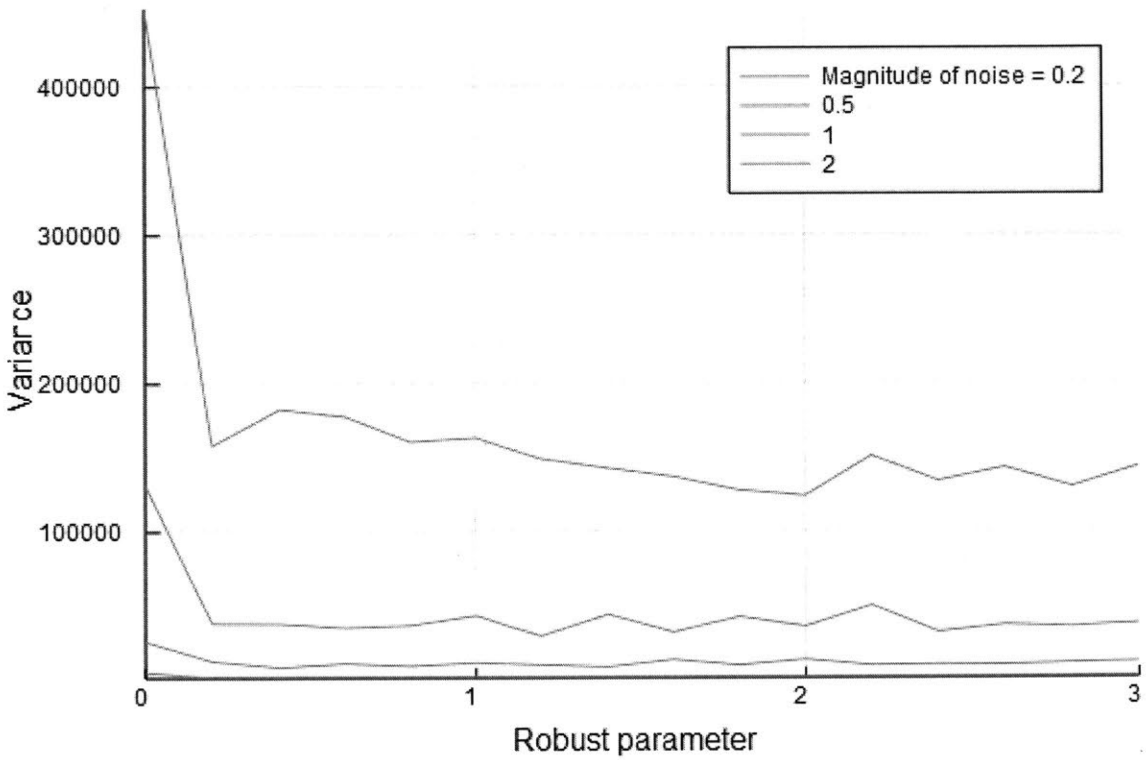

Figure 3-2: Cost variance for different values of the robust parameter r.

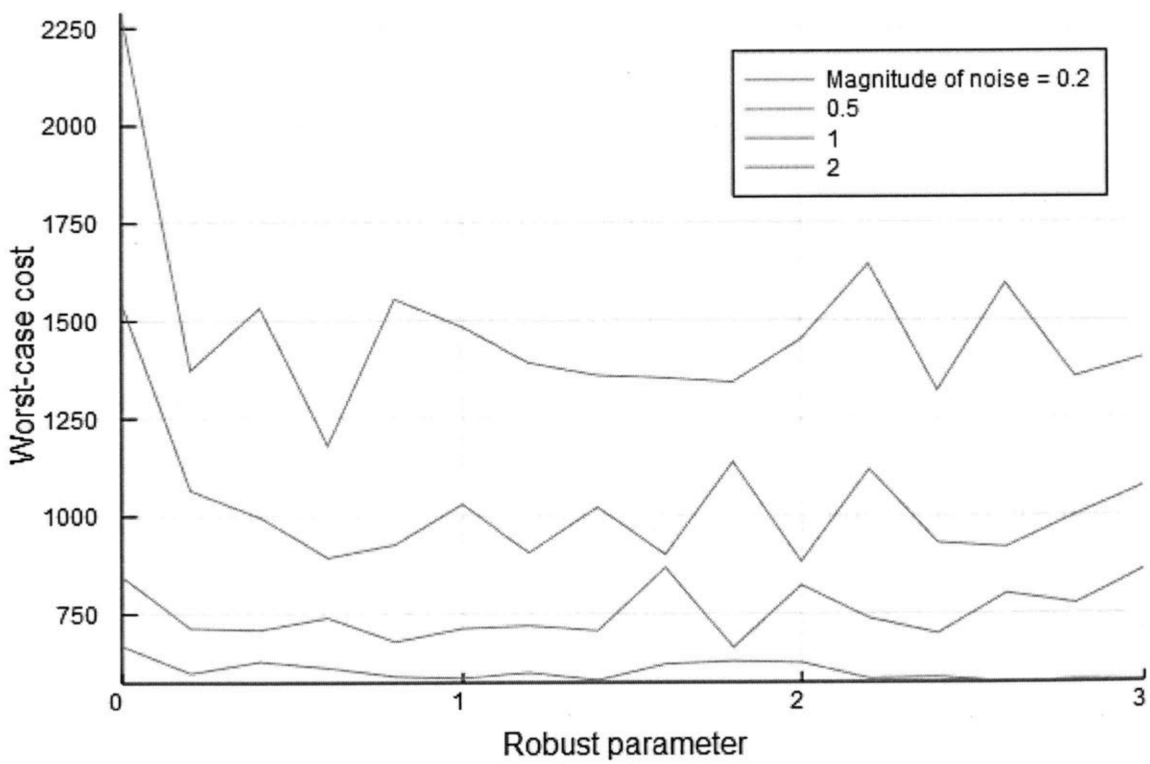

Figure 3-3: Worst-case costs for different values of the robust parameter r.

over the 100 scenarios drops rapidly as we introduce robustness to the model, then levels off. Similarly, Figure 3-3 shows that adding minimal protection ($r = 1$) to the model reduces the worst-case cost to about half of that which was observed in the nominal case, regardless of the true magnitude of the noise. However, in our simulations, after the variance of the observed costs levels off, we see that increasing the robust parameter further leads to over-conservative solutions, and performance suffers without adding extra protection for the planner.

We note in particular that using $S_t = \frac{\bar{S}}{T}$ as a warm start caused us to attain solutions that looked like either of the analytic solutions that we derived in Section 3.3.3. We observed that the actual solutions tended to shift sharply between the two analytic solutions as the robust parameter increased. As we described previously, one of these solutions is the optimal nominal solution, where the planner purchases an equal amount of the commodity in each time period. Introducing increasing uncertainty to the model induces the optimal solution to move closer to a recommendation that the planner buy as much as of the commodity as possible every time period, until the total demand has been met.

We interpret this transition as the onset of the effect of price uncertainty on the model. The abrupt transition between these two states occurs when the model decides that avoiding the uncertainty of potentially higher prices at the end of the time horizon is no longer worth spreading out the purchases to maximize the discounts from the seller.

In our simulations, we observed that the solution we obtained from using this warm start was always less than 1% suboptimal compared to the actual best solution from randomizing the warm starts. This suggests that we can find good solutions that have simple and interpretable descriptions.

3.5.2 Three-seller Case

We increase the number of sellers from one to three. Again, we begin by examining the tractability of the model, shown by how the solution time scales with T, the length of the time horizon. We let T vary from 1 to 50, set $\bar{S} = 10T/3$, $\hat{S} = 10$,

$P_0 = 5$, and $\theta_{jt} = (0.001, 0.0095, 0.009)$ and $\omega_{jt} = (0.0002, 0.00021, 0.00023)$ for all t. The time taken by the solver to reach an optimal solution is shown in Figure 3-4. We observe that the model, while naturally slower than the single-seller case, remains reasonably tractable, solving in under a minute for $T = 20$, and just over 10 minutes for $T = 50$.

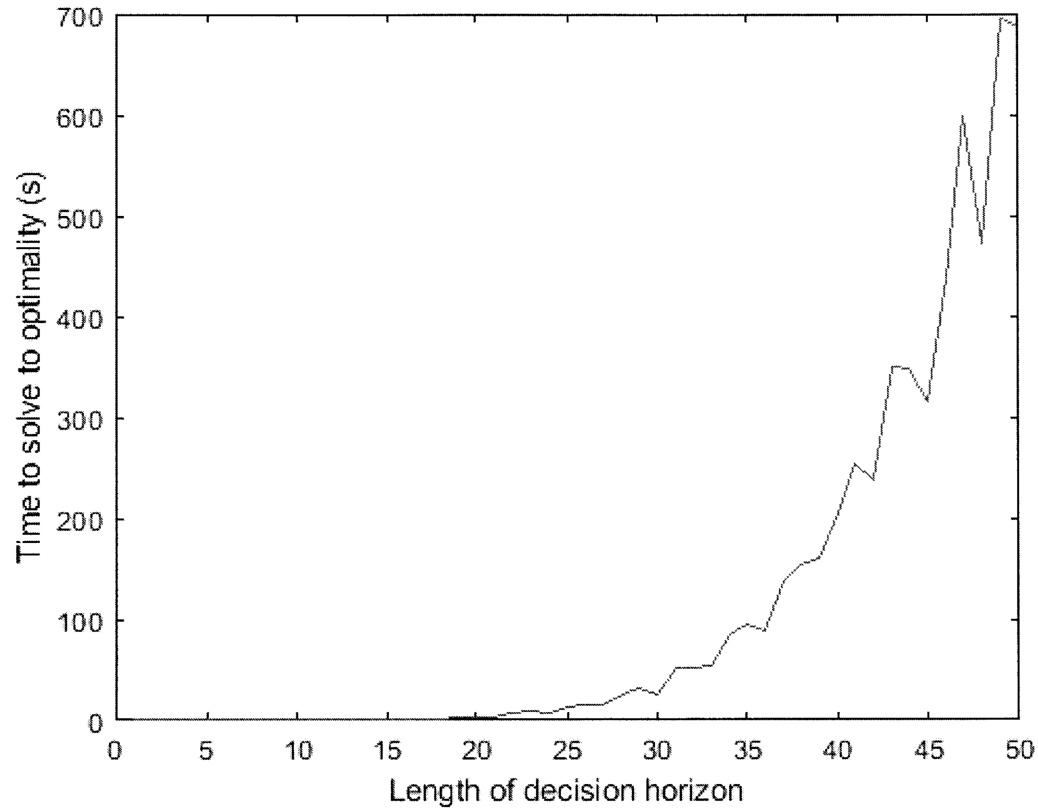

Figure 3-4: Solving time for different time horizons, three sellers.

In the single-seller case, we observed a qualitative shift in the solutions, driven by the dominating factor in the cost uncertainty. A similar, though more complex, shift is seen in the multiple-seller case. We illustrate this phenomenon with three cases, identical in all parameters except for the robust parameter r.

We consider three sellers, i.e., $J = 3$. As before, we set $T = 30$, and suppose that the discount parameters are constant and known for each run. For our multi-seller experiments, we required the planner to buy $100J$ units in total, but still no more than 10 units in a single time period. We now consider varying values of θ_{jt} and ω_{jt}, where we are particularly interested in the situation where no seller dominates another, i.e., has a higher θ_{jt} and ω_{jt} than the other. To this end, we set $\theta = (0.001, 0.0095, 0.009)$ and $\omega = (0.0002, 0.00021, 0.00023)$.

We first use $S_{jt} = \frac{\bar{S}}{3T}$ as a warm start, and again show that we can get interpretable solutions that have less than 1% suboptimality from the best solution found with randomized starting points.

Case 1: $r = 0$:

The optimal solution is to purchase 0.37 units from the first seller, 1.21 from the second, and 8.4 from the third seller in each time period. This is a locally optimal nominal solution.

Case 2: $r = 0.05$:

In this case, the optimal solution is to purchase 10 units from the third seller in each time period. The robust parameter is still small enough that price uncertainty still does not dominate the cost, so the purchase is still spread out over the entire time horizon. However, introducing even a small amount of uncertainty is enough to move the purchased quantities away from the first two sellers, which have higher worst-case costs, and completely to the third seller. In general, the differences in quantities purchased between sellers are magnified in the presence of uncertainty, similar to how we saw that the quantities purchased are pushed to their extreme values across time in Section 3.3.3.

Case 3: $r = 2$:

As with the single-seller case, we now see the effect of price uncertainty dominating the solution when our robust parameter is set high enough. The model is now mainly concerned with the cumulative effects of price drift over time, and the optimal solution is now to complete the purchase as quickly as possible, i.e., purchasing 10 units from all three sellers for the first 10 periods. The worst-case cost savings of doing this now dominate any cost increase from purchasing from all three sellers evenly.

In general, for any set of parameters, we observed that either the optimal purchasing quantities were constant over time, or tended to be clumped into consecutive time periods, although these were not necessarily at the very start or the end of the decision horizon. As before, this shift occurred abruptly as the robust parameter r increased, although at different levels of r that depended unpredictably on the other parameters. For our problem, the level of discounts was comparatively low, and so the redistribution across sellers was much more sensitive to the initial change in the robust parameter, compared to the clumping across time.

We next evaluate the quality of the solutions found by presenting computational results solved over a range of 3150 different parameter configurations, which are listed in Appendix C.

To evaluate the models' performance, we test the solutions against three different levels of noise, each time exposing each model to 100 instances of randomly generated demand at that noise level.

Robust parameter	Mean cost	Maximum cost	SD of actual costs
0	1446.0	1682.7	123.5
0.5	1452.2	1679.8	123.8
1	1452.2	1676.1	123.7
1.5	1452.2	1680.0	123.8
2	1452.3	1680.3	123.8
2.5	1452.2	1679.0	123.8
3	1452.2	1679.7	123.7

Table 3.1: Performance of optimization model strategy for $\sigma_P = 0.01$.

Robust parameter	Mean cost	Maximum cost	SD of actual costs
0	1446.0	2012.1	154.6
0.5	1452.4	1928.6	137.5
1	1452.4	1890.8	136.1
1.5	1452.1	1881.3	139.3
2	1452.4	1882.5	139.8
2.5	1452.2	1903.0	137.2
3	1452.3	1873.6	137.0

Table 3.2: Performance of optimization model strategy for $\sigma_P = 0.1$.

As before, we are looking at the effect of robustifying the model, in terms of the mean cost, the worst-case (maximum) cost, and the variance. For our evaluation, we give the performance of the model for seven different settings of the robust parameter, but each time averaging over a range of values for all the other experimental

Robust parameter	Mean cost	Maximum cost	SD of actual costs
0	1446.0	2417.4	223.5
0.5	1452.6	2245.1	172.5
1	1452.5	2156.5	168.3
1.5	1452.0	2104.9	178.0
2	1452.5	2121.1	179.9
2.5	1452.1	2184.0	171.3
3	1452.3	2089.6	171.4

Table 3.3: Performance of optimization model strategy for $\sigma_P = 0.2$.

parameters, which were assumed to be accurately chosen when exposing the solutions to noise. The robust parameter being set to zero corresponds to the nominal formulation.

For all the cases, we observe that the robust model only has a mean cost that is higher by a fraction of a percent than the nominal solution. However, the worst-case cost observed improves significantly as we increase the protection against uncertainty to $r = 0.5$ and $r = 1$, and the variance in the actual observed cost over our 100 instances also decreases significantly.

The first case, Table 3.1, represents the scenario where the true noise is much more attenuated than our model was protecting against. In this case, the improvement to the actual worst-case cost over that for the nominal solution is minimal, but the increase in average cost is also very small. Although robustifying the model is not as necessary here, we do not lose much by doing so.

The second and third cases represent the respective scenarios where the true noise is close to what we have estimated (Table 3.2), or higher than what we have estimated (Table 3.3). In these cases, we observe that the maximum cost decreases by slightly under 10% in both cases as we increase the robustness of the model, but with only a miniscule increase in the average cost. We also observe a significant decrease in the standard deviation of the costs - a reduction of over 10% for $\sigma_P = 0.1$ and of over 25% for $\sigma_P = 0.2$. This shows that for these cases, the tradeoff is very much in favor of robustifying the problem. We gain a lot of protection against the price uncertainty, and only pay a small price to do so.

3.6 Decision Trees for Insights from the Optimal Solution

In the previous section, we saw that although the robust optimization model can potentially reduce cost variance with a very small increase in mean cost, the model does not yet solve quickly enough, at least with multiple sellers, that it can be used to solve the online purchasing execution problem. In other words, a seller would not be able to use it with real-time data by directly solving the model.

We use a modern machine learning approach to address this shortcoming in two different ways. We also observed in the previous section that generally we can find three qualitatively different interpretable solutions, corresponding to various parameter settings for the model, that are close to the best solution we can find with random starts. Although it is not known where the theoretical boundaries that divide the regions of the parameter space are, in Section 3.3.3 we were able to motivate them in terms of the dominant contributor to the uncertainty in the worst-case cost.

This suggests that a decision tree approach might be able to generate high-quality predictors that take as input data attributes and model parameters, and output some useful predictions about the model. In this section, we use the Optimal Decision Tree methods from [34] and [36] to explore some of these analytical properties of the different types of solutions that our model outputs. In Section 3.7, we show that the trees we construct here can be used to generate high-quality solutions for the purchase execution problem on the fly.

Besides being able to generate solutions in milliseconds, the decision tree approach has the additional benefit of being highly interpretable. This mirrors real decision-making and allows a practitioner to communicate the insights and output more easily to non-experts. The Optimal Decision Tree approach, in particular, avoids the difficulties of the "pathology of decision tree induction" [121] that arise from greedy approaches to constructing decision trees.

We begin by constructing a decision tree that predicts the objective function of the model. A planner might use this decision tree in several ways - for instance, to examine the impact of a different set of parameters on the purchase cost, or to have some idea how stable the current parameters are (before considering a move to another branch).

To do this, we build an Optimal Regression Tree. Following [36], we use a mixed-integer optimization approach to formulate the structure of the decision tree as MIO constraints. To decide on the minimum bucket size and depth, we used a grid search on a validation data set to tune the tree model. Figure 3-5 is the beginning of the decision tree that we generated for predicting cost (the full tree is given over Figures 3-5 to 3-13).

For all of our models and parameters, we were able to find the optimal decision trees for our training data within about 5 minutes each. To test our decision trees, we randomly split 3150 cases of generated data into 50% training set, 25% validation set, and 25% test set. The parameters we used are given in Appendix D. We used the validation set to tune parameters for the bucket size and depth of the tree, resulting in a minimum bucket size of 5 and depth of 5.

The first decision tree had an R-squared value of 0.9958 on the test data, implying that once the decision tree has been generated, the planner can accurately predict the cost of executing a purchase with new parameters, even for large problems, without having to re-solve the optimization models.

We also built Optimal Regression Trees for predicting two features of the interpretable solutions that we saw previously. These were the clustering observed in the increased proportion purchased from a supplier, and in the increased proportion purchased in a few time periods of the decision horizon.

Two representative examples are given here. Figure 3-14 is the beginning of a decision tree that predicts the optimal proportion of the total quantity to purchase in the first 10 days, i.e., 1/3 of the time horizon (the full tree is given over Figures 3-14 to 3-19. Figure 3-20 is the beginning of a decision tree that predicts the optimal proportion of the total quantity to purchase from the first supplier (the full tree is given over Figures 3-20 to 3-27). We can use these decision trees to understand the form of the optimal solution for a new set of parameters, without having to solve the

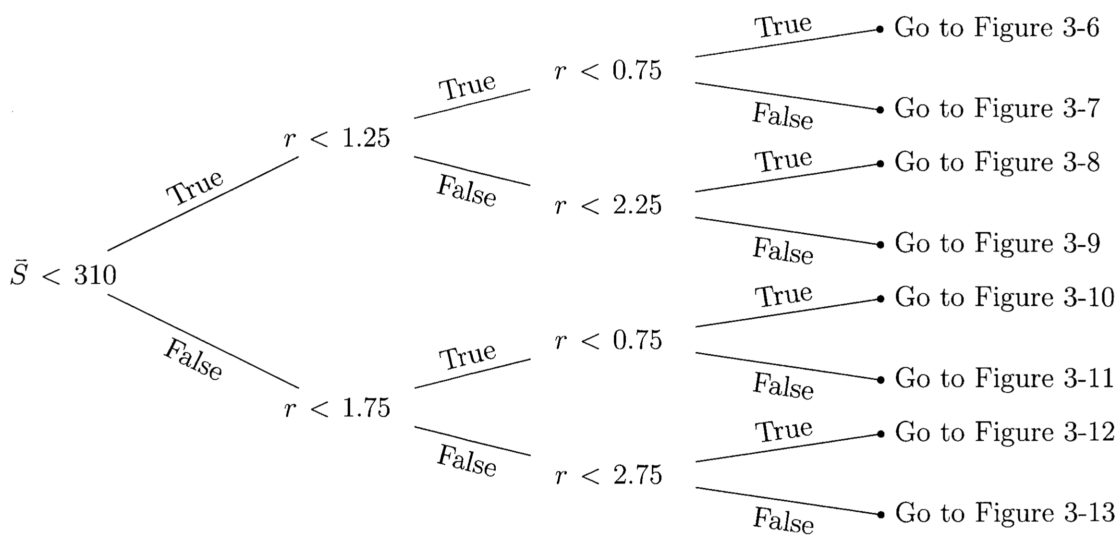

Figure 3-5: Beginning of Optimal Tree for predicting cost.

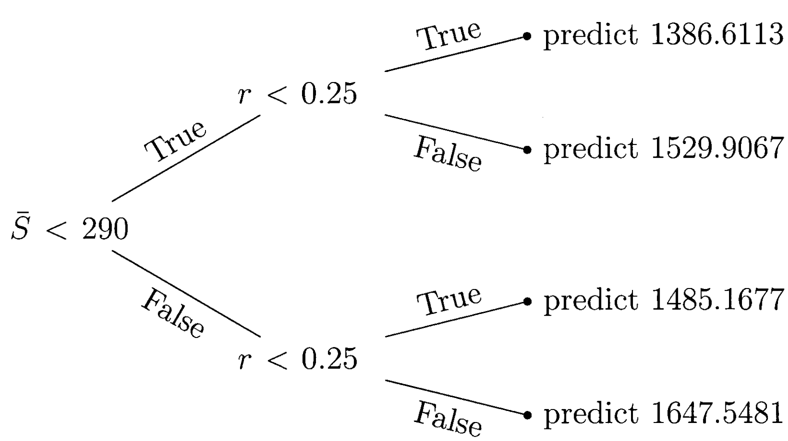

Figure 3-6: Branch of Optimal Tree for predicting cost.

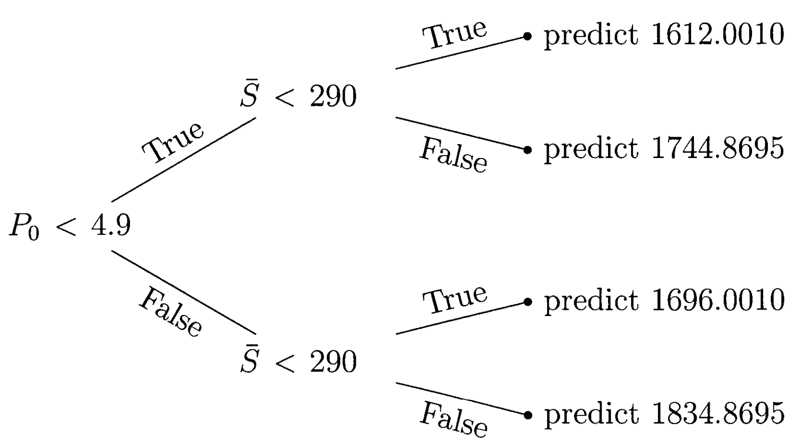

Figure 3-7: Branch of Optimal Tree for predicting cost.

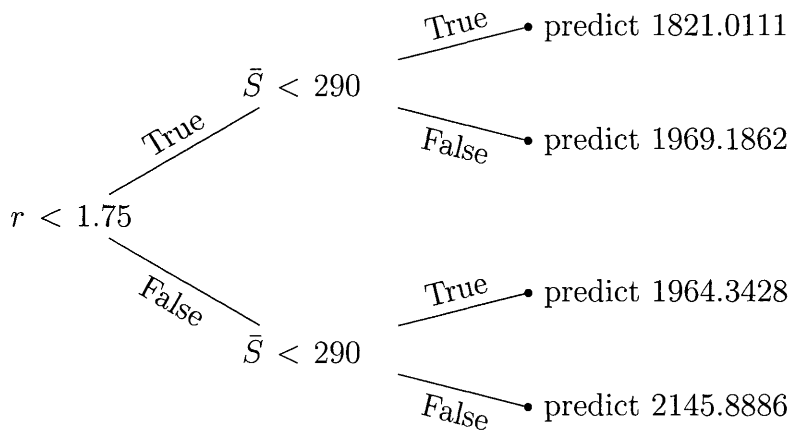

Figure 3-8: Branch of Optimal Tree for predicting cost.

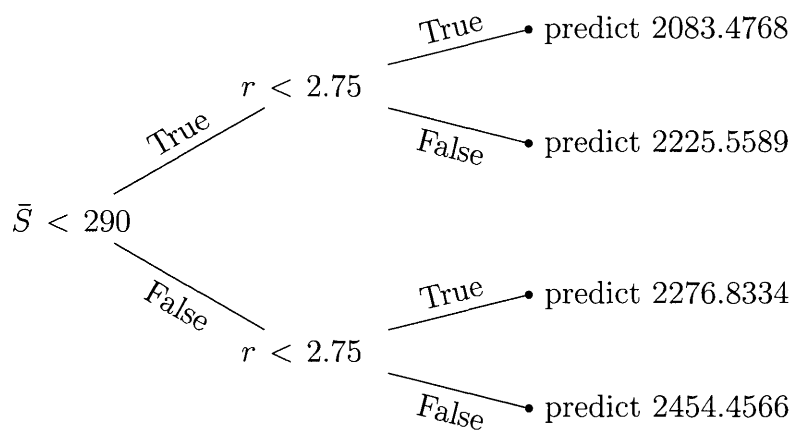

Figure 3-9: Branch of Optimal Tree for predicting cost.

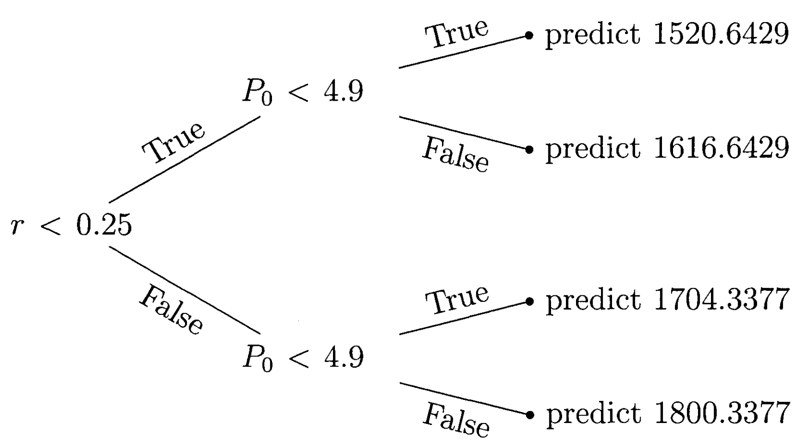

Figure 3-10: Branch of Optimal Tree for predicting cost.

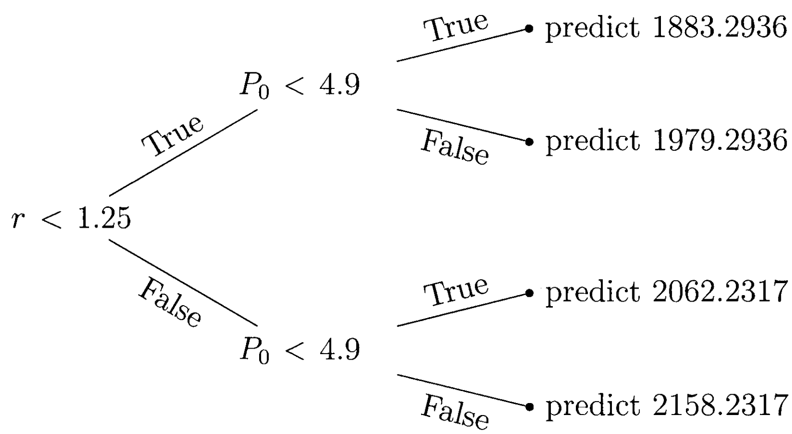

Figure 3-11: Branch of Optimal Tree for predicting cost.

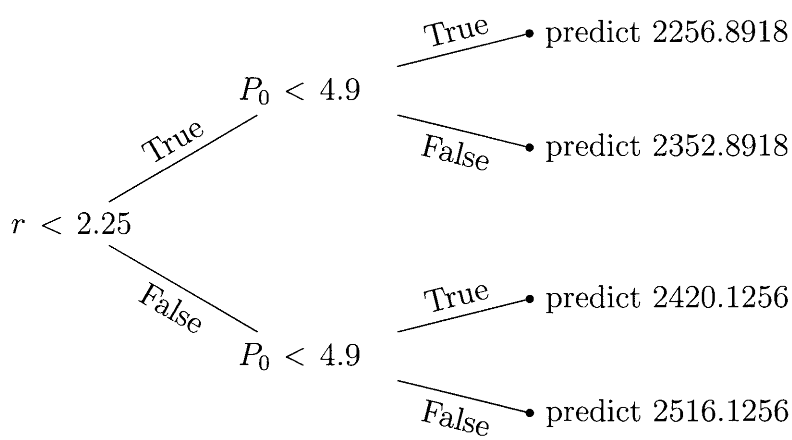

Figure 3-12: Branch of Optimal Tree for predicting cost.

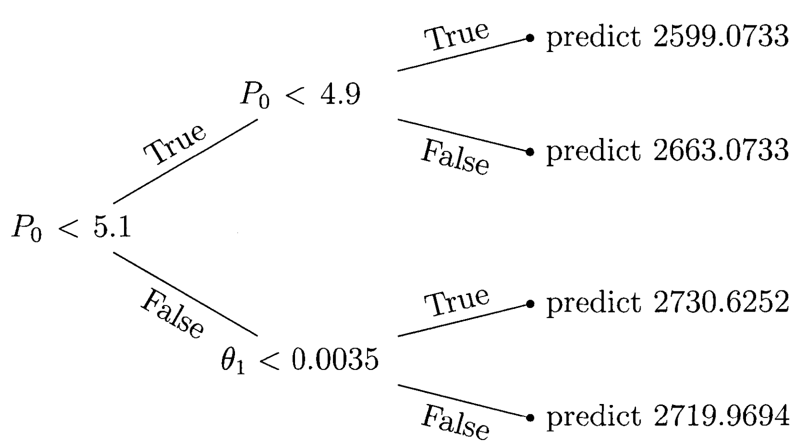

Figure 3-13: Branch of Optimal Tree for predicting cost.

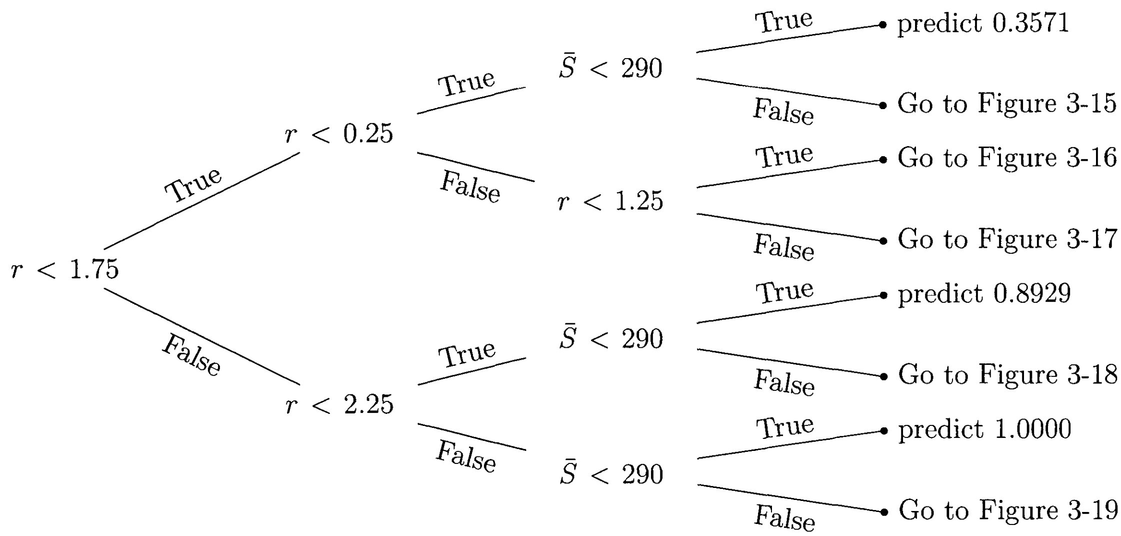

Figure 3-14: Beginning of Optimal Tree for predicting optimal proportion to purchase in first 10 days.

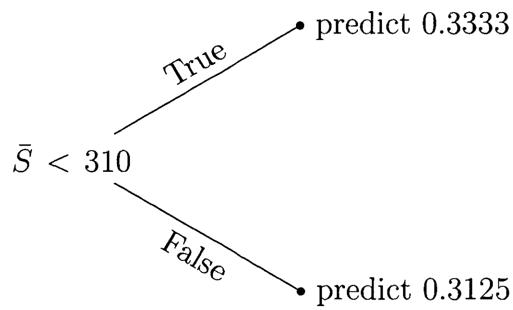

Figure 3-15: Branch of Optimal Tree for predicting optimal proportion to purchase in first 10 days.

optimization model again from scratch.

Trees with both these forms also had a R-squared value of over 0.99 on the test data, suggesting that as before, the planner would get very accurate results on a previously unseen set of parameters.

From just these three examples of our decision trees, we can glean some interesting and useful insights, to give a sense of how they could potentially be useful to a planner:

1. The decision trees confirm our observation that the solutions of the model have an abrupt transition between the nominal and robust solutions that we derived in Section 3.3.3. From Figure 3-14 and Figure 3-20, we can see that for $r < 0.25$ (low values of the robust parameter), the decision trees predict that about a

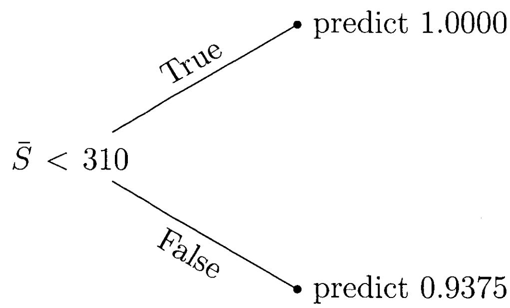

Figure 3-16: Branch of Optimal Tree for predicting optimal proportion to purchase in first 10 days.

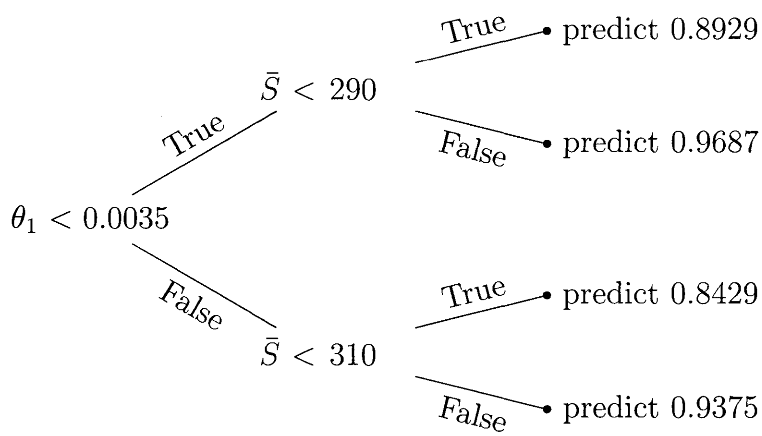

Figure 3-17: Branch of Optimal Tree for predicting optimal proportion to purchase in first 10 days.

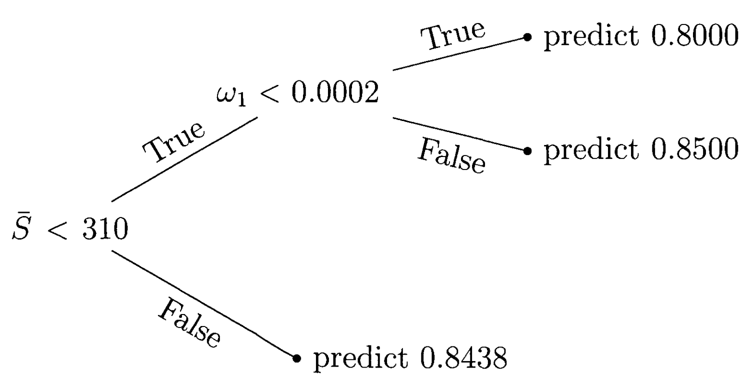

Figure 3-18: Branch of Optimal Tree for predicting optimal proportion to purchase in first 10 days.

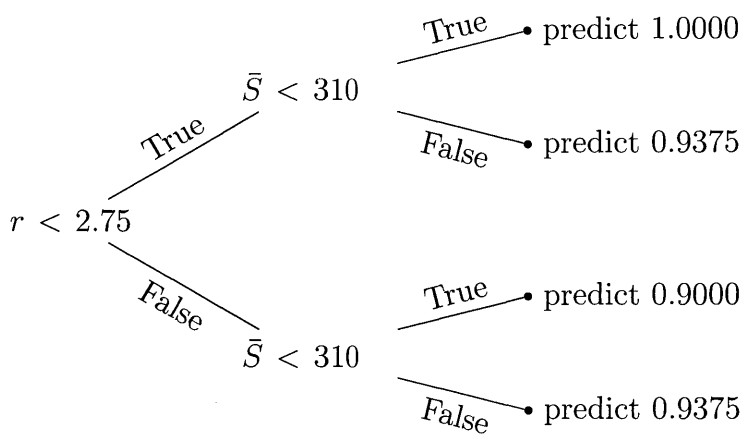

Figure 3-19: Branch of Optimal Tree for predicting optimal proportion to purchase in first 10 days.

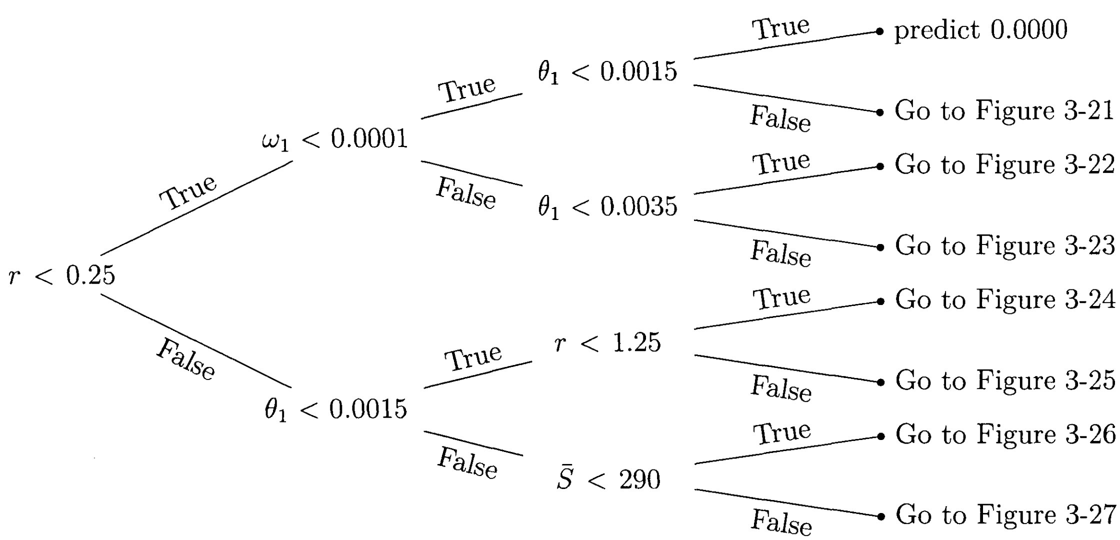

Figure 3-20: Beginning of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

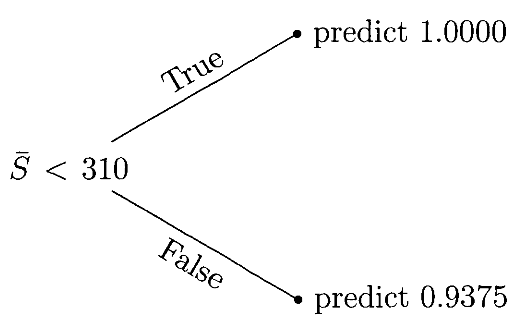

Figure 3-21: Branch of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

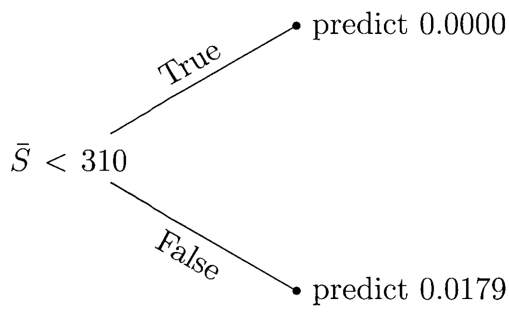

Figure 3-22: Branch of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

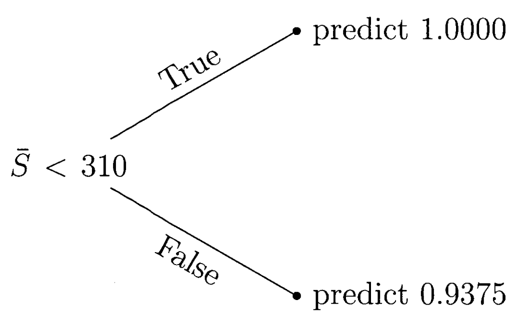

Figure 3-23: Branch of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

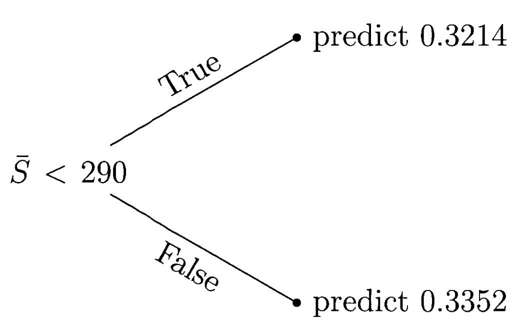

Figure 3-24: Branch of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

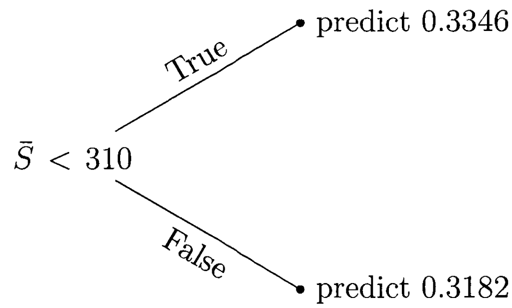

Figure 3-25: Branch of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

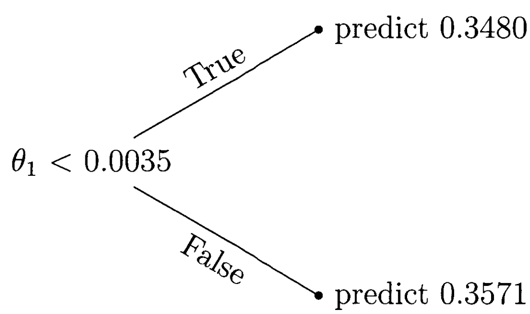

Figure 3-26: Branch of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

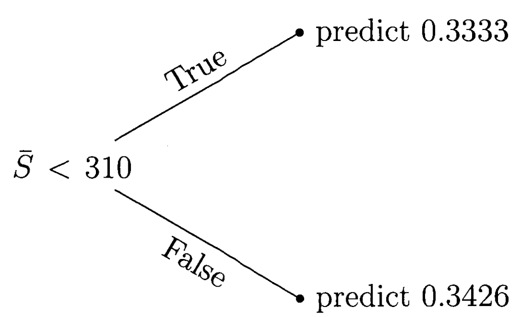

Figure 3-27: Branch of Optimal Tree for predicting optimal proportion to purchase from Seller 1.

third of the total quantity should be purchased in the first 10 days, suggesting that the optimal solution here is close to the nominal solution, i.e., purchasing a constant quantity over time.

2. Figure 3-20 additionally shows how the quantity purchased from Seller 1 varies with the magnitude of the discount; for $\theta_1 < 0.0015$, we purchase entirely from the other two sellers, while for $\theta_1 > 0.0015$, we purchase almost entirely from Seller 1. The large differences between predictions for some of the neighboring branches (e.g., Figure 3-21 and Figure 3-22) suggest that this is an important predictor to consider, as changing the discounts may lead to an abrupt change in the optimal solution.

3. Knowing the form of the solutions, we can use these decision trees to make high-quality guesses for a new set of parameters, without having to re-solve the model from scratch. For example, suppose we want to find the nominal solution for $r < 0.25$. Figure 3-14 suggests that we should predict a solution that is roughly constant in each time period, and Figure 3-20 (and corresponding decision trees for Seller 2 and Seller 3, if necessary) help us choose the proportion to purchase from each seller. We can also use Figure 3-5 to predict the cost at our new set of parameters.

4. The early splitting on the discount parameters in Figure 3-20 indicate that although r is important for determining the qualitative form of the solution, the discounts are an important predictive factor for the proportion to be purchased from each seller. They are, however, much less important for predicting the distribution across time, or the final cost.

5. We note that many of the final splits in Figure 3-14 and Figure 3-20 are on $\bar{S}$ at 290 or 310 units, but all of these only lead to small differences in the final prediction. As we know that the maximum quantity we can purchase from a single seller is capped at 300 units, this suggests that in these branches, we are starting to "overflow" from Seller 1. In a real scenario, a planner observing this

phenomenon might be able to decrease the cost further by raising the upper purchase bound, at least for that seller.

3.7 Decision Trees for Online Solving

We now demonstrate one way that a suite of decision trees, of the same forms as the previous section, can be used to generate solutions for the robust purchase execution problem. If these solutions are of sufficiently comparable quality to those obtained from solving the optimization model directly, it would allow a supplier to use real-time data on decision trees that have been previously trained on different parameter settings, and make decisions without having to re-solve the optimization model on the fly.

To do this, we constructed, from our training data, a decision tree for each seller of the same form as Figure 3-20. We also constructed, for each time period, a decision tree of the same form as Figure 3-14. This gives us, for each seller j and time period t, a prediction p_j of the proportion of the total quantity purchased from seller j and a prediction q_t of the proportion of the total quantity purchased in time period t. We take the combined prediction $p_j q_t$, and scale it to sum to the total quantity that needs to be purchased, i.e., our strategy is to purchase $S_{jt}^{tree} = \frac{p_j q_t}{\sum_{j',t'} p_{j'} q_{t'}} \bar{S}$ from seller j in time period t.

Given how we have constructed this strategy from the output proportions of the decision trees, we expect this strategy to be a good approximation of the interpretable solutions to the optimization model that we previously observed. Because we found that for our problem, one of these interpretable solutions always performed well, we also expect decision trees constructed in this manner to perform similarly (on the order of 1% suboptimality).

To evaluate this strategy over our test data, we found the theoretical worst-case actualization given by Equation 3.21, and divided that cost by the corresponding worst-case objective value for the actual solution to the optimization model, for the same parameter settings. As mentioned previously, we had 3150 different parameter settings in total, reported in Appendix D, which were split into 50% training set, 25% validation set, and 25% test set.

Over all the test data, the average of this ratio was 1.0017, and the largest value of the ratio for any set of parameters was 1.0197. This result is close to the suboptimality that we reported for the interpretable solutions in Section 3.4, as we would hope, and suggests that using the suite of decision trees that we constructed yields a purchasing strategy that is highly competitive with the strategy obtained by using the optimization model, with the benefit of requiring much less computational time. Of course, this presupposes that we have previously solved the optimization models for many parameter settings in order to generate data for the decision trees.

Finally, we perform the same tests for the decision tree strategy for different levels of noise as we did with the optimization model. At each level of the robust parameter, we expose the strategies generated for each of the 3150 parameter configurations in Appendix C to 100 instances of randomly generated demand at three different noise levels. This will show us if the decision tree strategies continue to perform competitively with the optimization model strategies when we estimate the noise level incorrectly.

Robust parameter	Mean cost	Maximum cost	SD of actual costs
0	1446.5	1681.5	123.0
0.5	1452.9	1679.5	124.1
1	1453.0	1676.3	123.8
1.5	1453.0	1680.3	124.0
2	1453.0	1680.5	124.0
2.5	1452.4	1679.1	123.7
3	1452.4	1679.8	123.7

Table 3.4: Performance of decision tree strategy for $\sigma_P = 0.01$.

As before, we are looking at the effect of robustifying the model, in terms of the mean cost, the worst-case (maximum) cost, and the variance, where we report the performance of the decision tree strategy for seven different settings of the robust parameter, but each time averaging over a range of values for all the other experimental parameters. The robust parameter being set to zero corresponds to the nominal

Robust parameter	Mean cost	Maximum cost	SD of actual costs
0	1446.5	2008.7	154.1
0.5	1453.0	1924.8	137.7
1	1453.1	1887.0	136.3
1.5	1452.8	1883.4	139.6
2	1453.2	1882.2	139.9
2.5	1452.3	1884.8	137.2
3	1452.5	1876.6	137.0

Table 3.5: Performance of decision tree strategy for $\sigma_P = 0.1$.

Robust parameter	Mean cost	Maximum cost	SD of actual costs
0	1446.4	2412.3	223.0
0.5	1453.2	2237.2	172.4
1	1453.3	2148.9	168.7
1.5	1452.7	2109.1	178.4
2	1453.3	2119.2	179.6
2.5	1452.2	2156.2	171.3
3	1452.5	2095.5	171.4

Table 3.6: Performance of decision tree strategy for $\sigma_P = 0.2$.

formulation.

The first case, Table 3.4, represents the scenario where the true noise is much more attenuated than our model was protecting against, while the second and third cases represent the respective scenarios where the true noise is close to what we have estimated (Table 3.5), or higher than what we have estimated (Table 3.6).

Compared to the earlier results in Tables 3.1-3.3 from the robust optimization model, the decision tree strategy has almost the same mean cost - a fraction of a percent higher, and is also close to the worst-case cost and variance observed. The decision tree strategy even has a lower worst-case cost and variance for the last case where the true noise is higher than our estimate, though the results are not conclusive enough for us to generalize from this with certainty.

However, we can conclude that at least for the range of parameters tested, the suite of decision trees yields a strategy that performs competitively with the robust optimization model, and it continues to perform similarly well even when the magnitude of the noise in the price has been mis-estimated.

3.8 Discussion

We have shown that our robust optimization model generates high-quality solutions that are much less affected by uncertainty. For our model, we have been able to significantly decrease the cost variance and worst-case cost, with only a small accompanying tradeoff in average cost. We were able to do this with tractable formulations that can be solved in reasonable times for problems of similar scale to those seen in real life.

We have also shown that we can use Optimal Regression Trees to predict with high accuracy some of the analytical properties of the solutions to our robust model, which enables planners to glean actionable rules underlying the optimal purchasing strategy. Over a given space of parameters, we were able to gain insights that suggested general rules underlying the effect of various parameter settings on when and how the planner should purchase the commodity. We were additionally able to use the decision trees to generate purchasing strategies without having to solve the optimization model, and showed that these purchasing strategies performed almost as well as the strategies that were actually given by the optimization model.

One interesting direction for future work would be to use the trees that we generated here to solve the initial problem more quickly, and to consider more sophisticated rules with the Optimal Trees. It would also be of interest to be able to relate the solutions corresponding to different parameter settings, as this would allow us to use the robust model presented here as a component of a larger operational formulation, of which purchasing is only one of the actions that must be planned.

Chapter 4

Integration of Robust Purchase Execution and Robust Inventory Routing

4.1 Introduction

We consider the problem of a planner who wishes to integrate two operational parts of a business, namely the purchasing of raw commodities with uncertain prices, and the fulfilment of uncertain customer demand over a fixed horizon. We suppose here that the planner is responsible for resupplying the customers to maintain their stocks of a single commodity, and has to decide on feasible schedules to achieve this.

While the planner would like to reduce the cost of resupplying each customer, their task is complicated by the high cost of stockouts and the presence of uncertainty in the demand of each customer. There is also a fixed cost involved in maintaining each vehicle, and a variable cost depending on the routes that are selected.

The planner also has to decide on a purchasing strategy that decides between a number of sources, to be carried over the same temporal horizon. Here, we consider a planner whose objective is to minimize the worst-case cost, but has to consider both the uncertainty in the quantity of the commodity that will be required by the customers, and the uncertainty in the price of the commodity at which the planner executes the purchase.

With the improvements in mixed-integer optimization solvers over the last few decades, coupled with continuing speedups in computer hardware, models of increasing size that were once intractable are now solved daily. But at real-world scales, practitioners still need to formulate their models carefully, and apply appropriate solution techniques and heuristics to maintain the tractability of the problems. In our work here, we consider one such scenario, where the planner has access to models for subproblems that might be individually tractable for the appropriate problem size and time limit, but where solving the combined problem would be difficult.

In approaching integrated problems, we ought not to ignore that companies might already have ways to solve the individual problems. Our approach allows the reuse of these methods, and is straightforward to improve when there are changes made to the formulations of the underlying subproblems, or to the algorithms used to solve them.

As with the previous chapters, we will consider the specific case of a planner in the heating oil industry. Over the course of a winter, say, the planner would like to have a plan for purchasing heating oil to meet customers' demand, and to have a schedule for refueling the customers with a given fleet of vehicles. While the planner could just solve the two subproblems separately using the work of the previous chapters, here we ask if there is any improvement from being able to generate solutions for both problems simultaneously.

Our contributions in this chapter are summarized as follows: We formulate the integrated problem of robust inventory routing and robust purchase execution. Using the models and solution techniques developed in previous chapters, we are able to use Lagrangean decomposition to tractably obtain high-quality bounds on the optimal solution of the integrated problem, and show experimentally that this lets the planner decrease both the cost and cost variance of the decisions. The decrease in mean cost over solving the problems separately is about 10% for smaller problem sizes.

The remainder of this chapter is structured as follows: in Section 4.2, we survey some of the related literature. In Section 4.3, we formulate the integrated problem, and in Section 4.4 we explain how to use Lagrangean decomposition to get high-quality solutions to the integrated problem. We detail our experiments and computational results in Section 4.5. We finally conclude with overall discussion in Section 4.6.

4.2 Related Work

4.2.1 Vendor-Managed Inventory Routing (VMI)

Because our problem does not consider real-time updating of the customers' demand, this problem context differs from the field of VMI. However, we note that robust optimization has not yet been used in the literature as a framework for VMI formulations. In fact, much of the VMI literature only discusses deterministic demand ([68], [69], [150], [151]). [1] and [66] use a constant rate of demand, and finally [55] models a problem that can be solved for Poisson demand.

4.2.2 Lagrangean Decomposition

Mixed-integer optimization models are often comprised of "easy" constraints and "hard" constraints, which make the problem computationally intractable. One well-studied approach to solving such problems is Lagrangean relaxation, wherein the "hard" constraints are replaced with penalty terms in the objective function. Typically, the relaxed optimization problem is now tractable, yielding a bound to the original problem which can be used for branch-and-bound algorithms or to aid other heuristics. (See, for example, [82], and some applications are given in [83].)

Lagrangean decomposition is a generalization of Lagrangean relaxation which is particularly useful where no such division into "easy" and "hard" can be obviously made, or when the original model comprises two or more sets of structured constraints. In this approach, identical copies of the variables that appear in both these constraint sets are made, and the condition that they should be identical is dualized. This decomposes the model into separate problems for each constraint set, and these can be solved individually to reduce computational intractability, yielding bounds that are often substantially better than those derived from other Lagrangrean relaxations [94].

4.2.3 Robust Inventory Routing

In Chapter 2, we considered the problem of robust inventory routing. In particular, the relevant literature was summarized and discussed in Section 2.2, and in Sections 2.3.2 and 2.3.3, we developed the models and solution algorithms that will be used for the relevant subproblem in this chapter.

4.2.4 Purchase Execution under Uncertainty

In Chapter 3, we considered the problem of robust purchase execution. In particular, the relevant literature was summarized and discussed in Section 3.2, and in Section 3.4, we developed the models and solution techniques that will be used for the relevant subproblem in this chapter.

4.3 Formulation of Integrated Problem

We assume that as discussed in 3.3.1, we can construct an uncertainty set $\mathcal{U}_\epsilon$ capturing the fluctuations in the market price. We also use the temperature-dependent model of uncertainty $\mathcal{U}_d$ that was previously defined in (2.12).

We aim to minimize the worst-case cost of purchasing sufficient quantities of the commodity and distributing it to customers, protecting against stockouts for the prices and demand in their respective uncertainty sets. When calculating our constraints, this is the order of daily operations: first we consider purchasing quantities, then we consider resupplying customers, and finally we consider the depletion of the customers' demand. Our formulation for the robust integrated purchasing execution and inventory routing problem is:

$$\min_{Q_{jt},S_{jt},\mathbf{u},\mathbf{v},\mathbf{g}} \max_{\boldsymbol{\epsilon_t}\in\mathcal{U}_\epsilon} \quad \sum_{t=1}^{T}\sum_{j=1}^{J} Q_{jt}S_{jt} + \sum_{t=1}^{T}\sum_{\theta=1}^{\Theta} c_\theta v_\theta^t \tag{4.1}$$

$$\text{s.t.} \quad 0 \le C_0 + \sum_{j=1}^{J}\sum_{k=1}^{t} S_{jk} - \sum_{i=1}^{N}\sum_{k=1}^{t-1} u_i^k, \;\; \forall t \in [T], \tag{4.2}$$

$$C_0 + \sum_{j=1}^{J}\sum_{k=1}^{t} S_{jk} - \sum_{i=1}^{N}\sum_{k=1}^{t-1} u_i^k \le C_{depot}, \;\; \forall t \in [T], \tag{4.3}$$

$$Q_{jt} = P_t - \theta_{jt}S_{jt} - \omega_{jt}\sum_{k=1}^{t-1} S_{kt}, \;\; \forall j \in [J], \;\; \forall t \in [T], \tag{4.4}$$

$$P_t = P_{t-1} + \epsilon_t, \;\; \forall t \in [T], \tag{4.5}$$

$$0 \le S_{jt} \le \hat{S}, \;\; \forall j \in [J], \;\; \forall t \in [T], \tag{4.6}$$

$$0 \le Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t} d_i^\tau, \;\; \forall i \in [N], \;\; \forall t \in [T], \;\; \forall \boldsymbol{d} \in \mathcal{U}_d, \tag{4.7}$$

$$Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t-1} d_i^\tau \le Q_i, \;\; \forall i \in [N], \;\; \forall t \in [T], \;\; \forall \boldsymbol{d} \in \mathcal{U}_d, \tag{4.8}$$

$$\sum_{\theta=1}^{\Theta} v_\theta^t \le M, \;\; \forall t \in [T], \tag{4.9}$$

$$u_i^t \le \sum_{\theta=1}^{\Theta} g_{i,\theta}^t, \;\; \forall i \in [N], \;\; \forall t \in [T], \tag{4.10}$$

$$\sum_{i=1}^{N} g_{i,\theta}^t \le S v_\theta^t, \;\; \forall \theta \in [\Theta], \;\; \forall t \in [T], \tag{4.11}$$

$$g_{i,\theta}^t = 0, \;\; \forall i \in [N], \;\; \forall \theta : i \notin \theta, \;\; \forall t \in [T], \tag{4.12}$$

$$g_{i,\theta}^t \ge 0, \;\; \forall i \in [N], \;\; \forall \theta \in [\Theta], \;\; \forall t \in [T],$$

$$u_i^t \ge 0, \;\; \forall i \in [N], \;\; \forall t \in [T],$$

$$v_\theta^t \in \{0, 1\}, \;\; \forall \theta \in [\Theta], \;\; \forall t \in [T].$$

(4.1) expresses the worst-case cost minimization objective. (4.2) and (4.3) enforce capacity constraints on the depot after purchases and deliveries each time period. (4.4) and (4.5) express the price dynamics of fuel. (4.6) restricts the quantity of fuel we can purchase from a single seller in any time period. (4.7) guarantees that each customer is resupplied so that their supply of the commodity is never depleted, and (4.8) enforces their capacity constraints. (4.9) respects the fleet size. (4.10) ensures that the amount of fuel assigned to refuel a customer is also assigned to some route in the same time period. (4.11) both allows us to assign fuel to a route only if the route is actually selected, and if so, also enforces vehicle capacity limits. (4.12) ensures that assignments are only made for customers that are on a given route.

4.4 Lagrangean Decomposition

We observe that Problem 4.1 comprises the two components of purchasing and routing/scheduling, where the only variables that the two subproblems have in common are the quantities being delivered, u_i^t. This suggests Lagrangean decomposition as an approach to attain stronger bounds than the conventional Lagrangean relaxation.

We first explain the process of Lagrangean decomposition for the more general formulation where two robust optimization subproblems are linked only through a subset of variables, which for simplicity we assume do not appear in the cost functions (as is true for our problem).

Suppose, then, that we have to solve for variables $\mathbf{v}, \mathbf{w}, \mathbf{x}$ to minimize the cost function $c_1(\mathbf{v}) + c_2(\mathbf{w})$. We have uncertain parameters $\delta \in \mathcal{U}_1$ and $\epsilon \in \mathcal{U}_2$, and for any particular instance of uncertainty, the feasiblility constraints for the two subproblems are given by $(\mathbf{v}, \mathbf{x}) \in C_1(\delta)$ and $(\mathbf{w}, \mathbf{x}) \in C_2(\epsilon)$ respectively.

Then we can write the general problem as:

$$\min_{\mathbf{v},\mathbf{w},\mathbf{x}} \quad c_1(\mathbf{v}) + c_2(\mathbf{w}) \tag{4.13}$$

$$\text{s.t.} \quad (\mathbf{v}, \mathbf{x}) \in \cap_{\delta \in \mathcal{U}_1} C_1(\delta), \tag{4.14}$$

$$(\mathbf{w}, \mathbf{x}) \in \cap_{\epsilon \in \mathcal{U}_2} C_2(\epsilon). \tag{4.15}$$

We can make "copies" of $\boldsymbol{x}$ to get the equivalent problem:

$$\min_{\mathbf{v},\mathbf{w},\mathbf{x},\mathbf{y}} \quad c_1(\mathbf{v}) + c_2(\mathbf{w}) \tag{4.16}$$

$$\text{s.t.} \quad (\mathbf{v},\mathbf{x}) \in \cap_{\delta\in\mathcal{U}_1} C_1(\delta), \tag{4.17}$$

$$(\mathbf{w},\mathbf{y}) \in \cap_{\epsilon\in\mathcal{U}_2} C_2(\epsilon), \tag{4.18}$$

$$\mathbf{x} - \mathbf{y} = \mathbf{0}. \tag{4.19}$$

We associate the dual variables $\boldsymbol{\lambda}$ with the constraints (4.19), and take the Lagrangean dual problem:

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{v},\mathbf{w},\mathbf{x},\mathbf{y}} \quad c_1(\mathbf{v}) - \boldsymbol{\lambda}'\mathbf{x} + c_2(\mathbf{w}) + \boldsymbol{\lambda}'\mathbf{y} \tag{4.20}$$

$$\text{s.t.} \quad (\mathbf{v},\mathbf{x}) \in \cap_{\delta\in\mathcal{U}_1} C_1(\delta), \tag{4.21}$$

$$(\mathbf{w},\mathbf{y}) \in \cap_{\epsilon\in\mathcal{U}_2} C_2(\epsilon). \tag{4.22}$$

Note that the inner minimization problem can be separated into the subproblems:

$$\min_{\mathbf{v},\mathbf{x}} \quad c_1(\mathbf{v}) - \boldsymbol{\lambda}'\mathbf{x} \tag{4.23}$$

$$\text{s.t.} \quad (\mathbf{v},\mathbf{x}) \in \cap_{\delta\in\mathcal{U}_1} C_1(\delta), \tag{4.24}$$

and

$$\min_{\mathbf{w},\mathbf{y}} \quad c_2(\mathbf{w}) + \boldsymbol{\lambda}'\mathbf{y} \tag{4.25}$$

$$\text{s.t.} \quad (\mathbf{w},\mathbf{y}) \in \cap_{\epsilon\in\mathcal{U}_2} C_2(\epsilon). \tag{4.26}$$

Since the purchase execution subproblem is only affected by the daily total quantity resupplied to customers, and not the quantities resupplied to each individual customer, we create auxiliary variables x_t to represent these in the purchase execution subproblem. Then to translate Problem 4.1 into this notation, we make "copies" of the variables x_t, and call them y_t in the inventory routing subproblem, adding the appropriate auxiliary constraints. Then the equivalent subproblems for the separated inner minimization problem over $\boldsymbol{\lambda}$ are:

$$
\begin{aligned}
\min_{Q_{jt},S_{jt},\mathbf{x}} \max_{\boldsymbol{\epsilon_t}\in\mathcal{U}_\epsilon} \quad & \sum_{t=1}^{T}\sum_{j=1}^{J} Q_{jt}S_{jt} - \sum_{t=1}^{T}\lambda_t x_t && (4.27)\\
\text{s.t.} \quad & 0 \le C_0 + \sum_{j=1}^{J}\sum_{k=1}^{t} S_{jk} - \sum_{k=1}^{t} x_k, \ \ \forall t \in [T], && (4.28)\\
& C_0 + \sum_{j=1}^{J}\sum_{k=1}^{t} S_{jk} - \sum_{k=1}^{t-1} x_k \le C_{depot}, \ \ \forall t \in [T], && (4.29)\\
& Q_{jt} = P_t - \theta_{jt}S_{jt} - \omega_{jt}\sum_{k=1}^{t-1} S_{kt}, \ \ \forall j \in [J], \ \ \forall t \in [T], && (4.30)\\
& P_t = P_{t-1} + \epsilon_t, \ \ \forall t \in [T], && (4.31)\\
& 0 \le S_{jt} \le \hat{S}, \ \ \forall j \in [J], \ \ \forall t \in [T], && (4.32)\\
& x_t \ge 0, \ \ \forall t \in [T],
\end{aligned}
$$

and

$$
\begin{aligned}
\min_{\mathbf{u},\mathbf{v},\mathbf{g},\mathbf{y}} \quad & \sum_{t=1}^{T}\sum_{\theta=1}^{\Theta} c_\theta v_\theta^t + \sum_{t=1}^{T} \lambda_t y_t && (4.33)\\
\text{s.t.} \quad & 0 \leq Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t} d_i^\tau, \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall \boldsymbol{d} \in \mathcal{U}, && (4.34)\\
& Z_i + \sum_{\tau=1}^{t} u_i^\tau - \sum_{\tau=1}^{t-1} d_i^\tau \leq Q_i, \ \ \forall i \in [N], \ \ \forall t \in [T], \ \ \forall \boldsymbol{d} \in \mathcal{U}, && (4.35)\\
& \sum_{\theta=1}^{\Theta} v_\theta^t \leq M, \ \ \forall t \in [T], && (4.36)\\
& u_i^t \leq \sum_{\theta=1}^{\Theta} g_{i,\theta}^t, \ \ \forall i \in [N], \ \ \forall t \in [T], && (4.37)\\
& \sum_{i=1}^{N} g_{i,\theta}^t \leq S v_\theta^t, \ \ \forall \theta \in [\Theta], \ \ \forall t \in [T], && (4.38)\\
& \sum_{i=1}^{N} u_i^t = y_t, \ \ \forall t \in [T], && (4.39)\\
& g_{i,\theta}^t = 0, \ \ \forall i \in [N], \ \ \forall \theta : i \notin \theta, \ \ \forall t \in [T], && (4.40)\\
& g_{i,\theta}^t \geq 0, \ \ \forall i \in [N], \ \ \forall \theta \in [\Theta], \ \ \forall t \in [T], \\
& u_i^t \geq 0, \ \ \forall i \in [N], \ \ \forall t \in [T], \\
& v_\theta^t \in \{0,1\}, \ \ \forall \theta \in [\Theta], \ \ \forall t \in [T],
\end{aligned}
$$

which we recognize as being analogs to the problems that we solved earlier, namely, Problem 2.1 (robust inventory routing formulation) and Problem 3.15 (robust multi-seller multi-period purchase formulation), but with the dual variables now considered in the cost.

We present a result, found in [94], which leads to a primal interpretation of Lagrangean decomposition, namely, as optimizing the primal objective function over the intersection of the convex hulls of the decomposed constraint sets.

Consider the following problem:

$$\max_{x} \quad fx \tag{4.41}$$

$$\text{s.t.} \quad Ax \leq b, \tag{4.42}$$

$$Cx \leq d, \tag{4.43}$$

$$x \in X, \tag{4.44}$$

where f, b, d, A, C are vectors and matrices of appropriate dimensionality, and the feasible set X imposes other constraints and integer requirements on the problem. Let Y be any set containing X.

The Lagrangean dual from using Lagrangean decomposition is:

$$V_D = \min_u \begin{pmatrix} \max\limits_x & (f-u)x & + & \max_y & uy \\ \text{s.t.} & Cx \leq d, & & \text{s.t.} & Ay \leq b, \\ & x \in X, & & y \in Y. & \end{pmatrix}$$

Consider the problem of optimizing (4.41) over the convex hulls of the polyhedral constraint sets:

$$V_Q = \max_x \quad fx \tag{4.45}$$

$$\text{s.t.} \quad x \in Conv(Ax \leq b, x \in Y,) \tag{4.46}$$

$$x \in Conv(Cx \leq d, x \in X.) \tag{4.47}$$

Then we have:

Theorem 4. *(Corollary 3.4 in [94])*

$$V_Q = V_D.$$

We also have optimality conditions for when the subproblems do eventually yield solutions for which the copied variables are identical, for multipliers that are optimal for the Lagrangean decomposition dual:

Theorem 5. *(Lemma 4.1 in [94]) If* $\boldsymbol{\lambda}_*$ *are optimal solutions to the Lagrangean decomposition dual (4.20),* $\boldsymbol{v}, \boldsymbol{x}$ *are optimal solutions to (4.23),* $\boldsymbol{w}, \boldsymbol{y}$ *are optimal solutions to (4.25), and* $\boldsymbol{x} = \boldsymbol{y}$*, then* $\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{x}$ *are optimal solutions to (4.13).*

Proof. If $\boldsymbol{x} = \boldsymbol{y}$, then the weak optimality conditions are satisfied since we have feasibility and complementary slackness. □

4.4.1 Subgradient Algorithm

While the uncertainty in the routing subproblem affects the actual feasibility of the solutions, in that we might observe stockouts if customers are not resupplied with sufficient heating oil, the uncertainty in the price of heating oil only affects the optimality of the solution for the purchase execution subproblem. This is an important observation because it means that any stage, if the current iteration is feasible for the robust inventory routing problem, it must be feasible for the integrated problem.

We now present the algorithm which we used to solve the integrated problem. For our specific choices of step size and stopping conditions, we follow closely the application in [95]. k counts the iterations, and π_k is the parameter by which the algorithm reduces the step size μ_k. We set $\pi_{\min}$ to 0.001 and $k_{\max}$ to 1000. UB_1 and UB_2 are optimal values to the subproblems, which give us an upper bound for the problem, while LB_1^k and LB_2^k are the optimal values which yield lower bounds at the kth iteration.

4.5 Computational Experiments

We test both the scalability of our approach to the combined formulation, and the resulting reduction in costs. To examine this tradeoff, we imported customer locations from a few instances in the TSPLIB, the standard Traveling Salesman Problem test bed, with sizes of these instances ranging from 51 to 5915. These were the same

Algorithm 2: INT-LD

Input: $\pi_{\min}$, $k_{\max}$, all parameters in (4.27) and (4.33).
Output: $UB, LB, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}$
$k = 0$;
$\lambda_t = 0 \ \forall t \in [T]$;
Solve (4.33) to get $\boldsymbol{y}^0$ and UB_1;
Solve (4.27) with $\boldsymbol{x} = \boldsymbol{y}^0$ to get UB_2;
$UB \leftarrow UB_1 + UB_2$;
while $k \leq k_{\max}$ **and** $\pi_k \geq \pi_{\min}$ **do**
  $k \leftarrow k + 1$;
  Solve (4.33) to get $\boldsymbol{y}^k$ and LB_1^k;
  Solve (4.27) to get $\boldsymbol{x}^k$ and LB_2^k;
  $LB^k \leftarrow LB_1^k + LB_2^k$;
  if $k > 1$ **and** $LB^k \leq LB^{k-1}$ **then**
    $\pi_k \leftarrow \pi_{k-1}/2$;
  end
  $\mu_k \leftarrow \pi_k(UB - LB^k) / \sum_t (y_t^k - x_t^k)^2$;
  $\lambda_t \leftarrow \lambda_t + \mu_k(y_t^k - x_t^k) \ \forall t \in [T]$;
end

instances that we used in Section 2.5, and the parameters set are described in detail there and in Appendix A.

As we did previously in Section 2.5.2, we assume here that if a customer experiences stockouts, the planner must send out an emergency refuelling vehicle which costs twice as much as the usual fleet per unit distance.

To avoid end-of-horizon effects, we only considered the cost of demand and purchasing over the first 140 days. For the purchasing subproblem, we increased the tractability of the model by grouping demand into periods of a week, which gave us exactly 20 periods. We assumed that the planner started with a quantity of $10N$ units of heating oil, with a total capacity of $30N$ units, and that demand for each week had to be purchased by the start of that week. We used the approach described in 3.4.1 to formulate these capacity and demand constraints over the time horizon, and the robust parameter r was set to 1.

We solved the model for three suppliers, assuming that the planner could purchase at most $7N$ units per time period from each seller. This was set so that no single seller would be able to satisfy all the demand, but that the problem did not require all three sellers to be used. We set $\theta = (0.001, 0.0095, 0.009)$ and $\omega = (0.0002, 0.00021, 0.00023)$ for problem sizes up to 200, but reduced them to a tenth of that for the larger problems to keep prices positive for the increased purchase quantities. We used a starting price of 5 dollars per unit.

We first report the scalability of the algorithm. We let the subgradient algorithm run for two hours for each problem size. In Lagrangean decomposition, the subproblems can be solved in parallel, and so the computational time is dominated by the subproblem which takes the longest time to solve, as the non-optimization steps of the algorithm run in comparatively negligible time. For our problem context, the purchase execution subproblem could be solved in much shorter time (typically less than 10 seconds for $T = 20$) than the inventory routing subproblem. In addition, the purchase execution subproblem does not become more computationally intractable as the number of customers increases, because it only relies on meeting aggregated demand.

On the other hand, the robust vehicle routing problem rapidly becomes too expensive to iterate over many times. To find just the fifth incumbent solution with Gurobi, for instance, takes only 147 seconds with $N = 51$, but 1.45 hours for $N = 575$, and over 4 hours for $N = 5915$. For problems with more than a hundred customers, then, the algorithm only ran for 2-5 iterations. In comparison, [95] report that a similarly-parameterized subgradient method on a graph problem terminated in the order of 50 iterations. We might thus expect that faster heuristics for the inventory routing problem might allow a planner to make a tradeoff between letting the algorithm run for more iterations, and a less accurate estimate of the routing cost.

Figures 4-1, 4-2 and 4-3 show the reduction in the mean, the maximum and the standard deviation of the observed cost over 100 instances, of which further experimental details and parameter settings are given in Appendix E.

For the problems with smaller sizes, integrating the problems gives more improvement, with 9.3% to 12.5% decreases in mean cost. We also see similarly significant decreases in worst-case costs and the variance of the observed costs. Given much more computational time and iterations, it might be expected that the larger prob-

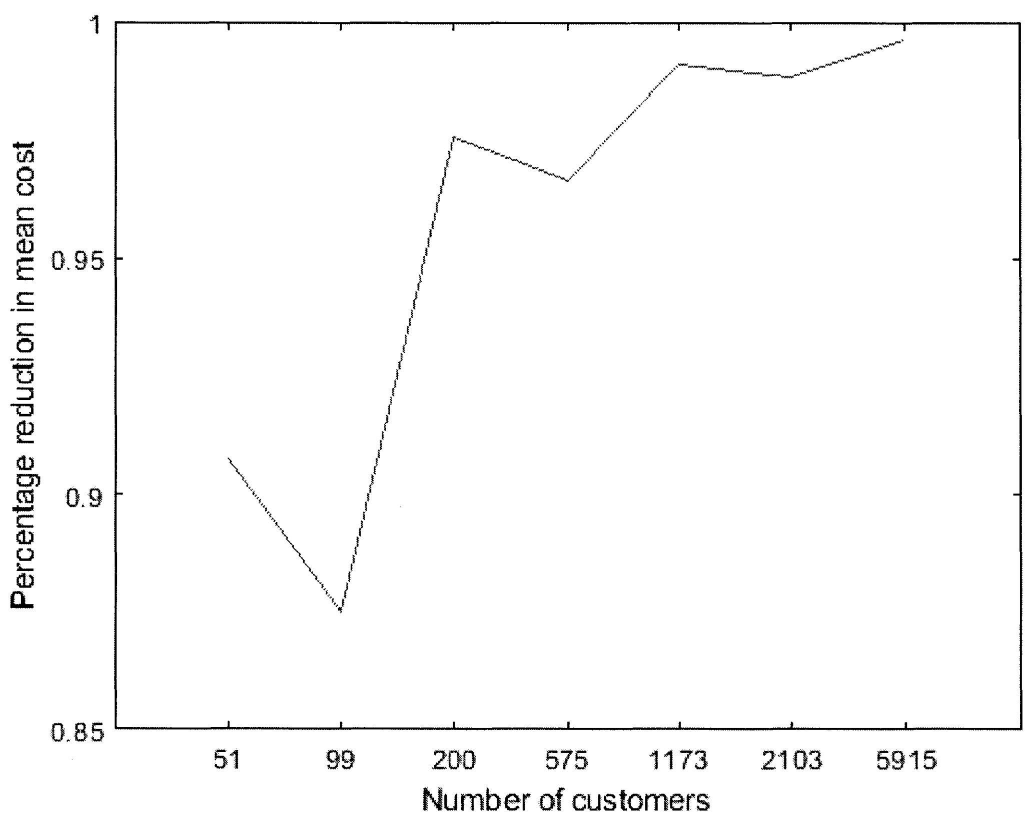

Figure 4-1: Percentage reduction in mean cost over naive purchasing strategy.

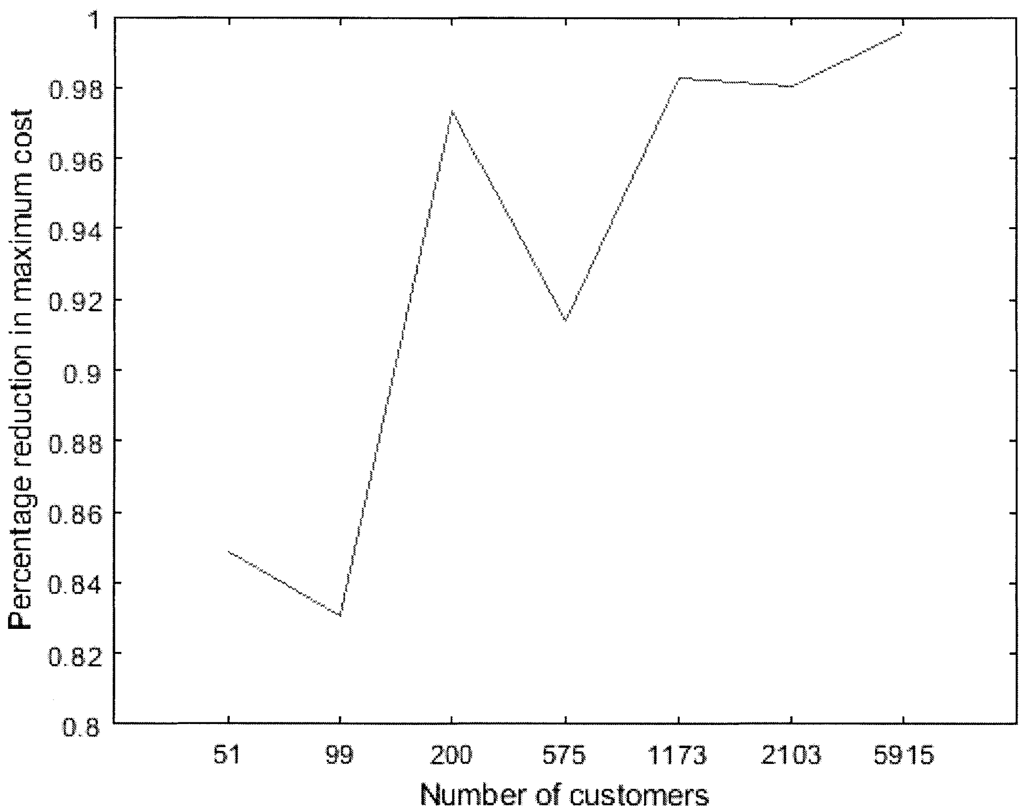

Figure 4-2: Percentage reduction in maximum cost over naive purchasing strategy.

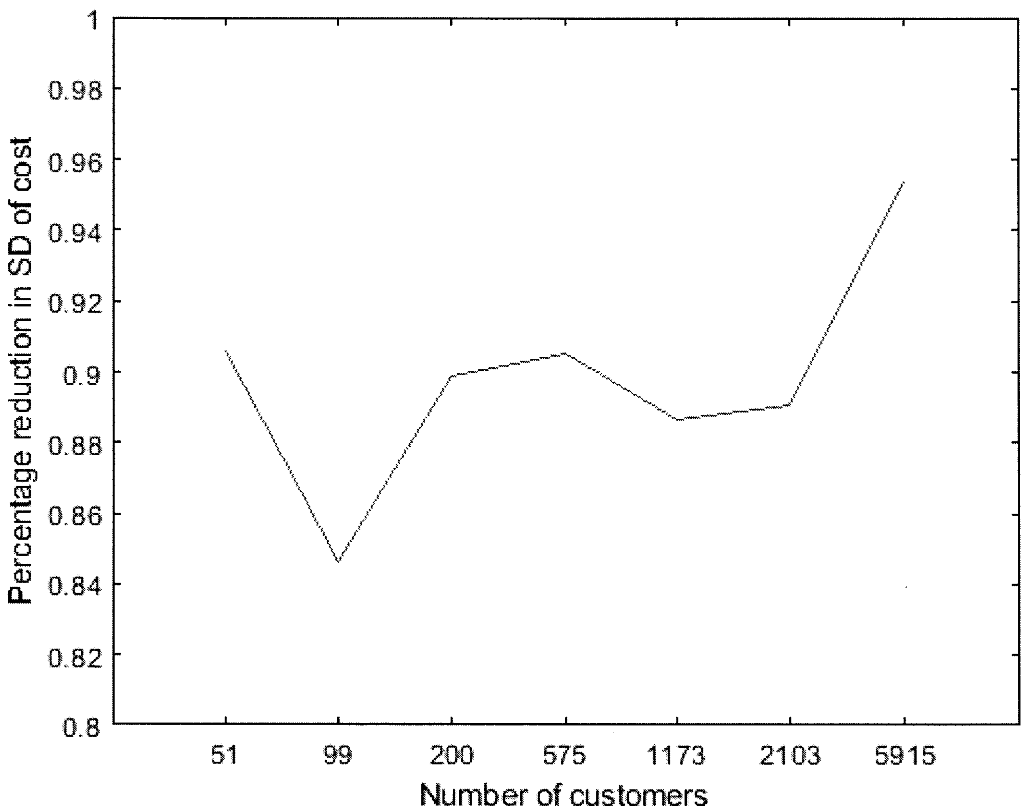

Figure 4-3: Percentage reduction in standard deviation of cost over naive purchasing strategy.

lems would be able to achieve better results, but we were still able to get small improvements in all three metrics for the larger problems even when only one or two iterations were possible.

4.6 Discussion

In this chapter, we showed that there are potential benefits from combining the two problems that we had considered previously. Although the combined problem is significantly less tractable than each of the subproblems, we were able to use Lagrangean decomposition to get high-quality feasible solutions in a reasonable time.

For problems of smaller sizes, we were able to decrease the mean cost by around 10%. While the larger problems exhibited much less of an improvement, this was mainly due to the scheduling subproblems being expensive computationally and we were still able to get small cost decreases from the integration. We also observed a decrease in the worst-case costs and cost variability for all problem sizes.

A future direction of interest would be to investigate using the previous solutions more effectively to solve the iterated subproblems during the Lagrangean decomposition. By reducing the time required to solve even larger problems, we ought to be able to achieve improvements that are comparable to the smaller problems.

Chapter 5

Conclusions and Future Directions

In this thesis, we have seen the potential significant benefits of applying modern optimization techniques to integrate the two difficult problems of purchasing and inventory routing in the heating oil industry. We used the robust optimization framework to model uncertainty in temperature, customer demand, and market prices, and developed novel techniques to address the resulting uncertainty sets tractably.

In the problem domain we have considered, the iterations for subgradient ascent in the integrated model are dominated by the inventory routing subproblem, because that scales computationally with the number of customers, which is much larger than the typical time horizon of the problem. The robust purchase subproblem does not face this issue, because there we can consider the combined demand instead.

This suggests that one fruitful approach to improving this work, particularly for larger problems, might be to develop *looser* heuristics for the inventory routing problem of Chapter 2. Although in isolation we would expect this to produce less optimal results for the subproblem, it may well increase the number of iterations that can be performed in a reasonable time. While the computational results of Section 2.5.1 suggest that this would not improve the results for the routing problem directly, it might be possible to bring the improvements seen in Section 4.5 for larger problems more in line with what we were able to attain for smaller problems.

Another possible direction to extend this work is to extend Chapters 3 and 4 to incorporate adaptive optimization techniques. This would allow the work to be used more generally for VMI problems, and allow planners to react more quickly to actual weather and market conditions as they develop over time.

Finally, we were able to use the decision tree methods in [36] to obtain high-quality strategies for the online robust purchase execution problem. This is an exciting result that could be more generally applicable to other optimization models, and we would be interested to understand more deeply what are the features of optimization problems that could render them amenable to a similar treatment.

Appendix A

Inventory Routing Simulation Data

Let the number of customers in the data set be N and let each customer have a capacity of $Q = 20$. We consider the length of the planning horizon, T, to be 151 time units. However, to account for end-of-horizon effects, in our experiments we solve the model for $T = 151$, but only calculate costs for the first 141 time units. We assume a homogenous fleet of vehicles, each with capacity $S = 200$.

1. *Estimated initial amounts*: We generate the *estimated* initial level of oil for each customer $i \in N$ using the following formula:

   $$z_i^{est} = Q \times (1 - \min(0.9, |X_i|/3)),$$

   where X_i are i.i.d. standard normal random variables sampled once for each customer. We generate z_i^{est} once for each customer for all the training scenarios, and once for each customer for all the testing scenarios.

2. *Realized initial amounts*: Once the estimated initial amounts are fixed, we generate *actual* customer levels at the start of the horizon, called z_i. These are generated with randomness proportional to the estimated amount already consumed. More precisely, for each scenario, we generate the initial customer level using:

   $$z_i = z_i^{est} + (Q - z_i^{est}) \times U_i,$$

where U_i are i.i.d. uniform random variables distributed as $U_i \sim U(-1/2, 1/2)$. We finally clip the z_i within the interval $[0.5, Q]$.

3. *Estimated Temperature*: We set the base temperature, T_{base}, to $70°F$. Estimated temperature for day $t \in [T]$ is computed as:

$$T_t^{est} = T_{base} - 5 - t * 0.2.$$

4. *Realized Temperature*: We consider different scenarios with the noise in temperature, δ_t, varying in the set $\{0.02, 0.04, 0.06, \ldots, 1.0\}$. For each value for δ_t, we create instances with temperature generated using the following relation:

$$T_t = T_t^{est} + \max(-3 * \delta_t, \min(3 * \delta_t, \delta_t * X_t)).$$

5. *Fleet Size*: For datasets with less than 1000 customers, we assume a fleet with approximately $\sqrt{N}/2$ vehicles. Otherwise, we assume approximately $3D/ST$ vehicles, where D is the total mean demand across all the customers over the entire planning period, S is the vehicle capacity and T is the planning period. For data sets of size 51, 99, 200, 575, 1173, 2103 and 5915, this means a fleet size of 3, 4, 7, 11, 24, 42 and 117 vehicles respectively. These numbers were chosen based on the total average demand and vehicle capacity.

6. *Routes*: We start with an automatically set of generated routes such that for datasets with 51, 99 and 200 customers we have 4 routes covering each customer and for the larger datasets, where clusters are more stable, we have only 1 route covering each customer. The characteristics of the routes used in our experiments are included in Table A.1, showing the number of routes (`Num Routes`), Minimum cost of the routes (`Min cost`), Maximum cost of the routes (`Max cost`), Average cost of the routes (`Avg cost`), number of routes covering each customer (`Per cust`), Minimum route size (`Min size`), Maximum route size (`Max size`) and Average route size (`Avg size`).

Size	Routes	Min cost	Max cost	Avg cost	Per cust	Min size	Max size	Avg size
51	15	113.36	234.08	153.57	4	10	20	13.60
99	27	198.1877	521.12	297.89	4	11	23	14.67
200	55	2454.26	10200.68	5216.82	4	11	24	14.54
575	18	406.06	903.31	709.18	1	27	36	31.94
1173	36	212.18	628.14	392.40	1	12	37	32.58
2103	58	196.80	764.52	422.35	1	8	46	36.25
5915	174	3585.99	40454.50	17088.69	1	16	42	33.99

Table A.1: Characteristics and coverage of the routes used in the experiments for various datasets.

Appendix B

Parameter settings for purchase execution experiments (1 seller)

These are the sets of parameters which were tested in the experiments described in Section 3.5.1.

- Upper bound for each supplier in each time period: 10 units
- Time horizon: 30 periods
- Total amount: $\{260, 280, 300, 320, 340\}$ (units)
- Starting price: $\{4.8, 4.9, 5.0, 5.1, 5.2\}$(dollars)
- Starting quantity: 10 units
- θ: $\{0.001, 0.002, 0.005\}$
- ω: $\{0.0002, 0.00021, 0.00023\}$
- Robust parameter r: $\{0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$

Appendix C

Parameter settings for purchase execution experiments (3 sellers)

These are the sets of parameters which were tested in the experiments described in Section 3.5.2.

- Upper bound for each supplier in each time period: 10 units
- Time horizon: 30 periods
- Total amount: $\{260, 280, 300, 320, 340\}$ (units)
- Starting price: $\{4.8, 4.9, 5.0, 5.1, 5.2\}$ (dollars)
- Starting quantity: 10 units
- θ: $\{0.001, 0.002, 0.005\} \times (1, 0.95, 0.9) \cup \{0.001, 0.002, 0.005\} \times (1, 0.9, 0.95)$
- ω: $(0.0002, 0.00021, 0.00023)$
- Robust parameter r: $\{0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$

Appendix D

Parameter settings used to generate optimal decision trees

These are the sets of parameters which were tested in the experiments described in Section 3.6.

- Upper bound for each supplier in each time period: 10 units
- Time horizon: 30 periods
- Total amount: $\{260, 280, 300, 320, 340\}$ (units)
- Starting price: $\{4.8, 4.9, 5.0, 5.1, 5.2\}$ (dollars)
- Starting quantity: 10 units
- θ: $\{0.001, 0.002, 0.005\} \times (1, 0.95, 0.9) \cup \{0.001, 0.002, 0.005\} \times (1, 0.9, 0.95)$
- ω: $\{0.0001, 0.0016, 0.0002\} \times (1, 1.05, 1.1)$
- Robust parameter r: $\{0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$
- Warm starts: $S_t = \frac{\bar{S}}{3T}$

Appendix E

Robust Integrated Inventory Routing and Purchase Execution Simulation Data

For the demand subproblem, we used the datasets and problem parameters that were described in Appendix A.

We let Algorithm 2 run for two hours for each problem size. Most of this time was taken up with solving the routing subproblem, particularly for the larger sizes, as each iteration requires both subproblems to be solved once.

Table E.1 shows the mean cost, maximum cost and standard deviation of the actual costs, where in each case we tested the solutions against 100 simulated instances with standard deviation of noise $\sigma_P = 0.2$, a fairly noisy scenario.

As a baseline, which we call *Naive*, we compare it to the model where at each period, we purchase exactly the quantity required for the demand that a single run of the routing subalgorithm outputs, and split the purchase equally between each of the three sellers.

Customers	Approach	Mean cost	Maximum cost	SD of actual costs
51	Naive	54072	75263	4865
51	Model	49078	63863	4407
99	Naive	137649	182008	9543
99	Model	120445	151168	8075
200	Naive	3.027e6	3.111e6	19634
200	Model	2.954e6	3.028e6	17643
575	Naive	954089	1.198e6	55688
575	Model	922169	1.095e6	50417
1173	Naive	8.850e6	9.345e6	112511
1173	Model	8.769e6	9.186e6	99720
2103	Naive	1.486e7	1.575e7	201827
2103	Model	1.469e7	1.544e7	179786
5915	Naive	1.678e8	1.703e8	568135
5915	Model	1.672e8	1.696e8	542026

Table E.1: Costs for data sets of different sizes.

Bibliography

[1] Beatriz Abdul-Jalbar, José Gutiérrez, and Joaquín Sicilia. Integer-ratio policies for distribution/inventory systems. *International Journal of Production Economics*, 93-94:407–415, 2005. Proceedings of the Twelfth International Symposium on Inventories.

[2] Daniel Adelman. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514, 2004.

[3] El-Houssaine Aghezzaf. Robust distribution planning for supplier-managed inventory agreements when demand rates and travel times are stationary. *Journal of the Operational Research Society*, 59(8):1055–1065, 2007.

[4] Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40, 2001.

[5] Patrick Amestoy, Iain Duff, Jacko Koster, and Jean-Yves L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[6] Patrick Amestoy, Abdou Guermouche, Jean-Yves L'Excellent, and Stéphane Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

[7] Claudia Archetti, Luca Bertazzi, Gilbert Laporte, and Maria Grazia Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391, 2007.

[8] Claudia Archetti, Nicola Bianchessi, Stefan Irnich, and Maria Grazia Speranza. Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21(3):353–374, 2014.

[9] Claudia Archetti and Maria Grazia Speranza. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014.

[10] Jan Arnold. *Commodity procurement with operational and financial instruments.* Springer, 2010.

[11] Jan Arnold, Stefan Minner, and Björn Eidam. Raw material procurement with fluctuating prices. *International Journal of Production Economics*, 121(2):353–364, 2009.

[12] Tirukkattuppalli Subramanyam Arthanari and Yadolah Dodge. *Mathematical programming in statistics.* New York: Wiley, 1981.

[13] Peter Auer, Robert Holte, and Wolfgang Maass. Theory and applications of agnostic PAC-learning with small decision trees. In *Machine Learning Proceedings 1995*, pages 21–29. Elsevier, 1995.

[14] Philippe Augerat. *Polyhedral study of the capacitated vehicle routing.* PhD thesis, Universite Joseph Fourrier, Grenoble, 1995.

[15] Roberto Baldacci, Nicos Christofides, and Aristide Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2008.

[16] Roberto Baldacci, Eleni Hadjiconstantinou, and Aristide Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, 52(5):723–738, 2004.

[17] Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

[18] Roberto Baldacci, Paolo Toth, and Daniele Vigo. Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175(1):213–245, 2010.

[19] Chaithanya Bandi and Dimitris Bertsimas. Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical Programming*, 134(1):23–70, 2012.

[20] Aharon Ben-Tal, Dick Den Hertog, and Jean-Philippe Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1-2):265–299, 2015.

[21] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization.* Princeton University Press, 2009.

[22] Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

[23] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[24] Kristin Bennett. *Decision tree construction via linear programming.* Center for Parallel Optimization, Computer Sciences Department, University of Wisconsin, 1992.

[25] Kristin Bennett and Jennifer Blue. Optimal decision trees. *Rensselaer Polytechnic Institute Math Report*, 214:24, 1996.

[26] Luca Bertazzi, Adamo Bosco, and Demetrio Laganà. Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega*, 56:112–121, 2015.

[27] Luca Bertazzi, Adamo Bosco, and Demetrio Laganà. Min-max exact and heuristic policies for a two-echelon supply chain with inventory and transportation procurement decisions. *Transportation Research Part E: Logistics and Transportation Review*, 93(Supplement C):57–70, 2016.

[28] Luca Bertazzi, Giuseppe Paletta, and Maria Grazia Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119–132, 2002.

[29] Luca Bertazzi, Martin Savelsbergh, and Maria Grazia Speranza. Inventory routing. In *The vehicle routing problem: latest advances and new challenges*, pages 49–72. Springer, 2008.

[30] Dimitris Bertsimas. A vehicle routing problem with stochastic demand. *Operations Research*, 40(3):574–585, 1992.

[31] Dimitris Bertsimas, David Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[32] Dimitris Bertsimas and Constantine Caramanis. Finite adaptability in multistage linear optimization. *Automatic Control, IEEE Transactions on*, 55(12):2751–2766, 2010.

[33] Dimitris Bertsimas and Frans de Ruiter. Duality in two-stage adaptive linear optimization: Faster computation and stronger bounds. *INFORMS Journal on Computing*, 28(3):500–511, 2016.

[34] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

[35] Dimitris Bertsimas and Jack Dunn. *Machine learning under a modern optimization lens.* Dynamic Ideas Press, 2018.

[36] Dimitris Bertsimas, Jack Dunn, and Nishanth Mundru. Optimal prescriptive trees. *Under review*, 2018.

[37] Dimitris Bertsimas, Vineet Goyal, and Xu Andy Sun. A geometric characterization of the power of finite adaptability in multistage stochastic and adaptive optimization. *Mathematics of Operations Research*, 36(1):24–54, 2011.

[38] Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018.

[39] Dimitris Bertsimas, Paul Hummel, and Andrew Lo. Optimal control of execution costs for portfolios. *Computing in Science & Engineering*, 1(6):40–53, 1999.

[40] Dimitris Bertsimas, Dan Iancu, and Pablo Parrilo. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

[41] Dimitris Bertsimas and Angela King. An algorithmic approach to linear regression. *Operations Research*, 64(1):2–16, 2015.

[42] Dimitris Bertsimas and Angela King. Logistic regression: From art to science. *Statistical Science*, 32(3):367–384, 2017.

[43] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

[44] Dimitris Bertsimas, Eugene Litvinov, Xu Andy Sun, Jinye Zhao, and Tongxin Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *Power Systems, IEEE Transactions on*, 28(1):52–63, 2013.

[45] Dimitris Bertsimas and Andrew Lo. Optimal control of execution costs. *Journal of Financial Markets*, 1(1):1–50, 1998.

[46] Dimitris Bertsimas and Rahul Mazumder. Least quantile regression via modern optimization. *The Annals of Statistics*, 42(6):2494–2525, 2014.

[47] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, 2003.

[48] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[49] Dimitris Bertsimas and David Simchi-Levi. A new generation of vehicle routing research: robust algorithms, addressing uncertainty. *Operations Research*, 44(2):286–304, 1996.

[50] John R Birge and Francois Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[51] Robert Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, pages 107–121, 2012.

[52] Leo Breiman. *Classification and regression trees.* Routledge, 2017.

[53] Valentina Cacchiani, Vera Hemmelmayr, and Fabien Tricoire. A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, 163:53–64, 2014.

[54] Ann Campbell, Lloyd Clarke, Anton Kleywegt, and Martin Savelsbergh. The inventory routing problem. In *Fleet management and logistics*, pages 95–113. Springer, 1998.

[55] Sila Çetinkaya and Chung-Yee Lee. Stock replenishment and shipment scheduling for vendor-managed inventory systems. *Management Science*, 46(2):217–232, 2000.

[56] Krishna Chepuri and Tito Homem-De-Mello. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research*, 134(1):153–181, 2005.

[57] Nicos Christofides and Samuel Eilon. An algorithm for the vehicle-dispatching problem. *OR*, pages 309–318, 1969.

[58] Nicos Christofides, Aristide Mingozzi, and Paolo Toth. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1):255–282, 1981.

[59] Leandro Coelho, Jean-François Cordeau, and Gilbert Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2013.

[60] Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 1997.

[61] Jean-François Cordeau, Gilbert Laporte, Martin Savelsbergh, and Daniele Vigo. Vehicle routing. *Handbooks in Operations Research and Transportation Science: Transportation*, 14:367–428, 2006.

[62] Louis Cox, Qiu Yuping, and Warren Kuehner. Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Annals of Operations Research*, 21(1):1–29, 1989.

[63] IBM ILOG CPLEX. V12.1: User's manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.

[64] Georges Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958.

[65] George Dantzig and John Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[66] Mohammad Darwish and Osama Odah. Vendor managed inventory model for single-vendor multi-retailer supply chains. *European Journal of Operational Research*, 204(3):473–484, 2010.

[67] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[68] Stephen Disney and Denis Towill. The effect of vendor managed inventory (VMI) dynamics on the bullwhip effect in supply chains. *International Journal of Production Economics*, 85(2):199–215, 2003.

[69] Yan Dong and Kefeng Xu. A supply chain model of vendor managed inventory. *Transportation Research Part E: Logistics and Transportation Review*, 38(2):75–95, 2002.

[70] Moshe Dror. Routing propane deliveries. *Logistics Systems: Design and Optimization*, pages 299–322, 2005.

[71] Moshe Dror and Michael Ball. Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics (NRL)*, 34(6):891–905, 1987.

[72] Moshe Dror, Michael Ball, and Bruce Golden. A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4(1):1–23, 1985.

[73] Moshe Dror, Gilbert Laporte, and Pierre Trudeau. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science*, 23(3):166–176, 1989.

[74] Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[75] Energy Information Administration (EIA), U.S. Department of Energy. Short-term energy outlook. URL: `https://www.eia.gov/outlooks/steo/archives/Dec17.pdf`, 2017. Last visited 2018-07-30.

[76] Energy Information Administration (EIA), U.S. Department of Energy. Heating oil explained: Use of heating oil. URL: `https://www.eia.gov/energyexplained/index.php?page=heating_oil_use`, 2018. Last visited 2018-07-26.

[77] Saher Esmeir and Shaul Markovitch. Anytime learning of decision trees. *Journal of Machine Learning Research*, 8(May):891–933, 2007.

[78] T Fabian, JL Fisher, MW Sasieni, and A Yardeni. Purchasing raw material on a fluctuating market. *Operations Research*, 7(1):107–122, 1959.

[79] Frank Fabozzi, Petter Kolm, Dessislava Pachamanova, and Sergio Focardi. *Robust Portfolio Optimization and Management.* John Wiley & Sons, 2007.

[80] Awi Federgruen and David Simchi-Levi. Analysis of vehicle routing and inventory-routing problems. *Handbooks in Operations Research and Management Science*, 8:297–373, 1995.

[81] Awi Federgruen and Paul Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984.

[82] Marshall Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

[83] Marshall Fisher. An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15(2):10–21, 1985.

[84] Peter Francis, Karen Smilowitz, and Michal Tzur. The period vehicle routing problem and its extensions. In *The vehicle routing problem: latest advances and new challenges*, pages 73–102. Springer, 2008.

[85] Ricardo Fukasawa, Humberto Longo, Jens Lysgaard, Marcus Poggi de Aragão, Marcelo Reis, Eduardo Uchoa, and Renato Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511, 2006.

[86] Michel Gendreau, Alain Hertz, and Gilbert Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290, 1994.

[87] Michel Gendreau, Gilbert Laporte, and René Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

[88] Paul Berhanu Girma. Do heating oil prices adjust asymmetrically to changes in crude oil prices. *Journal of Business & Economics Research*, 9(1), 2011.

[89] Kamal Golabi. Optimal inventory policies when ordering prices are random. *Operations Research*, 33(3):575–588, 1985.

[90] Bruce Golden, Subramanian Raghavan, and Edward Wasil. *The Vehicle Routing Problem: Latest Advances and New Challenges: latest advances and new challenges*, volume 43. Springer Science & Business Media, 2008.

[91] Donald Goldfarb and Garud Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

[92] Bram Gorissen, İhsan Yanıkoğlu, and Dick den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

[93] Chrysanthos Gounaris, Wolfram Wiesemann, and Christodoulos Floudas. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, 61(3):677–693, 2013.

[94] Monique Guignard and Siwhan Kim. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming*, 39(2):215–228, 1987.

[95] Monique Guignard and Moshe B Rosenwein. An application of Lagrangean decomposition to the resource-constrained minimum weighted arborescence problem. *Networks*, 20(3):345–359, 1990.

[96] Helene Gunnarsson, Mikael Rönnqvist, and Dick Carlsson. A combined terminal location and ship routing problem. *Journal of the Operational Research Society*, 57(8):928–938, 2006.

[97] Gurobi Optimization, Inc. Gurobi optimizer reference manual. URL: `http://www.gurobi.com/documentation/8.0/refman/refman.html`, 2015. Last visited 2018-07-11.

[98] Ali Güzel. Optimal commodity procurement under stochastic prices. *Technology*, 7(1):29–39, 2004.

[99] Justin Hall. How the cost of crude oil affects truck freight rates. URL: `http://shiplps.com/how-the-cost-of-crude-oil-affects-truck-freight-rates`, 2016. Last visited 2017-12-04.

[100] Shawkat Hammoudeh, Huimin Li, and Bang Jeon. Causality and volatility spillovers among petroleum prices of WTI, gasoline and heating oil in different locations. *The North American Journal of Economics and Finance*, 14(1):89–114, 2003.

[101] Vera Hemmelmayr, Karl Doerner, and Richard Hartl. A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3):791–802, 2009.

[102] Vera Hemmelmayr, Karl Doerner, Richard Hartl, and Martin Savelsbergh. Delivery strategies for blood products supplies. *OR Spectrum*, 31(4):707–725, 2009.

[103] Vera Hemmelmayr, Karl Doerner, Richard Hartl, and Martin Savelsbergh. Vendor managed inventory for environments with stochastic product usage. *European Journal of Operational Research*, 202(3):686–695, 2010.

[104] Stefan Irnich, Paolo Toth, and Daniele Vigo. Chapter 1: The family of vehicle routing problems. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 1–33. SIAM, 2014.

[105] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[106] Brian Kingsman. Commodity purchasing. *OR*, pages 59–79, 1969.

[107] Brian Kingsman. Purchasing raw materials with uncertain fluctuating prices. *European Journal of Operational Research*, 25(3):358–372, 1986.

[108] Brian Kingsman. *Raw materials purchasing: an operational research approach*, volume 4. Elsevier, 2014.

[109] Anton Kleywegt, Vijay Nori, and Martin Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1):94–118, 2002.

[110] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[111] Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[112] Gilbert Laporte, Hélène Mercure, and Yves Nobert. An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks*, 16(1):33–46, 1986.

[113] Gilbert Laporte, Yves Nobert, and Martin Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.

[114] Geoffrey Lauprete, Alexander Samarov, and Roy Welsch. Robust portfolio optimization. In *Developments in Robust Statistics*, pages 235–245. Springer, 2003.

[115] Hyafil Laurent and Ronald Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[116] Bing Liu. Route finding by using knowledge about the road network. *Systems, Man and Cybernetics, Part A: Systems and Humans*, 27(4):436–448, 1997.

[117] Alvaro Lorca, Xu Andy Sun, Eugene Litvinov, and Tongxin Zheng. Multistage adaptive robust optimization for the unit commitment problem. *Operations Research*, 64(1):32–51, 2016.

[118] Jens Lysgaard, Adam Letchford, and Richard Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, 2004.

[119] Somayeh Moazeni, Thomas Coleman, and Yuying Li. Optimal portfolio execution strategies and sensitivity to price impact parameters. *SIAM Journal on Optimization*, 20(3):1620–1654, 2010.

[120] Somayeh Moazeni, Thomas Coleman, and Yuying Li. Regularized robust optimization: the optimal portfolio execution case. *Computational Optimization and Applications*, 55(2):341–377, 2013.

[121] Sreerama Murthy and Steven Salzberg. Lookahead and pathology in decision tree induction. In *International Joint Conference on Artificial Intelligence*, pages 1025–1033. Citeseer, 1995.

[122] Karthik Natarajan, Melvyn Sim, and Joline Uichanco. Tractable robust expected utility and risk models for portfolio optimization. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, 20(4):695–731, 2010.

[123] George Nemhauser. Integer programming: The global impact. URL: `https://smartech.gatech.edu/bitstream/handle/1853/49829/presentation.pdf`, 2013. Last visited 2018-07-11.

[124] Mohammad Norouzi, Maxwell Collins, Matthew Johnson, David Fleet, and Pushmeet Kohli. Efficient non-greedy optimization of decision trees. In *Advances in Neural Information Processing Systems*, pages 1729–1737, 2015.

[125] Steven Norton. Generating better decision trees. In *International Joint Conference on Artificial Intelligence*, volume 89, pages 800–805, 1989.

[126] Fernando Ordóñez. Robust vehicle routing. *Tutorials in Operations Research*, pages 153–178, 2010.

[127] Joaquín Pacheco, Ada Alvarez, Irma García, and Francisco Angel-Bello. Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates. *Journal of the Operational Research Society*, 63(5):569–581, 2012.

[128] Harold Payne and William Meisel. An algorithm for constructing optimal binary decision trees. *IEEE Transactions on Computers*, 100(9):905–916, 1977.

[129] Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. In *Integer Programming and Combinatorial Optimization*, pages 393–403. Springer, 2014.

[130] Andre Perold. The implementation shortfall: Paper versus reality. *The Journal of Portfolio Management*, 14(3):4–9, 1988.

[131] Eric Prescott-Gagnon, Guy Desaulniers, and Louis-Martin Rousseau. Heuristics for an oil delivery vehicle routing problem. *Flexible Services and Manufacturing Journal*, 26(4):516–539, 2014.

[132] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[133] J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Elsevier, 2014.

[134] Ted Ralphs, Leonid Kopman, William Pulleyblank, and Leslie Trotter. On the capacitated vehicle routing problem. *Mathematical Programming*, 94(2-3):343–359, 2003.

[135] Funda Sahin and E. Powell Robinson. Information sharing and coordination in make-to-order supply chains. *Journal of Operations Management*, 23(6):579–598, 2005.

[136] Frédéric Semet and Eric Taillard. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, 41(4):469–488, 1993.

[137] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*, volume 16. SIAM, 2014.

[138] Oguz Solyali, Jean-François Cordeau, and Gilbert Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, 2012.

[139] Nguyen Hung Son. From optimal hyperplanes to optimal decision trees. *Fundamenta Informaticae*, 34(1, 2):145–174, 1998.

[140] Ilgaz Sungur, Fernando Ordóñez, and Maged Dessouky. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5):509–523, 2008.

[141] Christos Tjortjis and John Keane. T3: a classification algorithm for data mining. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 50–55. Springer, 2002.

[142] Paolo Toth and Daniele Vigo. *The vehicle routing problem.* Society for Industrial and Applied Mathematics, 2001.

[143] Paolo Toth and Daniele Vigo. *Vehicle Routing: Problems, Methods, and Applications*, volume 18. SIAM, 2014.

[144] Panagiotis Tzirakis and Christos Tjortjis. T3C: improving a decision tree classification algorithm's interval splits on continuous attributes. *Advances in Data Analysis and Classification*, 11(2):353–370, 2017.

[145] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi, and Walter Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624, 2012.

[146] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[147] Ryan Walker, Paul McKenzie, Christine Liddell, and Chris Morris. Spatial analysis of residential fuel prices: Local variations in the price of heating oil in Northern Ireland. *Applied Geography*, 63:369–379, 2015.

[148] Matt Waller, M. Eric Johnson, and Tom Davis. Vendor-managed inventory in the retail supply chain. *Journal of Business Logistics*, 20(1):183, 1999.

[149] Weather Underground. Weather history for KBOS. URL: http://www.wunderground.com/history/airport/KBOS/2013/11/1/CustomHistory.html?dayend=31&monthend=3&yearend=2014&req_city=&req_state=&req_statename=&reqdb.zip=&reqdb.magic=&reqdb.wmo=, 2014. Last visited 2015-09-12.

[150] Yuliang Yao, Philip Evers, and Martin Dresner. Supply chain integration in vendor-managed inventory. *Decision Support Systems*, 43(2):663–674, 2007. Emerging Issues in Collaborative Commerce.

[151] Tinglong Zhang, Liang Liang, Yugang Yu, and Yan Yu. An integrated vendor-managed inventory model for a two-echelon system with order cost reduction. *International Journal of Production Economics*, 109(1):241–253, 2007. Special Section on Cost Engineering.