2012

Robust Optimization with Multiple Ranges and Chance Constraints

Ruken Duzgun
Lehigh University

ROBUST OPTIMIZATION WITH MULTIPLE RANGES AND CHANCE CONSTRAINTS

by

Ruken Düzgün

Presented to the Graduate and Research Committee
of Lehigh University
in Candidacy for the Degree of
Doctor of Philosophy

in
Industrial Engineering

Lehigh University

May 2012

Approved and recommended for acceptance as a proposal in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Date

Dr. Aurélie C. Thiele
Dissertation Advisor

Accepted Date

Committee:

Dr. Aurélie C. Thiele, Chairwoman

Dr. Joseph C. Hartman

Dr. Katya Scheinberg

Dr. Robert H. Storer

Acknowledgments

It is a pleasure to thank to the kind people around me, only some of whom it is possible to mention here, for making this thesis possible.

This thesis would not have been possible without the help and support of my supervisor, Dr. Aurélie C. Thiele. I owe my deepest gratitude to her for her support (academic, moral and financial), patience, availability and kindness. She provided her advice and guidance whenever I got lost during the Ph.D. pursuit, encouragement whenever I worried and friendship whenever I needed. She has been invaluable on both an academic and a personal level, and I am very happy to have her in my life.

I would like to acknowledge the committee members for being in my dissertation committee and their constructive comments on this thesis. I am grateful to Dr. Katya Scheinberg for her valuable feedbacks and contribution to my thesis and her help in my career. I am thankful to Dr. Robert H. Storer and Dr. Joseph C. Hartman, for their advices and contributions to this study. I am especially grateful to Dr. Hartman for bringing up the application idea of Multi-Range Robust Optimization to Prioritizing Project Selection problem.

I would like to thank to my first research advisor, Dr. Jitamitra Desai. He was a great teacher and I was among the few lucky students in the department that enjoyed learning from him. I am also thankful to the other faculty members for their help in my doctoral education; particularly, to Dr. George Wilson, Dr. Ted Ralphs and Dr. Frank Curtis.

I gratefully acknowledge the financial, academic and technical support of the Lehigh University, Industrial and Systems Engineering Department, particularly in the fellowship and assistantships that provided the financial support for this research. I would also like to thank to Rita Frey, Kathy Rambo and Brianne Lisk for their endless help in every bureaucratic paperwork at the university.

My time at Lehigh was made enjoyable in large part due to the many friends and groups that became a part of my life. I would like to express my apology that I could not mention personally one by one. I am grateful for the time spent with them. I thank to Zeliha Akca for her kindness, time and generosity. She was the one who made life easier when nothing was easy. She opened her heart and her door to us on our first night in the US. I also thank to my dearest friends, Nadia Zea Viteri, Elcin Cetinkaya and Figen Bekisli. I cannot imagine how difficult life would be in Bethlehem without them. I am grateful to my friend Gokhan Metan for his time, advices and answers to my endless questions. And, to my other friends, Ilgin - Aykagan Ak, Burak Bekisli, Oznur - Erdem Arslan, Pinar - Hakan Guvelioglu, Hamra - Hakan Bakircioglu, Berrin Aytac, Zumbul - Oguz Atan, Burcu - Alper Uygur, Menal Guzelsoy, Cem Alper Ozen, Ana Alexandrescu and Tiffany Tong, thank you.

Lastly but not the least, I would like to thank to my parents and brothers for all their love and encouragement. And most of all, I would like to thank to my loving husband Tolga for his support, encouragement and great patience at all times. No word would be enough to express my appreciation to him. He had more confidence in me than I had for the most of the time and he was there, with me, whenever I needed him. Without him, none of this would have happened. Thank you very much.

Contents

List of Tables

List of Figures

Abstract

We present a robust optimization approach with multiple ranges and chance constraints.

The first part of the dissertation focuses on the case when the uncertainty in each objective coefficient is described using multiple ranges. This setting arises when the uncertain coefficients, such as cash flows, depend on an underlying random variable, such as the effectiveness of a new drug. Traditional one-range robust optimization would require wide ranges and lead to conservative results. In our approach, the decision-maker limits the numbers of coefficients that fall within each range and that deviate from the nominal value of their range.

We show how to develop tractable reformulations to this mixed-integer problem and apply our approach to a R&D project selection problem. Furthermore, we develop a robust ranking heuristic, where the manager ranks projects according to densities (ratio of cash flows to development costs) or Net Present Values. While both heuristics perform well in experiments, the NPV-based heuristic performs better; in particular, it finds the optimal solution more often.

We show the how to use multi-range robust optimization approach to have a robust project selection problem. While this approach can imitate the stochastic optimization's scenario settings, our problem is significantly faster than stochastic optimization, since we do not have the burden of having many scenarios. We also develop a robust approach to price optimization in presence of other retailers.

The last part of the dissertation connects robust optimization with chance constraints and shows that the Bernstein approximation of robust binary optimization problems leads to robust counterparts of the same structure as the deterministic models, but with modified objective coefficients that depend on a single new parameter introduced in the approximation.

Chapter 1

Literature Review

This chapter describes the traditional models of optimization under uncertainty, provides definitions for robust optimization, and discusses recent advances in the robust optimization literature.

1.1 Traditional Models of Optimization under Uncertainty

1.1.1 Stochastic Programming

Information in real-life applications is often revealed in stages, forcing the manager to make decisions with only limited knowledge of the data, and to adjust his strategy as he observes the realization of random parameters such as customer demand over time. Dantzig [35] first investigated decision-making under uncertainty in the 1950s, pioneering the field now known as Stochastic Programming (SP). This methodology assumes that parameters are random but obey a known discrete distribution and that the decision-maker minimizes (or maximizes) the expected objective value over the possible scenarios. The most frequent setup has two stages, with two groups of decision variables: *here-and-now* (or first-stage) variables, which represent decisions made before the manager can observe the resolution of the uncertainty, and *wait-and-see* (or second-stage) variables, which represent recourse actions. The SP problem is linear but of potentially very large size, motivating the use of structure-specific solution methods. The structure here is that the constraints either connect first-stage decision variables with each other, or first-stage decision variables with second-stage decision variables for a specific scenario, but never second-stage decision variables for different scenarios. Such a structure is said to be L-shaped because of the position of the non-zero elements in the coefficient matrix. The second-stage problem, formulated for given first-stage variables, is called the *recourse problem*. It can be shown to be convex and piecewise linear. The main algorithm used to solve this problem relies on generating pieces of the recourse function as needed, which can be interpreted as a delayed constraint generation algorithm. The reader is referred to Birge and Louveaux [29], Kall and Wallace [43], Prékopa [58], and Ruszczyński and Shapiro [59] and the references therein for a wide-ranging treatment of SP.

If recourse decisions are required to be integer (which is called SP with integer recourse), the integrality constraints increase the complexity of the second-stage problem significantly and the master problem becomes very hard to solve. Stochastic integer programs (where some variables are forced to be integer, in the first and/or second stages) are computationally intractable. Algorithms used to solve these problems include variants of the L-shaped method based on Benders decomposition (Laporte and Louveaux [48], Carøe and Tind [30]), branch-and-cut (Sen and Sherali [62], Ntaimo and Sen [55]) and branch-and-bound (Ahmed et. al. [2]), among others.

Drawbacks. Under the assumption that the stochastic parameters are independently distributed, Dyer and Stougie [38] show that the two-stage SP problems are NP-hard. Furthermore, the number of scenarios grows exponentially with the number of parameters when the latter are independent, creating tractability issues. (For instance, a retailer considering fifteen independent products with demand for each taking three possible values would need to generate over fourteen million scenarios.) Moreover, it is often difficult to estimate probability distributions accurately. Even when the distributions of random parameters for past time periods can be estimated with a high degree of precision using historical data, these distributions might differ from future ones in unpredictable ways due to changing conditions. The difficulties faced by two-stage stochastic programming are compounded in the multistage case, where multiple piecewise linear recourse functions, corresponding to different time periods, need to be approximated by generating linear pieces as needed. Shapiro and Nemirovski [65] provides an in-depth discussion of the complexity of two-stage and multistage SP problems, and argues that multistage SP problems are, in general, intractable.

1.1.2 Dynamic Programming

Dynamic Programming (DP) deals with multi-stage decision-making. The key idea is that the manager behaves optimally at all time periods, which reduces the number of strategies to be considered. This is formulated in mathematical terms through a system of recursive equations known as the Bellman equations:

$$V_t(S_t) = \min_{x_t \in X_t} \{C_t(S_t, x_t) + E\{V_{t+1}(S_{t+1}|S_t, x_t)\}\} \ \forall S_t \in \mathcal{S}_t, \tag{1.1}$$

where $V_t(S_t)$ is the *optimal* value (cost-to-go function) associated with being in state S_t in stage t and S_{t+1} is the state at the next time period. $\mathcal{S}_t$ is the space of possible states at time t. Intuitively, the equations state that the manager will follow the optimal strategy from time $t+1$ onward, so that the problem at time t reduces to minimizing the current costs plus the optimal costs incurred from the next time period up to the end of the time horizon.

Solving Problem (1.1) requires the investigation of structural properties of the value functions to guarantee global optimality. DP is most insightful when the optimal policy can be proved to have a certain structure and the decision-maker only has to compute the parameters defining that policy (for instance, basestock levels in dynamic inventory management under uncertainty, see Bertsekas [13]). It is important to note that DP leads to optimal *policies*, rather than optimal numbers, so that the decision-maker does not need to resolve his problem as time progresses and uncertainty is revealed: he simply implements the pre-computed policy that corresponds to the state and the time period he is in. Because DP relies on the fact that the decision-maker will act on new information at the next stage, it is sometimes called *closed-loop optimization* (new observations are incorporated in a feedback loop), in contrast with open-loop optimization, which generates numbers rather than policies and where the problem needs to be re-solved at each time period to capture new information. (An advantage of open-loop optimization is that it is much less computationally demanding.) Readers may refer to Bertsekas [13] and Powell [57] for detailed treatments of DP.

Drawbacks. Value functions need to be stored for each time period and each possible state at that time period, creating severe dimensionality challenges as the size of the state space and/or the time horizon increases. This is known as the "curse of dimensionality", which makes DP impractical in many applications. Approximate Dynamic Programming attempts to overcome these shortcomings (Powell [57] and Bertsekas [14]) but remains hard to implement. In addition, as for stochastic programming, it can be difficult to estimate the underlying probability distributions accurately.

In summary, there are two main issues with the traditional methods of dynamic decision-making under uncertainty:

i. Probability distributions are difficult to estimate accurately,

ii. Dimensionality issues arise even when the distributions of the random parameters are exactly known.

We will see in subsequent parts of this article how robust optimization can address these issues.

1.2 Possible Definitions for Robust Optimization

Robust Optimization (RO) is another method that addresses data uncertainty. Unlike SP and DP, RO models uncertainty assuming that uncertain parameters belong to a bounded, convex uncertainty set. While SP minimizes expected cost (or maximize expected revenue), RO is a worst-case analysis and minimizes the maximum value of the objective over the uncertainty set. This approach was pioneered by Soyster [66] in the 1970s, although he did not call it "robust optimization" at the time. He assumed that each uncertain parameter took values in an uncertainty *interval* and proposed an optimization model to generate feasible solutions for the worst case. Because the model led to each parameter being equal to its worst-case value, it was thought to be too conservative for implementation in business; however, the issue of solution feasibility has remained an important topic of research, which saw critical advances in the 1990s.

1.2.1 Scenario-Based Robust Optimization

The expression "robust optimization" became popular in the mid-1990s, when Mulvey et al. [53] proposed a scenario-based description of the problem data and penalized the variance of the optimal solution for any realization of the scenarios, given a certain level of risk expressed by the decision-maker. This approach does not match Soyster's and is not in line with later uses of the expression "robust optimization", which will be described below. We mention it here because of its relation to stochastic programming as well as to avoid confusion for the reader familiar with [53] and subsequent approaches called robust optimization.

Consider the LP optimization model:

$$\begin{aligned} \min_{x\in\mathbb{R}_1^n, y\in\mathbb{R}_2^n} \quad & c^T x + d^T y \\ \text{s.t.} \quad & Ax = b, \\ & Fx + Gy = h, \\ & x, y \geq 0. \end{aligned} \tag{1.2}$$

Let $\Omega = (1, 2, ..., S)$ be the set of scenarios with p_s probability of scenario s, where each scenario $s \in \Omega$ is associated with the vector (d_s, F_s, G_s, h_s) of realizations for the uncertain coefficients. y_s is the second-stage control vector for scenario $s \in \Omega$; also, we introduce a new vector z_s, which measures the extent of the constraint infeasibility under data scenario s. The objective function $\xi = c^T x + d^T y$ becomes a random variable that takes values $\xi_s = c^T x + d_s^t y_s$ with probability p_s for $s \in \Omega$.

The robust counterpart of Problem (1.2) using the Mulvey-Vanderbei-Zenios framework can then be formulated as:

$$\begin{aligned} \min \quad & \sigma(x, y_1, ..., y_s) + \omega \cdot \rho(z_1, ..., z_s) \\ \text{s.t.} \quad & Ax = b, \\ & F_s x + G_s y_s + z_s = h_s, \forall s \in \Omega \\ & x, y_s \geq 0 \; \forall s \in \Omega. \end{aligned} \tag{1.3}$$

$\rho(z_1, \ldots, z_s)$ term in the objective function is a feasibility penalty function and $\sigma(x, y_1, \ldots, y_s)$ is the optimality robustness term. (The scalar ω captures the decision-maker's tradeoff between the two goals of feasibility and optimality.) If the solution is "close" to the optimal for any realization of the scenario $s \in \Omega$, then it is referred to as "solution-robust". If it remains "almost" feasible for any scenario realization, then it is referred to as "model-robust". The authors suggest possible functions in [53] to help users define solution-robustness and model-robustness. For instance, for moderate- to high-risk decisions under uncertainty, they advocate the use of the average objective plus a constant (λ) times its variance as an appropriate solution-robustness term:

$$\sigma(x, y_1, ..., y_s) = \sum_{s \in S} p_s \xi_s + \lambda \cdot \sum_{s \in S} p_s \left(\xi_s - \sum_{s' \in S} p_{s'} \xi_{s'} \right)^2$$

An example of feasibility penalty function is:

$$\rho(z_1, ..., z_s) = \sum_{s \in S} p_s \, z_s^T \, z_s$$

where both positive and negative violations of the control constraints are penalized equally.

In [53], the authors apply this technique to applications such as power capacity expansion, matrix balancing and image reconstruction, air-force airline scheduling, scenario immunization for financial planning, and minimum-weight structural design. Other applications include capacity expansion of telecommunication networks (Laguna [47]), a more in-depth look at power capacity expansion (Malcolm and Zenios [50]) and the portfolio management of callable bonds (Vassiadou-Zeniou and Zenios [70]).

Drawbacks. This approach to robust optimization changes the structure of the deterministic model;

the robust model is no longer linear. In addition, it suffers from the same dimensionality issues encountered in stochastic programming.

1.2.2 Set-Based Robust Optimization

Robust optimization addresses data uncertainty by assuming that uncertain parameters belong to a bounded, convex uncertainty set and maximizing the minimum value of the objective over that uncertainty set, while ensuring feasibility for the worst-case value of the constraints. Soyster's model [66] required that each uncertain parameter be equal to its worst-case value, and thus was deemed too conservative for practical implementation. In the mid-1990s, Ben-Tal and Nemirovski ([9], [10], [11]), El-Ghaoui and Lebret [40] and El-Ghaoui et al. [41] focus w.l.o.g. on uncertainty in the constraints of mathematical programming problems and define robust solutions as solutions that are feasible for the worst-case value of the parameters within an uncertainty set. They use ellipsoidal uncertainty sets and propose a tractable mathematical reformulation that turned linear programming problems into second-order cone problems, while reducing the conservatism of Soyster's [66] approach. Furthermore, Ben-Tal and Nemirovski [12] study robust optimization applied to conic quadratic and semidefinite programming. The reader is referred to Ben-Tal et. al. [6] for an overview of robust optimization, with an emphasis on ellipsoidal sets. One drawback of that framework is that it increases the complexity of the nominal problem.

Bertsimas and Sim [23, 24] and Bertsimas et al. [20] investigated in the early 2000s the special case where the uncertainty set is a polyhedron. Specifically, the uncertainty set consists in range forecasts (confidence intervals) for each parameter and a constraint called a budget-of-uncertainty constraint, which limits the number of coefficients that can take their worst-case value. The approach preserves the degree of complexity of the problem (the robust counterpart of a linear problem is linear) and allows the decision-maker to control the degree of conservatism of the solution. Bertsimas and Sim [24] also provide a probabilistic guarantee of constraint violation. A drawback of the approach is that it adds auxiliary variables and constraints to the initial formulation. Bertsimas and Thiele [26] surveys the robust optimization literature up to 2006, especially for polyhedral uncertainty sets.

Let c be the objective coefficient vector of size n. The general model we consider is:

$$\begin{aligned} \max \quad & \mathbf{c}'\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{1.4}$$

where $\mathcal{X}$ is the constraint set of $\mathbf{x}$, which may include integrality constraints. We further assume that all decision variables are non-negative, which is a natural assumption to make in the context of operations management, where decision variables represent for instance ordering quantities or amounts transported; this assumption is particularly justified in the project management application described in Section 3.2, where decision variables are binary.

We consider the case where the vector **c** is uncertain, which will correspond to uncertain project cash flows in Section 3.2. We can apply the traditional one-range robust optimization approach that Bertsimas and Sim developed in [20], [24] to the uncertain parameter **c**. Specifically, we model c_i, $i = 1, \ldots, n$, as an uncertain parameter in the interval $[\overline{c}_i - \widehat{c}_i, \, \overline{c}_i + \widehat{c}_i]$. (Note that, since decision variables are non-negative, the worst case will always be achieved at the low end of the range; therefore, knowledge of the high end of the range is not required to implement the approach and the confidence interval does not have to be symmetric.) Define the scaled deviation y_i such that $c_i = \bar{c}_i + \hat{c}_i \, y_i$ for all i. In line with Bertsimas and Sim [24], the scaled deviations are assumed to belong to the polyhedral uncertainty set:

$$\mathcal{P} = \{y | \sum_{i=1}^{n} |y_i| \leq \Gamma, \quad |y_i| \leq 1, \, \forall i\}.$$

The parameter $\Gamma \in [0, n]$ is the *budget of uncertainty* which specifies the maximum number of coefficients that can deviate from their nominal values.

- If $\Gamma = 0$, the only feasible element in $\mathcal{P}$ is the zero vector, so that the problem reduces to its deterministic counterpart.

- If $\Gamma = n$, each uncertain parameter takes its worst case value.

- Taking a value of Γ between 0 and n allows the decision-maker to achieve a trade-off between the nominal performance of the deterministic model and the risk protection of the most conservative model.

While the setup above assumes that the project cash flows are independent, it is straightforward to extend the approach to the case where cash flows are correlated by using the multi-factor model described in Bertsimas and Sim [24]. This extension is left to the reader.

The robust problem becomes:

$$\begin{array}{rll} \max & \min & \displaystyle\sum_{i=1}^{n} (\bar{c}_i + \hat{c}_i\, y_i)\, x_i \\ & \text{s.t.} & \mathbf{y} \in \mathcal{P} \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}. & \end{array} \tag{1.5}$$

Theorem 1.1 (One-range robust optimization (Bertsimas and Sim [24])) *The robust counterpart of Problem (5.1) is:*

$$\begin{array}{rll} \max & \displaystyle\sum_{i=1}^{n} \bar{c}_i\, x_i \; - \Gamma z_0 - \sum_{i=1}^{n} z_i & \\ s.t. & \mathbf{x} \in \mathcal{X} & \\ & z_i + z_0 \geq \hat{c}_i\, x_i, & \forall i, \\ & z_i, z_0 \geq 0 & \forall i. \end{array} \tag{1.6}$$

Proof. This is a direct application of Bertsimas and Sim [24] to Problem (1.5) after injecting the fact that the worst case is always achieved for $y_i \leq 0$ for all i and that the decision vector $\mathbf{x}$ is non-negative.

Those early works spearheaded significant research efforts on the theory and practice of robust optimization. Bertsimas and Brown [15] provide a methodology to construct uncertainty sets within the framework of robust optimization, using the decision-maker's risk preferences expressed by a coherent risk measure. RO has been applied to inventory management (Bertsimas and Thiele [27], Bienstock and Ozbay [28]), revenue management (Adida and Perakis [1]), portfolio management

(Bertsimas and Pachamanova [19], Kawas and Thiele [44]), telecommunications (Ye and Ordóñez [71]), among others. Notice that, while RO was initially developed to address data ambiguity (i.e., uncertain parameters whose value was unknown but constant), the methodology was subsequently applied to random variables with uncertain distributions; in the early 2000s, Bertsimas and Thiele [27] was the first work to model random variables as uncertain parameters and apply robust optimization to the deterministic problem. In the context of multi-stage decision-making, this leads to *open-loop problems*. A drawback is that these problems had to be reoptimized at each time period, as a necessary tradeoff to achieve tractable formulations.

In addition, if the knowledge of the probability distributions driving the random variables is imprecise, but the manager knows that the distribution belongs to a family of distributions, then it is possible to implement a robust optimization approach to the uncertain probability density functions themselves. The manager optimizes his objective over the worst-case value of the probabilities. This is referred to in the literature as the *minimax* or *min-max* approach (rather than robust optimization); see Dupacova [36], Shapiro and Kleywegt [64] and Shapiro [63]. Thiele [68] applies the robust optimization approach using polyhedral uncertainty sets to stochastic optimization problems and provides theoretical insights into the solution. Unfortunately, the RO approach to stochastic programming is only as tractable as the underlying SP problem for the nominal probabilities.

Goh and Sim [42] suggests tractable approximations to distributionally robust optimization using piecewise-linear decision rules. Ben-Tal et al. [4] proposes a framework for robust optimization that extends the standard notion of robustness by allowing the decision-maker to vary the protection level across the uncertainty set. This captures the fact that at least some partial probabilistic description of the world is available in many applications such as finance. The approach in [4] allows for different performance guarantees for different subsets within $\mathcal{P}$, where $\mathcal{P}$ represents the set of possible underlying probability measures for the random variables. For instance, performance guarantees of a portfolio can then be linked in a smooth way to the performance of the market as a whole. The authors call this new approach the *soft robust approach* and show that it preserves convexity properties of the nominal problem.

Dimensionality issues explain why researchers have focused on the alternative view of RO, which incorporates uncertainty to the *deterministic* formulation of the problem, instead of considering a stochastic description of uncertainty. While the tractability of the RO approach is appealing, the limitations of open-loop policies have incited researchers to investigate dynamics models in robust optimization in more depth. Of particular interest has been the development and analysis of decision rules with a specific structure, which are described in the following section.

1.3 Robust Dynamic Optimization

1.3.1 Linear Decision Rules and Adjustable Optimization

Ben-Tal et al. [8] extend the scope of RO by introducing the Adjustable RO methodology. Consider the uncertain linear programming problem:

$$\left\{ \min_x \left\{ c^T x : Ax \leq b \right\} \right\}_{\zeta \equiv [A,b,c] \in \mathcal{Z}} \tag{1.7}$$

where $\zeta \equiv [A, b, c]$ lies in the uncertainty set $\mathcal{Z} \subset \mathbb{R}^n \times \mathbb{R}^{m \times n} \times \mathbb{R}^m$ a nonempty compact convex set. (This section uses the same notation as [8].)

The Robust Counterpart (RC) of the uncertain problem (1.7) is defined as:

$$\min_x \left\{ \sup_{\zeta \equiv [A,b,c] \in \mathcal{Z}} (c^T x) : Ax - b \leq 0 \ \ \forall \zeta \equiv [A, b, c] \in \mathcal{Z} \right\}$$

In the traditional RO approach, all the variables represent "here-and-now" decisions, so that the optimal solution x must be chosen to optimize the worst-case objective over all possible values of ζ in $\mathcal{Z}$. Ben-Tal et al. [8] suggest to incorporate "wait-and-see" decisions, that is, decisions that are taken after the uncertainty is realized. In the RO terminology, the variables that correspond to recourse action are called *adjustable*, while the others are called *non-adjustable*. The vector x in (1.7) is partitioned according to non-adjustable (u) and adjustable (v) variables such that $x = (u^T, v^T)^T$ and Problem (1.7) becomes:

$$\min_{(s,u),v}\left\{s : c^T\begin{pmatrix} u \\ v \end{pmatrix} \le s, Uu + Vv \le b\right\}_{[U,V,b,c]\in\mathcal{Z}}$$

where (s, u) represents the non-adjustable part of the solution. The uncertain LP (1.7) problem can be rewritten w.l.o.g. (after redefining parameters appropriately) as:

$$LP_{\mathcal{Z}} = \left\{\min_{u,v} c^T u : Uu + Vv \le b\right\}_{[U,V,b]\in\mathcal{Z}} \tag{1.8}$$

The matrix V is called the *recourse matrix*. When V is not subject to uncertainty, Problem (1.8) is a *fixed recourse* LP.

Definition 1.1 *The Adjustable Robust Counterpart (ARC) of the uncertain LP problem $LP_{\mathcal{Z}}$ is defined as:*

$$ARC : \min_u \left\{c^T u : \forall\zeta = [U, V, b] \in \mathcal{Z}\ \ \exists v : Uu + Vv \le b\right\}. \tag{1.9}$$

ARC has a larger robust feasible set and hence is more flexible and less conservative than the robust counterpart (RC). On the other hand, it is a semi-infinite LP problem; it suffers from computational tractability issues and is NP-hard even for simple uncertainty sets. As a remedy, the authors introduce the Affinely Adjustable Robust Counterpart (AARC) of an uncertain LP by restricting the adjustable variables to be *affine* functions of the corresponding data. In other words, v is forced to be of the form:

$$v = w + W\zeta$$

for some w, W parameters to be determined.

Definition 1.2 *The Affinely Adjustable Robust Counterpart (AARC) of (1.8) is defined as the optimization problem:*

$$\min_{u,w,W} \left\{ c^T u : Uu + V(w + W\zeta) \leq b \ \forall \zeta = [U, V, b] \in \mathcal{Z} \right\}. \tag{1.10}$$

Ben-Tal et al. [8] shows that if the uncertainty set $\mathcal{Z}$ in a *fixed recourse* problem $LP_{\mathcal{Z}}$ is computationally tractable (in the sense that a tractable separation oracle exists), the AARC is also computationally tractable, that is, polynomially solvable. When the uncertainty set is also "well-structured" (i.e., given by a list of linear matrix inequalities such as polyhedral, conic quadratic or semidefinite representations), then the corresponding AARC is also "well-structured" and thus can be solved by linear or semidefinite programming techniques. On the other hand, if the recourse matrix V is subject to uncertainty, the AARC of $LP_{\mathcal{Z}}$ can become computationally intractable. [8] shows that in that case, AARC has a tight computationally tractable approximation (which is an explicit semidefinite programming problem). Ben-Tal et al. [7] apply the AARC heuristic to two-echelon multiperiod supply chain problem (specifically, a retailer-supplier flexible commitment (RSFC) problem) and derive a single deterministic convex optimization problem that is either a linear or a conic-quadratic problem.

Ordóñez and Zhao [56] identify the conditions on the uncertainty set that would lead to a tractable ARC approach in the case of a robust capacity expansion problem for network flows. They consider the classic network flow problem with additional decision variables representing arc capacity expansions. The capacity expansion problem can be written as:

$$\begin{aligned} z_D(b,c) = \quad \min_{x,y} \quad & c^T x \\ \text{s.t.} \quad & Nx = b \\ & x \leq u + y \\ & d^T y \leq q \\ & x, y \geq 0, \end{aligned} \tag{1.11}$$

where $c \in \mathbb{R}^m$ are the transportation cost coefficients, $x \in \mathbb{R}^m$ are the arc flow variables and $y \in \mathbb{R}^m$ are the decision variables representing the new arc capacities. $N \in \mathbb{R}^{n\times m}$ is the node-arc incidence matrix, initial arc capacities are given by $u \in \mathbb{R}^m$ and the demand-supply vector is given by $b \in \mathbb{R}^n$. Expanding the capacity of arc i costs d_i where q is the total budget for investment. The demand b and the travel times c belong to closed, convex and bounded uncertainty sets U_b and U_c respectively. It is assumed that the y variables are determined prior to the realization of the uncertain data $(b,\, c)$ and the flow variables x will adapt to the realized data. Then the ARC of robust capacity expansion problem (RCEP) is defined as:

$$
\begin{aligned}
z_{ARC} = \quad & \min_{y,\gamma} \quad \gamma \\
& \text{s.t.} \quad d^T y \le q \\
& \qquad\; y \ge 0 \\
& \qquad\; \forall c \in U_c,\;\; b \in U_b \; \exists x : \begin{cases} Nx = b \\ 0 \le x \le u + y \\ c^T x \le \gamma. \end{cases}
\end{aligned}
$$

Ordóñez and Zhao [56] presents three cases (problem with fixed demand, single commodity problem with uncertain demand and multicommodity problem with uncertain demand), where the RCEP is formulated as a conic problem and solved by interior point methods in polynomial time.

Shapiro and Nemirovski [65] states that the main reason for using linear decision rules is their tractability but linear decision rules are rarely optimal. In other words, there is no guarantee that the true optimal solution is close to the one given by the linear decision rule and the optimality gap is not known. Linear decision rules may perform poorly or even lead to infeasible instances even in the case of complete recourse. Chen et al. [32] gives the following example: suppose that the support of $\tilde{z}$ is $\mathcal{W} = \{-\infty, \infty\}$. Then the nonnegativity constraint on:

$$
w(\tilde{z}) = w^0 + \sum_{k=1}^{N} w^k \, \tilde{z}_k
$$

implies that

$$w^k = 0 \ \ \forall k \in \{1, ..., N\},$$

and the decision rule reduces to $w(\tilde{z}) = w^0$, which is independent of the underlying uncertainty and may lead to infeasibility.

Ben-Tal et al. [5] extends the AARC approach by relaxing the "uncertain-but-bounded" assumption of RO that characterizes the uncertain parameters. Requiring that all realizations of the uncertain data lie in the uncertainty set may be pessimistic (leading to overly large sets) or infeasible if the random variables have unbounded support. The approach in [5] exhibits controlled performance degradation for "large deviations" in the uncertain data. Assume that the uncertainty set $\mathcal{Z}$ represents the typical range for the uncertain data. When $\zeta \in \mathcal{Z}$, the solution must satisfy the constraints of the problem. When the data ζ falls outside the uncertainty set $\mathcal{Z}$, the violation of the constraints should not exceed a prescribed multiple of the deviation of the data from its normal range. These multiples serve as "global sensitivities" of the constraints. Ben-Tal et al. [5] calls this extension of AARC the *Comprehensive Robust Counterpart* of uncertain linear problems.

1.3.2 Piecewise Constant Rules and Adaptive Optimization

Bertsimas and Caramanis [16] introduce a variable degree of adaptability, on the grounds that complete adaptability can be too expensive and assuming the exact realization of uncertainty (which is required to implement AARC above) is overly optimistic. The generic problem Bertsimas and Caramanis consider is:

$$\min \left\{ c^T x : Ax \geq b, \ \ \forall A \in \mathcal{P} \right\}, \tag{1.12}$$

where $\mathcal{P}$ is a polytope. The re-optimization formulation is given by:

$$\max_{A \in \mathcal{P}} \left\{ \min \left\{ c^T x : Ax \geq b \right\} \right\}. \tag{1.13}$$

The authors introduce the concept of k-adaptability (see Problem (1.14)), which is formulated as a disjunctive optimization problem with infinitely many constraints. The decision-maker selects k solutions and commits to one solution after uncertainty is revealed; at least one of the k solutions must be feasible regardless of the realization of the uncertainty.

$$\begin{array}{ll} \min & \max\{c^T x_1, c^T x_2, ..., c^T x_k\} \\ s.t. & [Ax_1 \geq b \, or \, Ax_2 \geq b \, or \, ...Ax_k \geq b] \;\; \forall A \in \mathcal{P} \end{array} . \tag{1.14}$$

Problem (1.14) is equivalent to the k-partitioning problem (1.15) where the uncertainty set $\mathcal{P}$ has been partitioned into a finite number k of regions: $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup ... \cup \mathcal{P}_k$. The decision-maker learns which region of the partition will contain the realization of the uncertainty before he has to commit to a solution.

$$\min_{\mathcal{P}=\mathcal{P}_1 \cup ... \cup \mathcal{P}_k} \left[\begin{array}{ll} \min & \max\{c^T x_1, c^T x_2, ..., c^T x_k\} \\ s.t. & Ax_1 \geq b, \;\; \forall A \in \mathcal{P}_1 \\ & \vdots \\ & Ax_k \geq b, \;\; \forall A \in \mathcal{P}_k \end{array} \right] . \tag{1.15}$$

Problem (1.15) represents an application of finite adaptability to *single-stage optimization*. For two-stage optimization as in the ARC model (1.9), the problem becomes:

$$\min_{\mathcal{P}=\mathcal{P}_1 \cup ... \cup \mathcal{P}_k} \left[\begin{array}{ll} \min : c^T u & \\ \quad s.t. & Uu + Bv_1 \geq b, \;\; \forall (U,V) \in \mathcal{P}_1 \\ & \vdots \\ & Uu + Bv_k \geq b, \;\; \forall (U,V) \in \mathcal{P}_k \end{array} \right] . \tag{1.16}$$

Let V_{RO}, V_{ReOpt}, V^k_{adapt} be the optimal objective values of Problems (1.12), (1.13) and (1.15), respectively. We have:

$$V_{RO} \geq V^k_{adapt} \geq V_{ReOpt}.$$

Furthermore, with an additional continuity assumption, $\lim_{k\to\infty} V^k_{adapt} = V_{ReOpt}$. Hence, finite adaptability bridges the gap between robust optimization (total lack of adjustability, where all decision variables are here-and-now) and re-optimization (total adjustability, where all decision variables are wait-and-see).

While the k-adaptability problem was formulated above as a disjunctive program, it can also be formulated as a bilinear optimization program. For $k = 2$, the bilinear optimization problem becomes:

$$\begin{array}{ll} \min & \max\{c^T x_1, c^T x_2\} \\ s.t. & u_{i,j}[(A^l x_1)_i - b_i] + (1 - u_{i,j})[(A^l x_2)_j - b_j] \geq 0 \;\; \forall 1 \leq i,j \leq m \;\; \forall 1 \leq l \leq K \;\cdot \\ & 0 \leq u_{i,j} \leq 1, \;\; \forall 1 \leq i,j \leq m \end{array}$$

m is the number of rows of the constraint matrix A and K is the number of extreme points of the uncertainty set $\mathcal{P}$, where $\mathcal{P}$ is defined as the convex hull of its extreme points: $\mathcal{P} = conv(A^1, \ldots, A^K)$. Bertsimas and Caramanis [16] proves that obtaining the optimal partition $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ in $2 - adaptability$ is NP-hard in general, provides a hierarchy of the levels of adaptability and proposes a heuristic tractable algorithm.

Bertsimas and Goyal [18, 17] consider two-stage adaptive optimization problems and investigate the power and limitations of robust solutions and affine policies. In [18], Bertsimas and Goyal show that the robust optimization approach is a good approximation to solving the corresponding two-stage mixed integer stochastic optimization problem to optimality. They compare the optimal cost of the robust problem to the optimal costs of the stochastic problem and the adaptability problem. If the uncertainty set and the probability distribution over the uncertain set are symmetric, and if the second-stage variables are continuous variables, the optimal cost of the robust problem is at most twice the expected cost of the optimal two-stage solution to the stochastic problem. In [17], Bertsimas and Goyal show that an affine policy is optimal if the uncertainty set $u \in \mathbb{R}^m_+$ is a convex combination of $m + 1$ affinely independent points in $\mathbb{R}^m_+$ and this optimality result is almost tight. They also prove that if the uncertainty set is a polytope, the worst case cost occurs at an extreme point of the uncertainty set. On the other hand, for uncertainty sets with $m + 2$ non-zero extreme points, the affine policy is suboptimal. Bertsimas and Goyal [17] also give a lower and upper bound for the performance of an optimal affine policy compared to an optimal fully adaptable two-stage solution.

1.3.3 Other Approaches

Thiele et al. [69] develop a robust optimization approach for generic two-stage stochastic problems with uncertainty on the right-hand side. The authors apply a cutting plane algorithm based on Kelley's algorithm to the robust linear problem with general recourse and test the methodology on a multi-item newsvendor problem and production planning example where the demand is uncertain but must be met.

Chen et al. [31] generalizes the robust linear optimization frameworks of [11] and [24] by introducing a new uncertainty set that captures the asymmetry of the underlying random variables through the use of new deviation measures (forward and backward deviations). They integrate these deviation measures to the uncertainty set and obtain solutions to chance-constrained problems. Applying the linear decision rule of [8], Chen et al. [31] present a tractable robust optimization approach in order to find less conservative feasible solutions for stochastic programming with chance constraints. This framework also leads to the computational scalability of multistage stochastic models. Chen et al. [32] propose a novel framework to approximate multistage stochastic optimization by introducing two new piecewise linear decision rules. The first one is called *deflected linear decision rule*, which is best-suited for SP problems with semi-complete recourse and provides a tighter approximation of the original objective function than the linear decision rule. The second one is called *segregated linear decision rule*, which is best-suited for SP problems with general recourse. When combined with the first rule, the *segregated linear decision rule* exhibits better performance than both linear and deflected linear decision rules. Under these new piecewise linear rules, the authors show that the computational tractability of the problems is preserved. Chen and

Zhang [33] and See and Sim [61] extend the theory of robust optimization to approximate solutions of multistage problems. See and Sim [60] applies these robust optimization approaches to multi-period inventory control.

Chapter 2

Positioning and Contributions

2.1 Positioning

The robust optimization approaches we mentioned in the previous chapter assume that the uncertain parameters belong to a convex uncertainty set; in particular, for the model with polyhedral uncertainty sets, each parameter belongs to a pre-defined confidence interval (range forecast) and can take any value within that range. While this is valuable in many applications, it has limitations for random variables with more complex underlying distributions, such as cases where the uncertain parameters are driven by underlying random variables. For instance, in the case of drug trials, the potential revenue of a drug will depend on the effectiveness of the active chemical compound being tested; if the performance of the compound is disappointing, the resulting cash flows will fall in a low range; if the compound is effective in healing a wide array of patients, cash flows will fall in a high range. Trying to encompass all possible values of the cash flows into a single interval will generate an overly large range forecast, with an ill-defined nominal value lacking any realistic meaning if it falls between the two intervals, as the decision-maker never believes he will observe such cash flows.

This is particularly a concern in robust optimization, since it can be shown (see Bertsimas and Sim [24]) that at optimality, the worst-case coefficients of Problem (1.5) will be equal to either their worst case or their nominal value, assuming the budget of uncertainty is integer. Hence, it is important for the relevance of the robust optimization approach and its adoption by practitioners that the optimal values of the uncertain coefficients correspond to values these parameters can actually take. Similar arguments can be made in the case of demand for a new product, the sales of which depends on the degree of popularity or market share that the product will achieve. Such items, with a wide range of possible outcomes, require a finer-grained representation of uncertainty than the one-range model is able to provide.

Our focus throughout the dissertation is to develop robust optimization models that capture complex features of real-life uncertain systems, features that are not incorporated in the traditional robust optimization framework. We aim to thus make robust optimization more appealing to practitioners. In the first part of our work, we focus on problem setups where the ranges taken by uncertain coefficients depend on the realizations of underlying random variables; we then investigate connections between uncertainty sets and probabilistic constraints via "safe tractable approximations." We consider applications drawn from the field of R&D project selection, where the need to model multiple ranges arises frequently, for instance when project cash flows are uncertain but also depend on the effectiveness of the underlying compound tested by the pharmaceutical company. Project selection requires binary variables, for which ellipsoidal uncertainty sets are ill-suited as they lead to nonlinear integer problems (Bertsimas and Sim [25]); therefore, we will focus throughout this dissertation on polyhedral uncertainty sets, specifically, sets with range forecasts and budget-of-uncertainty constraints. The traditional robust optimization approach, with a single range for each uncertain coefficient, would require very large ranges and thus lead to overly conservative solutions. The multi-range robust optimization approach we propose allows for a more realistic description of uncertainty. While Metan and Thiele [52] introduces multiple ranges for product demand in a simple two-stage robust revenue management problem for a single product, that approach is an hybrid between robust optimization and stochastic programming, where the decision-maker gains advance knowledge of the range that product demand will fall into. It incorporates neither binary variables nor budgets of uncertainty and has a single source of uncertainty, and focuses on the impact of scenario probabilities on the quality of the optimal solution. To the best of our knowledge, we are the first to present a generic, comprehensive multi-range approach in the context of robust optimization.

The Research and Development (R&D) project selection problem has been studied since the 1960s. Competition between R&D companies has increased the importance of funding projects that would best meet their needs. While many methods to identify these projects have been investigated, there is no consensus on their practical effectiveness. Martino [51] presents various methods available for selecting R&D projects, in particular ranking methods, economic models, portfolio or optimization models and ad-hoc methods.

Early studies of the R&D project selection problem mostly use ranking methods. The most common ones are scoring models and the analytic hierarchy procedure (AHP) (see Baker and Freeland [3] for a literature review on these approaches.) Economic methods, which are recommended by Martino [51], consider the cash flows involved with the project, using metrics such as net present value (NPV), internal rate of return (IRR) and cash flow payback. Portfolio optimization methods implement mathematical programming to find the projects, from a candidate project list, that would give the maximum payoff to the firm. For instance, Childs and Triantis [34] use a real options framework in order to examine dynamic R&D investment policies and valuation of R&D programs, and Stummer and Heidenberger [67] use a multi-objective integer programming model to determine all efficient (Pareto-optimal) portfolios.

Data envelopment analysis (DEA) is another method for solving R&D project selection decisions. Linton et. al [49] proposed this method to split decisions on project portfolios into accept, consider-further and reject sub-groups. Eilat et. al [39] use a methodology based on an extended DEA that quantifies some qualitative concepts embedded in the balanced scorecard (BSC) approach. They employ a DEA-BSC model first to evaluate individual R&D projects, and then to evaluate alternative R&D portfolios.

R&D project selection problems include high levels of uncertainty in future cash flows; however, the most common approaches to project selection replace uncertain parameters by their expected values or rely on traditional, stochastic descriptions of randomness, although quantifying accurately the probability distributions of future cash flows for a R&D project and the probabilities of project success is very difficult in practice. As mentioned above, the classical robust optimization approach also suffers from over-conservatism in this setup due to the large ranges that would be required to implement it. This makes multi-range robust optimization a novel theoretical extension of robust optimization with valuable practical applications.

2.2 Contributions

Our contributions to the literature are as follows.

- We define the multi-range robust optimization framework and derive tractable reformulations.
- We connect robust optimization to chance constraints in binary optimization using safe tractable approximations.
- We provide robust rankings, which allow practitioners to gain insights into what makes a project valuable and to implement optimization-free heuristics.

Multi-Range Optimization

- We show that the linear relaxation of the inner minimization problem (which computes the worst-case objective for a given strategy and requires binary variables to model multiple ranges) has integer optimal solutions in both robust optimization models we consider.
- We apply the approach to a R&D project selection problem.
- We present a robust ranking heuristic to identify projects to fund without any optimization and test it in numerical experiments.
- Our computational results suggest that, in this setting, ranking projects according to Net Present Values rather than densities (ratio of cash flows to development costs), yields higher-quality solutions, i.e., solutions closer to optimality.

Robust Prioritizing Project Selection

- We show the how to use two-range robust optimization approach to have a robust project prioritizing problem.

- Multi-range robust optimization approach allows us to consider all the possible values for the uncertain parameter in a tractable optimization problem. We do not need many scenarios.

- While our robust approach can imitate the stochastic optimization approach's scenario settings, our problem is significantly faster than stochastic optimization approach, since we do not have the burden of having many scenarios.

Robust Pricing

- We develop a robust optimization approach to pricing decisions in presence of other retailers.

- We formulate tractable robust models for price optimization problems when the demand is a linear function of the prices.

Chance Constraints

- We show that the safe tractable approximation (called Bernstein approximation) to binary optimization problems is equivalent to a deterministic problem with modified cost coefficients, which only depend on problem data and one extra coefficient.

- We consider two cases: (i) when the uncertain parameters obey a jointly Normal distribution, which allows us to demonstrate the insights we can gain in the simplest setting when we know, in closed form, both the distributions of the uncertain parameters and of the objective function, (ii) when we only know the first two moments and the support of the distributions of the uncertain parameters. Our conclusions are valid for both.

- We investigate an iterative approach to address the risk of over-conservatism of the safe tractable approximation approach, which arises because the methodology uses Markov's Inequality for non-negative random variables, admittedly an unsophisticated bound connecting tail percentile with expected value.

- We compare our approach in numerical experiments with the one proposed by Bertsimas and Sim [23], also for binary optimization problems with uncertain coefficients but for a different modeling of uncertainty when probability distributions are not known, and argue that, while solution quality is comparable, the solution times in our approach are substantially smaller.

Chapter 3

Multi Range Robust Optimization

3.1 Multi-Range Robust Optimization

Instead of having a single range of uncertainty, we now assume that we have multiple ranges that the uncertain values can take values from. For notational simplicity, we assume that each uncertain parameter has the same number m of possible ranges, but the approach can be extended easily to the case where the number of ranges depends on the uncertain parameter. We will analyze two cases:

i. The simple case where the (pessimistic) decision-maker assumes that each uncertain parameter takes the worst value of the range it falls into, and the maximum number of parameters that can fall in a given range is bounded by a budget of uncertainty.

ii. The more complex case where the decision-maker extends the setup in Case 1 to introduce another family of budgets of uncertainty limiting the number of parameters that can take their worst-case value in a given range. This allows some parameters to be equal to their nominal value, rather than their worst-case value, in that range.

Case 1: Without a Budget For the Deviations Within The Ranges

Let c_i^{k-}, resp. c_i^{k+} be the lower, resp. higher, bound of range k for parameter i, $i = 1, \ldots, n$, $k = 1, \ldots, m$. The budget Γ_k constrains the maximum number of coefficients that can fall within range k, $k = 1, \ldots, m$. (The decision maker can also choose to introduce these budgets only for the lowest ranges, corresponding to the most conservative outcomes, to limit the conservatism of the approach.)

The robust problem can be formulated as a mixed-integer programming problem (MIP):

$$
\begin{array}{lll}
\max\limits_{\mathbf{x}\in\mathcal{X}} & \min\limits_{c,y} & \mathbf{c}'\mathbf{x} \\
& \text{s.t.} & c_i^{k-}\, y_i^k \leq c_i^k \leq c_i^{k+}\, y_i^k, \quad \forall i, k, \\
& & \sum_{k=1}^{m} y_i^k = 1, \quad \forall i, \\
& & c_i = \sum_{k=1}^{m} c_i^k, \quad \forall i, \\
& & \sum_{i=1}^{n} y_i^k \leq \Gamma_k, \quad \forall k, \\
& & y_i^k \in \{0, 1\}, \quad \forall i, k.
\end{array}
\tag{3.1}
$$

The tractability of the robust optimization paradigm relies on the decision-maker's ability to convert the inner minimization problem into a maximization problem, of such a structure that the master maximization problem (incorporating the outer maximization problem and the new inner maximization problem) can be solved efficiently. Strong duality has emerged as the tool of choice to implement this conversion (Bertsimas and Sim [24]); however, the model of uncertainty we propose require the use of integer (binary) variables, which makes the rewriting of a minimization problem as an equivalent maximization one considerably more difficult. It is thus natural to investigate whether the linear relaxation of the inner minimization problem in Problem (3.1) yields binary $\mathbf{y}$ variables at optimality. This is the purpose of Lemma 3.1.

Lemma 3.1 *The linear relaxation of the inner minimization problem:*

$$\begin{aligned}
\min_{\mathbf{c},\mathbf{y}} \quad & \mathbf{c}'\mathbf{x} \\
\textit{s.t.} \quad & c_i^{k-}\, y_i^k \le c_i^k \le c_i^{k+}\, y_i^k, && \forall i, k, \\
& \sum_{k=1}^{m} y_i^k = 1, && \forall i, \\
& c_i = \sum_{k=1}^{m} c_i^k, && \forall i, \\
& \sum_{i=1}^{n} y_i^k \le \Gamma_k, && \forall k, \\
& y_i^k \in \{0,1\}, && \forall i, k,
\end{aligned} \tag{3.2}$$

has a binary optimal vector $\mathbf{y}$ *for any given integer* Γ_l *and nonnegative vector* $\mathbf{x}$*.*

Proof. The objective is a minimization over $\mathbf{c}$ of $\mathbf{c}'\mathbf{x}$ where $c_i = \sum_{k=1}^{m} c_i^k$ for all i and $\mathbf{x}$ is non-negative. Hence, c_i^k will take the minimum value in its range, i.e., $c_i^k = c_i^{k-}\, y_i^k$ at optimality for all i, k. It follows that $c_i = \sum_{k=1}^{m} c_i^{k-}\, y_i^k$ for all i and the feasible set is reduced to $\sum_{k=1}^{m} y_i^k = 1,\ \forall i,$ $\sum_{i=1}^{n} y_i^k \le \Gamma_k,\ \forall k,$ and $y_i^k \in \{0,1\},\ \forall i, k,$. The feasible set of the linear relaxation has binary extreme points, thus proving the lemma.

This allows us to derive a tractable reformulation of Problem (3.1).

Theorem 3.2 *Problem (3.1) has the equivalent robust linear formulation:*

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} p_i - \sum_{k=1}^{m} \gamma^k\, \Gamma_k - \sum_{i=1}^{n}\sum_{k=1}^{m} z_i^k \\
\textit{s.t.} \quad & p_i - \gamma^k - z_i^k \le c_i^{k-}\, x_i, && \forall i, k, \\
& \mathbf{x} \in \mathcal{X} \\
& \gamma^k, z_i^k \ge 0, && \forall i, k.
\end{aligned} \tag{3.3}$$

Proof. As in the proof of Lemma 3.1, we notice that, due to the non-negativity of the vector $\mathbf{x}$, the optimal objective coefficients in the robust optimization framework are always achieved at the low end of the range. Therefore, we can rewrite the group of constraints:

$$c_i^{k-}\, y_i^k \le c_i^k \le c_i^{k+}\, y_i^k, \;\; c_i = \sum_{k=1}^{m} c_i^k,$$

as:

$$c_i = \sum_{k=1}^{m} c_i^{k-}\, y_i^k.$$

We use Lemma 3.1 to rewrite Problem (3.2) as:

$$\begin{aligned} \min_{c,y,u} \quad & \sum_{i=1}^{n} \sum_{k=1}^{m} c_i^{k-}\, y_i^k\, x_i, & \\ \text{s.t.} \quad & \sum_{k=1}^{m} y_i^k = 1, & \forall i, \\ & \sum_{i=1}^{n} y_i^k \le \Gamma_k, & \forall k, \\ & 0 \le y_i^k \le 1, & \forall i, k, \end{aligned} \tag{3.4}$$

which is a linear programming problem with a non-empty, bounded feasible set. We can then invoke strong duality to reformulate the minimization as a maximization problem, i.e., replace the primal formulation by its dual. Re-injecting yields Problem (3.3).

Case 2: With a Budget For the Deviations Within the Ranges

In practice, it is unlikely that every single uncertain parameter will take the worst-case value of the range it falls in. The purpose of this section is to extend the robust optimization approach presented in Section 3.1 to the case where the manager also decides how many parameters, at most, can take the worst-case value in the ranges they are in.

As before, the uncertain coefficients satisfy:

$$
\begin{aligned}
& c_i = \sum_{k=1}^{m} c_i^k,\ \forall i, \\
& c_i^{k-}\, y_i^k \le c_i^k \le c_i^{k+}\, y_i^k,\ \forall i, k, \\
& \sum_{k=1}^{m} y_i^k = 1,\ \forall i, \\
& \sum_{i=1}^{n} y_i^k \le \Gamma_k,\ \forall k, \\
& y_i^k \in \{0, 1\},\ \forall i, k.
\end{aligned}
$$

Because we need to define the deviation of each parameter within its given range, we further assume that the *nominal value* of parameter i in range k, denoted $\bar{c}_i^k$, is known for all $i = 1, \ldots, n$ and $k = 1, \ldots, m$. The *measure of uncertainty* for parameter i of range k is then defined as $\hat{c}_i^k = \bar{c}_i^k - c_i^{k-}$ for all $i = 1, \ldots, n$ and $k = 1, \ldots, m$. Again, because the decision variables are non-negative, the part of the range forecast above the nominal value will not be used in the robust optimization approach and the optimal uncertain coefficients satisfy:

$$
c_i = \sum_{k=1}^{m} (\bar{c}_i^k - \hat{c}_i^k\, z_i^k)\, y_i^k,
$$

where z_i^k is the scaled deviation of coefficient i, $i = 1, \ldots, n$, from its nominal value in range k, $k = 1, \ldots, m$ with:

$$
\begin{aligned}
& \sum_{i=1}^{n} \sum_{k=1}^{m} z_i^k \le \Gamma, \\
& 0 \le z_i^k \le 1,\ \forall i, k.
\end{aligned}
$$

Lemma 3.3 *For any feasible* $\mathbf{x} \in \mathcal{X}$*, the worst-case objective can be computed as a mixed-integer* *programming problem:*

$$
\begin{aligned}
\min_{\mathbf{c},\mathbf{y}} \quad & \sum_{i=1}^{n}\sum_{k=1}^{m} x_i \left(\bar{c}_i^k\, y_i^k - \hat{c}_i^k\, u_i^k\right) & \\
\textit{s.t.} \quad & u_i^k \le y_i^k, & \forall i,k, \\
& \sum_{i=1}^{n}\sum_{k=1}^{m} u_i^k \le \Gamma, & \\
& \sum_{k=1}^{m} y_i^k = 1, & \forall i, \\
& \sum_{i=1}^{n} y_i^k \le \Gamma_k, & \forall k, \\
& y_i^k \in \{0,1\}, & \forall i,k, \\
& u_i^k \ge 0, & \forall i,k.
\end{aligned}
\tag{3.5}
$$

Proof. Defining $u_i^k = z_i^k\, y_i^k$, we obtain:

$$c_i^k = \bar{c}_i^k\, y_i^k - \hat{c}_i^k\, u_i^k,\ \forall i,k,$$

where $0 \le u_i^k \le y_i^k$. The result follows from the fact that it is suboptimal to have $z_i^k > 0$ when $u_i^k = 0$ for any i,k.

The following lemma is key to the tractability of the robust optimization approach we present.

Lemma 3.4 *The constraint matrix of Problem (3.5) is totally unimodular.*

Proof. A matrix obtained by a pivot operation on a totally unimodular matrix is totally unimodular (Nemhauser and Wolsey [54]). The matrix A below is the constraint matrix of Problem (3.5) where the columns represent the variables $[u \quad y]$.

$$C = \begin{pmatrix} I_{nm} & -I_{nm} \\ 1_{1\times nm} & 0_{1\times nm} \\ 0_{m\times nm} & A_{n\times nm} \\ 0_{n\times nm} & B_{m\times nm} \end{pmatrix}$$

where I_{nm} is the $nm \times nm$ identity matrix and matrix $A_{n\times nm}$ has the following structure:

$$\begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots & & \vdots & \\ & \vdots & & & & & \ddots & & \vdots & \\ 0 & \cdots & 0 & & \cdots & & & 1 & \cdots & 1 \end{pmatrix}$$

Specifically, A is defined as:

$$A_{i,j} = \begin{cases} 1 & \text{if } (i-1)m < j \leq im \\ 0 & \text{otherwise,} \end{cases}$$

Matrix $B_{m\times nm}$ has the following structure:

$$\begin{pmatrix} 1 & \cdots & 0 & 1 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 1 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

Specifically, B has the following structure:

$$B = \begin{pmatrix} I_{m\times m} & I_{m\times m} & \cdots & I_{m\times m} \end{pmatrix}$$

We will do the following operations on C.

1) Let R_j is the j^{th} row and R^j is the j^{th} column of C. By doing the row operations, we obtain the r-th version of the matrix C which is denoted by $(C)^r$

For $j = nm + 2$ to $nm + 2 + m$, do

$$-R_j + R_{nm+1} \rightarrow R_{nm+1}$$

and call the resulting matrix $(C)^m$.

Now, for $j = 1$ to nm, do

$$R^j + R^{j+nm} \rightarrow R^{j+nm}.$$

and call the resulting matrix $(C)^{m(n+1)}$.

2) At the end of these row/colum operations we obtain the matrix $(C)^{m(n+1)}$, which is

$$(C)^{m(n+1)} = \begin{pmatrix} I_{nm} & 0_{nm} \\ 1_{1\times nm} & 0_{1\times nm} \\ 0_{m\times nm} & A_{n\times nm} \\ 0_{n\times nm} & B_{m\times nm} \end{pmatrix}$$

To conclude the proof, we will need the following result.

Lemma 3.5 *(Nemhauser and Wolsey [54], p. 544) Let A be a $(0, 1, -1)$ matrix with no more than two nonzero elements in each column. Then A is totally unimodular if and only if the rows of A can be partitioned into two subsets Q_1 and Q_2 such that if a column contains two nonzero elements, the following statements are true:*

a. If both nonzero elements have the same sign, then one is in a row contained in Q_1 and the other is in a row contained in Q_2.

b. If the two nonzero elements have opposite sign, then both are in rows contained in the same subset.

The matrix $(C)^{m(n+1)}$ satisfies these conditions of total unimodularity. Since a matrix obtained by pivot operations on a totally unimodular matrix is also totally unimodular, our constraint matrix

C is totally unimodular.

Theorem 3.6 *The robust counterpart is equivalent to the following problem, with a linear objective and linear constraints added to the original feasible set:*

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} p_i - \sum_{i=1}^{n}\sum_{k=1}^{m} z_i^k - \sum_{k=1}^{m} \gamma^k \, \Gamma_k - \Gamma \, \gamma_0 & \\
\textit{s.t.} \quad & \pi_i^k + \gamma_0 \geq \hat{c}_i^k \, x_i, & \forall i, k, \\
& \pi_i^k + p_i - \gamma^k - z_i^k \leq \bar{c}_i^k \, x_i, & \forall i, k, \\
& \mathbf{x} \in \mathcal{X} & \\
& \gamma^k, \gamma_0, \pi_i^k, z_i^k \geq 0 & \forall i, k.
\end{aligned} \tag{3.6}$$

Proof. Since the constraint matrix of Problem (3.5) is totally unimodular (Lemma 3.4) and the right-hand-side values of the constraints are integer, the linear relaxation of the problem has integer optimal solutions. It follows from strong duality, because the feasible set of the linear relaxation of Problem (3.5) is non-empty and bounded, that Problem (3.5) and the dual of its linear relaxation have the same optimal objective. Reinjecting the dual yields Problem (3.6).

3.2 Application to Project Management

3.2.1 Problem Setup

We now apply the setting described in Section 4.2 to an example in R&D project selection. The manager must decide in which projects to invest over a finite time horizon. Each project has known cash requirements at each stage of its development (for notational simplicity, we assume all projects have the same number of stages; this corresponds for instance to the case of drug trials of small, medium and large scale leading to possible approval by the Food and Drug Administration in the United States), but cash flows during and at the end of development are uncertain and depend on underlying random variables, such as the effectiveness of the active compounds or the market response to the new product. These random variables are realized only once (e.g., the drug compound is effective for the disease being treated), so that the coefficients for a given project all fall in the low range or all fall in the high range. We allow for cash flows to be generated during development as the company might file for patents or generate monetary value from the results of the intermediary stages; the biggest cash flows, however, will be generated at the end of the development phase.

We assume that there are two uncertainty ranges for each cash flow: a project might be successful and has high cash flows, or it might be a failure and has low cash flows. Note that cash flows are non-zero, even in the low state, as the drug might be found to be effective on a subset of the patients and retain some market value. Because no new information is revealed during the time horizon in this robust optimization setting, we do not consider the possibility of stopping a project after it has started, before the end of the development phase.

The goal is to maximize the worst-case cumulative Net Present Value of the projects the manager invests in, where the worst case is computed over the uncertainty sets described in Sections 3.1 and 3.1, subject to constraints on the amount of money available at each time period to spend on development. We will use the following notation throughout the paper.

General and cost parameters.

n : number of projects,

T : number of time periods,

S : number of development phases for each project,

B_t : available budget for the time period t where t = 1,..., T,

$CD_{i,s}$: development cost of project i in phase s,

r : discount rate at each time period.

Cash flow parameters.

$CF^{l-}_{i,s}$: lower bound of cash flow of project i in phase s if unsuccessful,

$\overline{CF}^{l}_{i,s}$: nominal value of the cash flow of project i in phase s if unsuccessful,

$CF^{l+}_{i,s}$: upper bound of cash flow of project i in phase s if unsuccessful,

$\widehat{CF}^{l}_{i,s}$: measure of uncertainty for cash flow of project i in phase s in low range,

$CF^{h-}_{i,s}$: lower bound of cash flow of project i in phase s if successful,

$\overline{CF}^{h}_{i,s}$: nominal value of the cash flow of project i in phase s if successful,

$CF^{h+}_{i,s}$: upper bound of cash flow of project i in phase s if successful,

$\widehat{CF}^{h}_{i,s}$: measure of uncertainty for cash flow of project i in phase s in high range,

Robust optimization parameters and decision variables.

Γ_l : uncertainty budget that restricts the number of projects whose cash flows will be in the low range,

Γ : uncertainty budget that restricts the number of projects whose cash flows deviate from their nominal value within their given range,

$x_{i,\tau}$: 1 if the project i is selected to begin at time τ, 0 otherwise,

y_i : 1 if the project i is in its low range (unsuccessful), 0 otherwise.

The deterministic project selection problem where each project can be selected at most once is formulated as:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau}}{(1+r)^{\tau-1}} \left[\sum_{s=1}^{S} \frac{CF_{i,s}}{(1+r)^s} \right] \\ \text{s.t.} \quad & \sum_{i=1}^{n} \sum_{\tau=\max\{1,t-S+1\}}^{t} CD_{i,t-\tau+1}\, x_{i,\tau} \le B_t \quad \forall t \\ & \sum_{\tau=1}^{T} x_{i,\tau} \le 1, \\ & x_{i,\tau} \in \{0,1\}, \qquad \forall i, \tau. \end{aligned} \tag{3.7}$$

3.2.2 Case 1: Robust Optimization Without a Budget for the Deviation Within the Ranges

First, we consider the simple case where the manager only limits the number of projects that will be unsuccessful, and assumes that each cash flow will take its worst case within a given range. Problem (3.1) becomes:

$$
\begin{array}{lll}
\max\limits_{x} \;\; \min\limits_{CF_{i,s},y_i} & \displaystyle\sum_{i=1}^{n}\sum_{\tau=1}^{T-S+1}\frac{x_{i,\tau}}{(1+r)^{\tau-1}}\left[\sum_{s=1}^{S}\frac{CF_{i,s}}{(1+r)^{s}}\right] & \text{Total cash flow over time} \\
\qquad\qquad \text{s.t.} & \text{Cash flow interval if in low range} & \\
& CF^{l-}_{i,s}\,y_i \le CF^{l}_{i,s} \le CF^{l+}_{i,s}\,y_i & \forall(i,s) \\
& \text{Cash flow interval if in high range} & \\
& CF^{h-}_{i,s}\,(1-y_i) \le CF^{h}_{i,s} \le CF^{h+}_{i,s}\,(1-y_i) & \forall(i,s) \\
& \text{Cash flow is either high or low} & \\
& CF^{l}_{i,s} + CF^{h}_{i,s} = CF_{i,s} & \forall(i,s) \\
& \text{At most } \Gamma_l \text{ projects in low range} & \\
& \displaystyle\sum_{i=1}^{n} y_i \le \Gamma_l & \\
& y_i \in \{0,1\}, & \forall i \\
& CF^{l}_{i,s}, CF^{h}_{i,s}, CF_{i,s} \ge 0 & \forall(i,s) \\
\text{s.t.} & \text{Budget constraint at each time period} & \\
& \displaystyle\sum_{i=1}^{n}\sum_{\tau=\max\{1,t-S+1\}}^{t} CD_{i,t-\tau+1}\,x_{i,\tau} \le B_t & \forall t \\
& \text{Each project started at most once} & \\
& \displaystyle\sum_{\tau=1}^{T} x_{i,\tau} \le 1, & \forall(i) \\
& x_{i,\tau} \in \{0,1\}, & \forall i,\tau.
\end{array}
\tag{3.8}
$$

The theoretical results in Section 3.1 show Problem (3.8) can be reformulated in a tractable manner.

Theorem 3.7 *Problem (3.8) is equivalent to the mixed-integer programming problem:*

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} \sum_{s=1}^{S} \frac{x_{i,\tau}\, CF_{i,s}^{h-}}{(1+r)^{\tau+s-1}} \;-\; z_l\,\Gamma_l - \sum_{i=1}^{n} z_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} \sum_{\tau=\max\{1,t-S+1\}}^{t} CD_{i,t-\tau+1}\, x_{i,\tau} \leq B_t, && \forall t, \\
& \sum_{\tau=1}^{T-S+1} x_{i,\tau} \leq 1, && \forall i, \\
& z_l + z_i \geq \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau}}{(1+r)^{\tau-1}} \sum_{s=1}^{S} \frac{(CF_{i,s}^{h-} - CF_{i,s}^{l-})}{(1+r)^{s}}, && \forall i, \\
& x_{i,\tau} \in \{0,1\}, && \forall i,\tau, \\
& z, z_i \geq 0, && \forall i.
\end{aligned}
\tag{3.9}
$$

Proof. The proof is a straightforward application of Theorem 3.2.

3.2.3 Case 2: Robust Optimization With a Budget for the Deviation Within the Ranges

Assume that cash flows for project i in phase s, with $i = 1,\ldots,n$, $s = 1,\ldots,S$, are either in $[\overline{CF}^{l}_{i,s} - \widehat{CF}^{l}_{i,s}, \overline{CF}^{l}_{i,s} + \widehat{CF}^{l}_{i,s}]$ or $[\overline{CF}^{h}_{i,s} - \widehat{CF}^{h}_{i,s}, \overline{CF}^{h}_{i,s} + \widehat{CF}^{h}_{i,s}]$. In line with the framework in Section 3.1, they can be written in mathematical terms as:

$$
CF_{i,s} = (\overline{CF}^{l}_{i,s} - \widehat{CF}^{l}_{i,s}\, z^{l}_{i,s})y^{l}_{i} + (\overline{CF}^{h}_{i,s} - \widehat{CF}^{h}_{i,s}\, z^{h}_{i,s})y^{h}_{i},
$$

with $0 \leq z^{l}_{i,s}, z^{h}_{i,s} \leq 1\ \forall i,s$ and $y^{j}_{i} \in \{0,1\}\ \forall j \in \{l,h\}$. Since the coefficients must belong to one of the two ranges, we only introduce a budget-of-uncertainty constraint on the number of coefficients that fall into their *low* range.

Given feasible binary variables $x_{i,\tau}$ (equal to 1 if project i is started at time τ and 0 otherwise), the worst-case cash flows are given by:

$$
\begin{aligned}
\min_{u^l,u^h,y} \quad & \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau}}{(1+r)^{\tau-1}} \left[\sum_{s=1}^{S} \frac{\overline{CF}^l_{i,s}\, y^l_i - \widehat{CF}^l_{i,s}\, u^l_{i,s} + \overline{CF}^h_{i,s}\, y^h_i - \widehat{CF}^h_{i,s}\, u^h_{i,s}}{(1+r)^s} \right] \\
\text{s.t.} \quad & u^l_{i,s} \le y^l_i, && \forall i, s, \\
& u^h_{i,s} \le y^h_i, && \forall i, s, \\
& y^l_i + y^h_i = 1, && \forall i, \\
& \sum_{i=1}^{n} y^l_i \le \Gamma_l, \\
& \sum_{i=1}^{n} \sum_{s=1}^{S} (u^l_{i,s} + u^h_{i,s}) \le \Gamma, \\
& y^j_i \in \{0,1\}, && \forall i, j, \\
& u^l_{i,s}, u^h_{i,s} \ge 0, && \forall i, s.
\end{aligned}
\tag{3.10}
$$

It is a direct application of Lemma 3.4 that the constraint matrix of Problem (3.10) is totally unimodular.

The robust optimization problem is given by:

$$
\begin{aligned}
\max_{x} \quad & \min_{u^l,u^h,y} \quad \sum_{i=1}^{n} \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau}}{(1+r)^{\tau-1}} \left[\sum_{s=1}^{S} \frac{\overline{CF}^{l}_{i,s}\, y^l_i - \widehat{CF}^{l}_{i,s}\, u^l_{i,s} + \overline{CF}^{h}_{i,s}\, y^h_i - \widehat{CF}^{h}_{i,s}\, u^h_{i,s}}{(1+r)^s} \right] \\
& \quad \text{s.t.} \quad \begin{array}{ll}
u^l_{i,s} \le y^l_i, & \forall i, s, \\
u^h_{i,s} \le y^h_i, & \forall i, s, \\
y^l_i + y^h_i = 1, & \forall i, \\
\sum_{i=1}^{n} y^l_i \le \Gamma_l, & \\
\sum_{i=1}^{n} \sum_{s=1}^{S} (u^l_{i,s} + u^h_{i,s}) \le \Gamma, & \\
y^j_i \in \{0,1\}, & \forall i, j, \\
u^l_{i,s}, u^h_{i,s} \ge 0, & \forall i, s,
\end{array} \\
\text{s.t.} \quad & \begin{array}{ll}
\sum_{i=1}^{n} \sum_{\tau=\max\{1,\, t-S+1\}}^{t} CD_{i,t-\tau+1}\, x_{i,\tau} \le B_t, & \forall t, \\
\sum_{\tau=1}^{T-S+1} x_{i,\tau} \le 1, & \forall i, \\
x_{i,\tau} \in \{0,1\}, & \forall i, \tau.
\end{array}
\end{aligned}
\tag{3.11}
$$

The following theorem provides a tractable reformulation of Problem (3.11). Because it is a straightforward application of Theorem 3.6, we state it without proof.

Theorem 3.8 *The robust optimization problem (3.11) is equivalent to the mixed-integer programming problem:*

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} p_i - \sum_{i=1}^{n} (z_i^l + z_i^h) - \Gamma_l \, \gamma_l - \Gamma \, \gamma_0 \\
\textit{s.t.} \quad & \sum_{i=1}^{n} \sum_{\tau=\max\{1,t-S+1\}}^{t} CD_{i,t-\tau+1} \, x_{i,\tau} \leq B_t, && \forall t, \\
& \sum_{\tau=1}^{T-S+1} x_{i,\tau} \leq 1, && \forall i, \\
& \sum_{s=1}^{S} \pi_{i,s}^l + p_i - \gamma_l - z_i^l \leq \sum_{\tau=1}^{T-S+1} x_{i,\tau} \sum_{s=1}^{S} \frac{\overline{CF}_{i,s}^l}{(1+r)^{\tau+s-1}}, && \forall i, \\
& \sum_{s=1}^{S} \pi_{i,s}^h + p_i - z_i^h \leq \sum_{\tau=1}^{T-S+1} x_{i,\tau} \sum_{s=1}^{S} \frac{\overline{CF}_{i,s}^h}{(1+r)^{\tau+s-1}}, && \forall i, \\
& \pi_{i,s}^l + \gamma_0 \geq \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau} \, \widehat{CF}_{i,s}^l}{(1+r)^{\tau+s-1}}, && \forall i, s, \\
& \pi_{i,s}^h + \gamma_0 \geq \sum_{\tau=1}^{T-S+1} \frac{x_{i,\tau} \, \widehat{CF}_{i,s}^h}{(1+r)^{\tau+s-1}}, && \forall i, s, \\
& x_{i,\tau} \in \{0,1\}, && \forall i, \tau, \\
& \pi_{i,s}^l, \pi_{i,s}^h \geq 0 && \forall i, s, \\
& z_i^l, z_i^h \geq 0, && \forall i, \\
& \gamma_l, \gamma_0 \geq 0.
\end{aligned}
\tag{3.12}
$$

The feasible set can be decomposed as follows:

- The first two groups of constraints are the same as in the deterministic model, representing the maximum amount of money to be allocated at each time period and the fact that a project can be started at most once.

- The third and fourth group of constraints are the dual constraints corresponding to the primary variables y_i^l and y_i^h, respectively, and incorporate the information about the nominal values of the cash flows. Because one of these decision variables (either y_i^l or y_i^h) will be non-zero for each i at optimality, by complementarity slackness, one of the dual constraints will be tight for each i, thus determining p_i as a function of the nominal cash flow for that range and the other dual variables. This will bring the nominal cash flows back into the objective.

- The fifth and sixth group of constraints are the dual constraints corresponding to the primary variables u^l_{is} and u^h_{is}, respectively, and incorporate the information about the uncertainty on the cash flows in each range. At most one of these decision variables (either u^l_{is} or u^h_{is}) will be non-zero for each i at optimality; if it is non-zero, by complementarity slackness, one of the dual constraints will be tight for each i, thus determining either π^l_{is} or π^h_{is} as a function of the uncertainty in that range and the other dual variables. (Otherwise the π^l_{is} and π^h_{is} variables will be at zero.) This will bring the cash flow uncertainty, through the half-range of the confidence intervals, into the objective when needed.

- The other constraints are sign constraints or binary constraints.

The robust formulation (3.12) has $n\,(3{+}T{+}2S){+}2$ decision variables and $T{+}n(3{+}2S)$ constraints in addition to sign and binary constraints; therefore, the size of the mixed-integer programming problem increases linearly with each of the parameters n, T, S (number of projects, length of time horizon, number of development stages) when the others are kept constant.

3.3 Robust Ranking Heuristic

While Problem (3.12) provides an exact formulation of the robust optimization problem for project management, we focus in this section on developing optimization-free heuristics to provide a feasible solution to the robust problem, which would give practitioners more insights into the strategy they implement and the impact of the cash flow parameters.

We are motivated by the fact that, when there is only one time period and one development phase ($T = 1$ and $S = 1$), the project selection problem has the structure of a *knapsack problem*, for which a well-known heuristic is to rank items by decreasing order of density (value to weight ratio) and fill the knapsack until the next item in the list does not fit (see, for instance, Kellerer et. al. [45])). In particular, we provide a *robust ranking procedure* to rank the projects with uncertain cash flows; to the best of our knowledge, we are the first to present such a ranking procedure in the context of robust optimization.

We will consider two ways to rank the projects: (a) according to decreasing density, (b) according to decreasing Net Present Value. Method (a) is motivated by its popularity to solve the generic knapsack problem; Method (b) is motivated by its superior performance in the numerical experiments provided in Section 3.4 and the widespread use of Net Present Value to select projects in practice. Once projects are ranked, we apply the greedy multiple-knapsack heuristic described in Kellerer et. al. [45] to generate a candidate solution. Specifically, we proceed down the ranked list of projects and assign project j to knapsacks $t, \ldots, t+S-1$, with t the smallest integer such that the project development costs fit in all of these knapsacks' capacity.

3.3.1 Case 1: Ranking for the Projects Without a Budget for the Deviation Within the Ranges

Recall that, if there is no budget for the deviation within the ranges, the cash flows always take their worst case within the range, and that the range (high or low) is only selected once, i.e., the range does not change with the development phase.

The high-level idea is to (i) compute two rankings, one using the low range of the cash flows and the other using the high range, (ii) use the low-range ranking until the budget of uncertainty has been used up, and then (iii) use the high-range ranking.

Ranking procedure.

Step 1 Compute the following parameters for all projects i.

Method (a): Densities

$$a_i^h = \sum_{s=1}^{S} \frac{CF_{i,s}^{h-}}{(1+r)^s\, CD_{i,s}}$$
$$a_i^l = \sum_{s=1}^{S} \frac{CF_{i,s}^{l-}}{(1+r)^s\, CD_{i,s}}$$

Method (b): Net Present Values

$$a_i^h = \sum_{s=1}^{S} \left(-CD_{i,s} + \frac{CF_{i,s}^{h-}}{(1+r)^s} \right)$$
$$a_i^l = \sum_{s=1}^{S} \left(-CD_{i,s} + \frac{CF_{i,s}^{l-}}{(1+r)^s} \right)$$

For either method, compute two rankings: in decreasing order of a_i^h, and in decreasing order of a_i^l.

Step 2 Add Γ_l projects to your ranking list corresponding to the projects with the largest Γ_l values of a_i^l. Then proceed to Step 3.

Step 3 Continue until all projects are ranked by choosing the unranked projects according to the largest values of a_i^h, discarding projects that have already been selected in Step 2.

3.3.2 Case 2: Ranking for the Projects With a Budget for the Deviation Within the Ranges

In this case, the cumulative cash flow of a project can take four possible values (four cash flow measures): low value of the low range, nominal value of the low range, low value of the high range, nominal value of the high range. The high-level idea is to (i) compute four rankings, one for each of the possible cash flow measures, (ii) use the "low value of the low range" ranking until one of the two budgets of uncertainty has been used up, (iii) use either the "nominal value of the low range" ranking or the "low value of the high range" ranking (depending on which budget is not yet zero) until the other budget of uncertainty has been used up, and (iv) complete the procedure using the "nominal value of the high range" ranking.

Ranking procedure.

Step 1 Compute the four following parameters for all projects i.

Method (a): Densities

$$A_i^h = \sum_{s=1}^{S} \frac{\overline{CF}_{i,s}^h}{(1+r)^s \, CD_{i,s}}, \quad A_i^l = \sum_{s=1}^{S} \frac{\overline{CF}_{i,s}^l}{(1+r)^s \, CD_{i,s}},$$
$$a_i^h = \sum_{s=1}^{S} \frac{CF_{i,s}^{h-}}{(1+r)^s \, CD_{i,s}}, \quad a_i^l = \sum_{s=1}^{S} \frac{CF_{i,s}^{l-}}{(1+r)^s \, CD_{i,s}}.$$

Method (b): Net Present Values

$$A_i^h = \sum_{s=1}^{S} \left(-CD_{i,s} + \frac{\overline{CF}_{i,s}^h}{(1+r)^s} \right), \quad A_i^l = \sum_{s=1}^{S} \left(-CD_{i,s} + \frac{\overline{CF}_{i,s}^l}{(1+r)^s} \right),$$
$$a_i^h = \sum_{s=1}^{S} \left(-CD_{i,s} + \frac{CF_{i,s}^{h-}}{(1+r)^s} \right), \quad a_i^l = \sum_{s=1}^{S} \left(-CD_{i,s} + \frac{CF_{i,s}^{l-}}{(1+r)^s} \right).$$

Using either method, create four rankings, ranking projects in decreasing order of each of the parameters A_i^h, A_i^l, a_i^h and a_i^l.

Step 2 Choose the projects corresponding to the largest $\min(\Gamma_l, \Gamma)$ values in the ranking based on the a_i^l parameters.

Step 3 If $\Gamma_l > \Gamma$ (all cash flows will now take their nominal value as we have used up the Γ budget, but the manager still expects $\Gamma_l - \Gamma$ projects to have cash flows in the low range), add $\Gamma_l - \Gamma$ projects to the ranked list by using the ranking based on the A_i^l parameters, skipping the projects that have already been selected in Step 2.

Step 4 If $\Gamma - \Gamma_l > 0$, add $\Gamma - \Gamma_l$ projects to the ranked list by using the ranking based on the a_i^h parameters, skipping the projects that have already been selected in Steps 2 and 3.

Step 5 Continue until all projects are ranked by using the ranking based on the A_i^h parameters, skipping the projects that have already been selected in Steps 2, 3 and 4.

3.4 Numerical Example

In this section, we investigate the practical performance of our robust optimization models and heuristics on an example. We focus on the case where $T = 1$ and $S = 1$, for which the mathematical formulation without uncertainty becomes a well-known knapsack problem. Furthermore, we consider two uncertainty ranges: high (indicated by the superscript h in relevant parameters) and low (indicated by the superscript l). We have two main goals in this experiment:

i. Test whether the robust optimization framework does protect against downside risk as advertised.

ii. Test the performance of the heuristics, (a) compared to the optimal solution, (b) compared to each other.

3.4.1 Setup

We first provide the robust optimization formulations for clarity. As this is a special case of Section 3.2, the results are stated without proof.

Case 1: Without a budget of uncertainty for the deviations within the ranges.

The robust optimization problem becomes:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} \frac{CF_i^{h-}\, x_i}{(1+r)} - z_l\, \Gamma_l - \sum_{i=1}^{n} z_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} CD_i\, x_i \leq B, \\
& z_l + z_i \geq \frac{(CF_i^{h-} - CF_i^{l-})}{1+r}\, x_i, \quad \forall i, \\
& x_i \in \{0,1\}, \quad \forall i \\
& z_l, z_i \geq 0, \quad \forall i.
\end{aligned} \tag{3.13}$$

The project density parameters (Method (a)) are given by:

$$A_i^h = \frac{CF_i^{h-}}{(1+r)\,CD_i}, \quad A_i^l = \frac{CF_i^{l-}}{(1+r)\,CD_i}.$$

The project Net Present Value parameters (Method (b)) are given by:

$$A_i^h = -CD_i + \frac{CF_i^{h-}}{1+r}, \quad A_i^l = -CD_i + \frac{CF_i^{l-}}{1+r}.$$

Case 2: With a budget of uncertainty for the deviations within the ranges.

The robust optimization problem becomes:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} p_i - \sum_{i=1}^{n} (z_i^l + z_i^h) - \Gamma_l\,\gamma_l - \Gamma\,\gamma_0 \\
\text{s.t.} \quad & \sum_{i=1}^{n} CD_i\,x_i \le B, \\
& \pi_i^l + p_i - \gamma_l - z_i^l \le \frac{\overline{CF}_i^l}{1+r}\,x_i, && \forall i, \\
& \pi_i^h + p_i - z_i^h \le \frac{\overline{CF}_i^h}{1+r}\,x_i, && \forall i, \\
& \pi_i^l + \gamma_0 \ge \frac{\widehat{CF}_i^l}{1+r}\,x_i, && \forall i, \\
& \pi_i^h + \gamma_0 \ge \frac{\widehat{CF}_i^h}{1+r}\,x_i, && \forall i, \\
& x_i \in \{0,1\}, && \forall i, \\
& \pi_i^l, \pi_i^h, z_i^l, z_i^h \ge 0 && \forall i, \\
& \gamma_l, \gamma_0 \ge 0.
\end{aligned} \tag{3.14}$$

The project density parameters (Heuristic (a)) are given by:

$$A_i^h = \frac{\overline{CF}_i^h}{(1+r)\,CD_i}, \quad A_i^l = \frac{\overline{CF}_i^l}{(1+r)\,CD_i},$$
$$a_i^h = \frac{CF_i^{h-}}{(1+r)\,CD_i}, \quad a_i^l = \frac{CF_i^{l-}}{(1+r)\,CD_i}.$$

The Net Present Value parameters (Heuristic (b)) are given by:

$$A_i^h = -CD_i + \frac{\overline{CF}_i^h}{1+r}, \quad A_i^l = -CD_i + \frac{\overline{CF}_i^l}{1+r},$$
$$a_i^h = -CD_i + \frac{CF_i^{h-}}{1+r}, \quad a_i^l = -CD_i + \frac{CF_i^{l-}}{1+r}.$$

3.4.2 Numerical Results

We tested our formulations and heuristics for 4 data sets. Data Sets 1 and 2 have 10 projects while Data Sets 3 and 4 have 20 projects. In all cases, development costs (CD_i) were generated using a Uniform distribution in $[80 - 120]$, nominal values of low cash flows ($\overline{CF}_i^l$) were generated using Uniform distribution in $(0.5 - 2.5) \cdot CD_i$, and nominal values of high cash flows ($\overline{CF}_i^h$) generated using Uniform distribution in $(2 - 3.5) \cdot CD_i$. For all i, the deviation parameters $\widehat{CF}_i^l$, $\widehat{CF}_i^h$ were selected as $0.2\,\overline{CF}_i^l$, $0.2\,\overline{CF}_i^h$ respectively. Budget for development costs was set to 500 in all cases. In addition, Data Sets 3 and 4 were also solved for a value of the budget equal to 1,000. The same distributions were used to compute the actual objective using random cash flows once the optimization problem had been solved. The probability of the cash flows being in the low range was taken equal to 0.5.

Optimal solution.

We solved Problem (3.12) for each data set and for each (Γ, Γ_l) combination. Figure 3.1 shows the histogram of revenues for Data Set 1 and the deterministic model, where parameter values are taken equal to their expected values, here $(\overline{CF}_i^h + \overline{CF}_i^l)/2$ for all i (red line with square markers) as well as two robust models: $(\Gamma, \Gamma_l) = (2, 1)$ and $(\Gamma, \Gamma_l) = (3, 4)$ (blue line with lozenge markers and green line with triangle markers, respectively). These budgets were chosen to have $\Gamma > \Gamma_l$ in one case and $\Gamma < \Gamma_l$ in the other. This histogram was generated using 1,000 scenarios. Figure 3.1 suggests that robust optimization is more conservative than its nominal counterpart (limits upside potential) but decreases the downside risk.

Figures 3.2 and 3.3 show the number of iterations versus budget of uncertainty Γ_l for five different Γ values, for Data Sets 1 and 3, respectively. Recall that Data Set 1 has 10 projects and Data

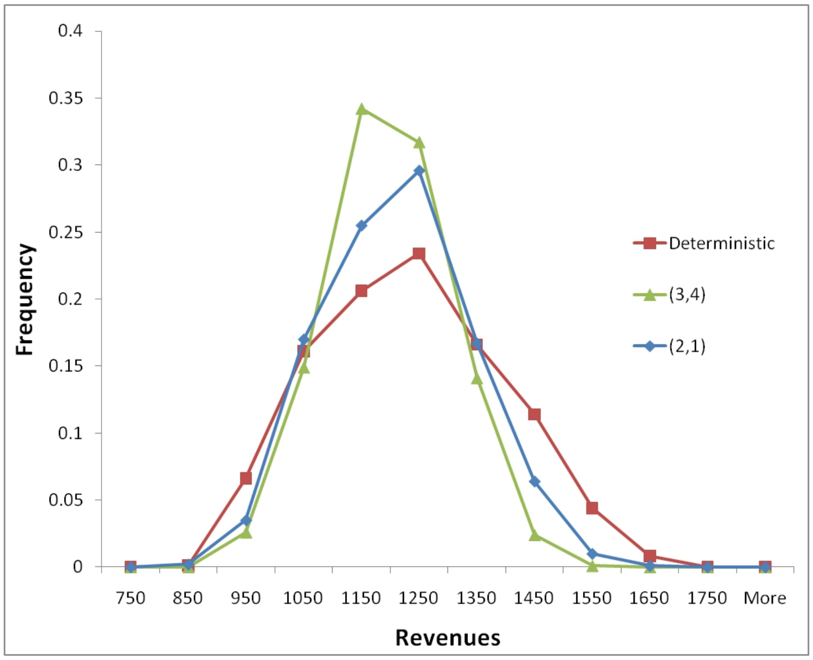


Figure 3.1: Histogram of Revenues.

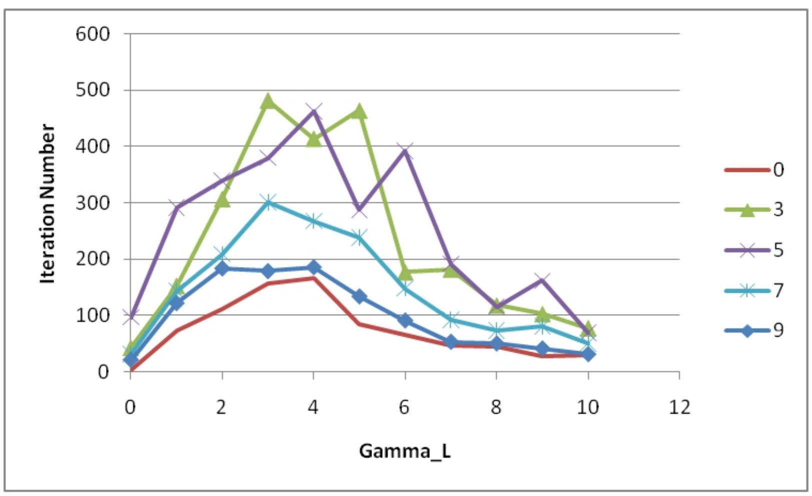


Figure 3.2: Number of Iterations versus Budget of Uncertainties for Data Set 1, Budget=500.

Set 3 has 20. (Our observations remain valid for other values of Γ, but the corresponding graphs were omitted for graph readability.) We observe that, for each Γ value, the number of iterations in the robust optimization models does not differ substantially from the number of iterations in the deterministic model when Γ_l is close to its bounds ($\Gamma_l = 0$ or $\Gamma_l = 10$), which means that most

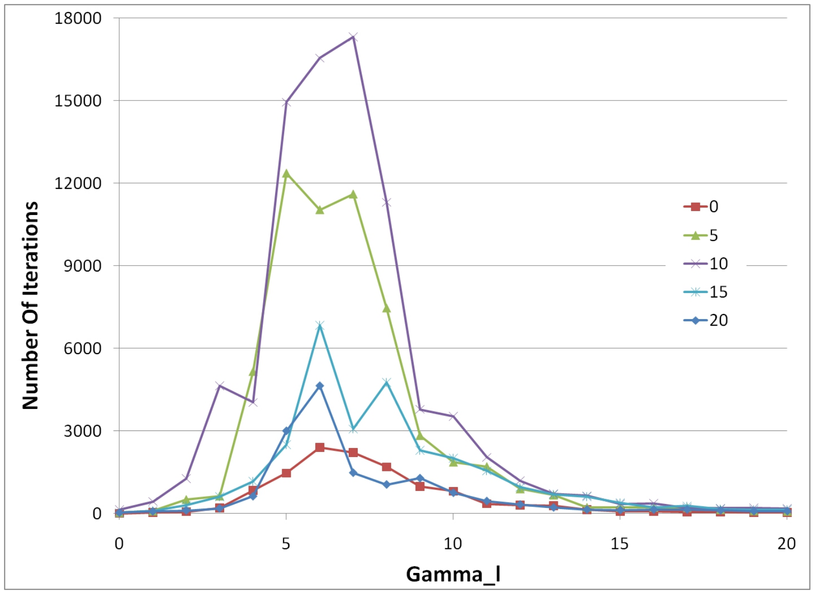


Figure 3.3: Number of Iterations versus Budget of Uncertainties for Data Set 3, Budget=1000.

projects are in the same uncertainty range.) When projects are more evenly assigned to low and high ranges (middle values of Γ_l), the number of iterations increases, sometimes substantially (see Figure 3.3, where the top curve corresponds to $\Gamma = 10$).

Since robust optimization maximizes the worst-case cash flow over the uncertainty set, it is natural to evaluate how well it protects against downside risk. To do that, we compute the first and fifth percentile of the distribution of the random objective where we have injected the optimal solution, for Data Set 1 and all (Γ, Γ_l) combinations, using 1,000 scenarios. These results are shown in Tables 3.1 and 3.2, respectively. Table 3.3 shows the expected value of the objective for reference. We see that robust optimization does indeed protect against downside risk, as evidenced in the increase in the values for the first and fifth percentile, with modest performance degradation (decrease in average objective value).

It is important to note that the optimal solution will not change once Γ or Γ_l increases past the number of projects being funded, which we will denote x. If p is the (estimated) probability of project cash flows falling in the low range, a decision-maker interested in protecting his cumulative cash flow against adverse events will select $\Gamma_l \geq p\,x$; however, x cannot be determined before the robust optimization problem has been solved (and depends somewhat on Γ and Γ_l, although the dependence is minimal in our experiments: the manager invests in 4 or 5 in all data sets with budget equal to 500, and 10 or 11 projects out of 20 in Data Sets 3 and 4 when the budget is equal to 1,000).

Therefore, we recommend that the decision-maker compute Tables 3.1, 3.2 and 3.3 for his own project selection problem, and choose an appropriate $(\Gamma,\, \Gamma_l)$ pair based on the tradeoff between downside risk (measured either by first or fifth percentile) and performance (measured by average objective) that he wishes to achieve. Also note that several $(\Gamma,\, \Gamma_l)$ pairs have the same optimal solution, due to the use of binary variables, and that what the manager ultimately needs to determine is the strategy he will implement, rather than a specific $(\Gamma,\, \Gamma_l)$ pair, which would only be used to compute the corresponding optimal strategy anyway. In the case of Data Set 1, we recommend to invest in projects 1, 3, 6, 8, 10; this strategy is optimal for $(\Gamma,\, \Gamma_l)$ pairs $(3, 4)$, $(4, 4)$, $(0, 3)$ and $(\Gamma, 3)$ for any $\Gamma \geq 5$. This choice maximizes both first and fifth percentiles over all possible $(\Gamma,\, \Gamma_l)$ combinations, achieving the biggest shift of the cumulative cash flow distribution to the right.

$\downarrow \Gamma\ \Gamma_l \rightarrow$	0	1	2	3	4	5-10
0	811.5	811.5	908.5	919.5	919	919
1	811.5	908.5	908.5	908.5	919	919
2	842.5	908.5	908.5	908.5	919	919
3	842.5	908.5	908.5	908.5	919.5	919
4	842.5	908.5	908.5	908.5	919.5	919
5-10	811.5	811.5	908.5	919.5	919	919

Table 3.1: First percentile values for each (Γ, Γ_l) pair with Data Set 1.

$\downarrow \Gamma\ \Gamma_l \rightarrow$	0	1	2	3	4	5-10
0	879.5	879.5	965	981	968	968
1	879.5	965	965	965	968	968
2	911	965	965	965	968	968
3	911	965	965	965	981	968
4	911	965	965	965	981	968
5-10	879.5	879.5	965	981	968	968

Table 3.2: Fifth percentile values for each (Γ, Γ_l) pair with Data Set 1.

$\downarrow \Gamma\ \Gamma_l \rightarrow$	0	1	2	3	4	5-10
0	1156.3	1156.3	1162.0	1148.1	1117.4	1117.4
1	1156.3	1162.0	1162.0	1162.0	1117.4	1117.4
2	1176.7	1162.0	1162.0	1162.0	1117.4	1117.4
3	1176.7	1162.0	1162.0	1162.0	1148.1	1117.4
4	1176.7	1162.0	1162.0	1162.0	1148.1	1117.4
5-10	1156.3	1156.3	1162.0	1148.1	1117.4	1117.4

Table 3.3: Expected revenue for each (Γ, Γ_l) pair with Data Set 1.

Heuristics.

Table 3.4 compares the objective function values of Method (a) (ranking according to densities) and Method (b) (ranking according to NPV) with the optimal objective function value. For this comparison, the development budget was taken equal to 500 in all data sets. # Opt. indicates the number of times the heuristics gives the same objective function value as the optimal solution when all possible (Γ, Γ_l) pairs are enumerated. We see that Method (b) generally performs better in terms of the number of times it finds the optimal value: it performs as well as Heuristic (a) for Data Set 2 and performs much better for the other three data sets. Highest performance is achieved for Data Set 3, where Heuristic (a) never found the optimal solution while Heuristic (b) had a success ratio of 76%.

	Method (a) vs Optimal		**Method (b) vs Optimal**	
	% Obj. Dif.	**# Opt.**	**% Obj. Dif.**	**# Opt.**
Data Set 1	3.28	15/121	2.12	90/121
Data Set 2	6.32	77/121	5.77	77/121
Data Set 3	1.82	0/441	3.27	336/441
Data Set 4	5.23	21/441	5.35	48/441

Table 3.4: Optimal objective function value versus heuristic results.

We now evaluate the optimal solution and the heuristic solutions. We generated 100 scenarios for the cash flows when the probability of falling into the low range is 0.5. We implemented the optimal and heuristic solutions of each data set for these scenarios (again, with a budget of 500 in all cases, to allow for easy comparison) and computed mean and standard deviation of the objective (cumulative discounted cash flows). Table 3.5 shows the average percentage (absolute) difference in mean and standard deviation of the simulation results over all (Γ, Γ_l) combinations for method (a) and method (b). We see that using the heuristics rather than the optimal solution does not significantly change the objective average, but does change the standard deviation more significantly. There was no sign pattern in the mean difference or standard deviation difference, which is why we only show absolute values.

	Method (a) vs Optimal		**Method (b) vs Optimal**	
	% |Dif.| (Mean)	**% |Dif.| (St. Dev.)**	**% |Dif.| (Mean)**	**% |Dif.| (St. Dev.)**
Data Set 1	0.96	9.60	0.86	8.77
Data Set 2	3.32	3.50	3.25	5.11
Data Set 3	3.92	28.76	2.53	7.01
Data Set 4	2.79	4.81	4.28	6.94

Table 3.5: Difference between simulated optimal solutions versus heuristic solutions.

3.5 Conclusions

We have presented an approach to robust optimization with multiple ranges for each uncertain coefficient, derived tractable exact reformulations and studied an application to R&D project selection. We have also provided a robust ranking heuristic and tested two possible ranking criteria: (a) according to project densities, and (b) according to project Net Present Values. Numerical experiments suggest that, while both heuristics exhibit good performance, Heuristic (b) performs better. The multi-range approach gives more flexibility to the decision-maker to specify how many coefficients can fall in each of the ranges and thus allows for a finer description of uncertainty within the robust optimization framework.

Chapter 4

Multi-Range Robust Optimization: Some Applications

This chapter illustrates some applications of multi range robust optimization. Project selection application in Chapter (3) was an example when uncertain parameters have underlying discrete random factors that affect the uncertainty. However, uncertainty does not always come from underlying discrete random factors. Sometimes multiple decision makers cause multiple uncertainty ranges. Sometimes we have some expert knowledge on the likelihoods of uncertainties to be realized. Our aim in this chapter of the thesis is to show some applications where we incorporate multi rance robust optimization to these cases.

4.1 Prioritizing Project Selection

This section describes a multi-range robust optimization approach applied to the problem of capacity investment under uncertainty. Our goal is to investigate the merits of an approach based on a concept called *multi-range robust optimization*, which was developed in Chapter (3), for the specific setting of project selection and prioritization presented in Koc et. al. [46]. We consider a number of possible projects with anticipated costs and cash flows, and an investment decision to be made under budget limitations. Uncertainty in anticipated parameter values – cost and net present value of each project in our case – could seriously damage the real-life viability of the suggested investment plan. We set up the multi-range robust optimization so that the possible values taken by the uncertain parameters match the three possible values of the cost or net present value distributions in the stochastic programming approach. While the stochastic programming approach implemented by Koc et. al. [46] suffers from tractability issues, the robust optimization approach solves the same capacity investment problem in seconds. We also show how to compute the project prioritization list to substantially decrease computation time.

The paper by Koc et. al. [46] was selected as the benchmark because the authors implement a stochastic programming framework to a real-life problem using real data, and the main "selling point" of robust optimization has long been that it is more tractable than stochastic programming in real-life applications. (Another selling point is related to the difficulty in estimating underlying probabilities correctly in the stochastic framework, but we will not consider this point here.) The main contribution in this chapter is to present multi-range robust optimization as a tractable alternative to stochastic programming when the budgets of uncertainty are set appropriately based on the probabilities of the stochastic programming model. A secondary contribution is that we show how to compute the project priority list in a far more efficient manner than what was proposed in Koc et. al. [46], thus substantially reducing computation times. When testing our approach using the problem setup provided in Koc et. al. [46], the stochastic programming approach does not solve to while our robust optimization approach solves its model to optimality within seconds.

The rest of the chapter is structured as follows. In Section 4.1.2, we describe how a simple change to the model implemented by Koc et. al. [46] will drastically improve the solution time of the stochastic programming problem by providing a more computationally efficient way of computing the project priority list. Section 4.2 presents the robust optimization formulation in the proposed setting, while Section 4.3 provides the details of the numerical implementation. Section 4.6 contains concluding remarks.

4.1.1 Problem Overview

Preliminaries

Our study is motivated by Koc et al. [46], who consider an investment problem with cost and NPV uncertainties. They use a company-made analysis of the projects, which is provided by South Texas Project Nuclear Operating Company and which lists the anticipated NPV and costs in three possible scenarios: pessimistic, optimistic and most likely cases. The analysis also categorizes the projects in two groups: low-risk and medium-risk. Koc et al. [46] compute a priority list so that the decision-maker can adjust immediately to changes in the budget (capacity) by implementing a greedy approach, i.e., she will go down the priority list selecting projects until capacity has been filled. In this paper we are interested in the approach called *Optimal Project Prioritization*, where Koc et al. [46] formulate an optimization model that incorporates budget, cost and profit scenarios and outputs an *optimal priority list*.

Our robust optimization approach differs from [46] at three levels.

- First, we develop a robust optimization model where we optimize the project portfolio performance and provide a robust priority list, which would be still viable under the worst cases of the cost and NPV outcomes as defined by our uncertainty set. [46], on the other hand, models the problem as a two-stage stochastic programming problem and maximizes the expected NPV of the selected costs calculated over predefined scenarios.

- We do not consider uncertainty on the right-hand side (budget) here and concentrate on the cost and NPV uncertainty assuming that the budget is given. If there were uncertainty on the RHS, robust optimization would require to assign the RHS its worst-case value. Therefore, budget uncertainty – if it is present in the formulation – will be addressed in the same manner as Koc et al. [46], using scenarios for different levels of the budget. In what follows, we assume that there is no budget uncertainty (both for our approach and our implementation of the Koc et al. [46] approach.)

- Finally, we do not make the assumptions on the behavior of the uncertain parameters that are made in [46]: we do not assume that the cost and NPV are perfectly correlated; furthermore, we do not assume that the projects in the same risk groups are perfectly correlated. We feel that our setting is more representative of real-life industry situations.

4.1.2 Improved Stochastic Formulation

Model

The notation and formulation of the optimal prioritization model in [46] are:

Indices and sets:

$i, i' \in I$	candidate projects
$p \in P$	priorities; $P = \{1, 2, \ldots \lvert I \rvert\}$
$t \in T$	time periods (years)
$\omega \in \Omega$	scenarios

Data:

a_i^ω	net present value of project i under scenario ω
b_t^ω	available budget in period t under scenario ω
c_{it}^ω	cost of project i in period t under scenario ω
q^ω	probability of scenario ω

Decision variables (binary):

x_i^ω	1 if project i is selected under scenario ω, 0 otherwise
$y_{i,i'}$	1 if project i has higher priority than i', 0 otherwise
z_{ip}	1 if project i is assigned priority level p, 0 otherwise

Formulation:

$$
\begin{aligned}
\max_{x,y,z} \quad & \sum_{\omega\in\Omega} q^{\omega} \sum_{i\in I} a_i^{\omega}\, x_i^{\omega} & & (a) \\
\text{s.t.} \quad & \sum_{i\in I} c_{i,t}^{\omega}\, x_i^{\omega} \le b_t^{\omega}, & t\in T, \omega\in\Omega \quad & (b) \\
& \sum_{i\in I} z_{i,p} = 1, & p\in P \quad & (c) \\
& \sum_{p\in P} z_{i,p} = 1, & i\in I \quad & (d) \\
& |P| y_{i,i'} \ge \sum_{p\in P} (|P|-p)(z_{i,p} - z_{i',p}), & i\neq i', i,i'\in I \quad & (e) \\
& y_{i,i'} + y_{i',i} = 1, & i < i', i,i'\in I \quad & (f) \\
& x_i^{\omega} \ge x_{i'}^{\omega} + y_{i,i'} - 1, & \omega\in\Omega, i\neq i', i,i'\in I \quad & (g) \\
& x_i^{\omega} \in \{0,1\} & i\in I, \omega\in\Omega \quad & (h) \\
& y_{i,i'} \in \{0,1\} & i\neq i', i,i'\in I \quad & (i) \\
& z_{i,p} \in \{0,1\} & i\in I, p\in P \quad & (j)
\end{aligned}
\tag{4.1}
$$

This formulation can be explained as follows:

Objective (a) The decision maker maximizes the expected NPV.

Constraints (b) The total cost cannot exceed the budget, in any given scenario.

Constraints (c) Each priority rank can only be assigned to one project.

Constraints (d) Each project can only be assigned to one priority rank.

Constraints (e) For any pair of projects (i, i'), if i' is assigned a lower priority than i then i is preferred to i'.

Constraints (f) For any pair of projects (i, i'), either i is preferred to i' or i' is preferred to i.

Constraints (g) For any pair of projects (i, i') and any scenario ω, if i' is selected in scenario ω and i is preferred to i', then i is selected in scenario ω as well.

Constraints (h),(i),(j) Decision variables are binary.

An important remark we made when we first attempted to implement the approach in Koc et al. [46] is that the decision variables $z_{i,p}$, which provide the priority level of the projects, are not necessary in the formulation. The essential knowledge – the pairwise comparisons of the projects' priorities – lies in the variable $y_{i,i'}$. This makes constraints (c),(d),(e) unnecessary. Note that the $|P|$ used in constraints (e) is nothing but a big-M constraint, which impairs the tightness of LP relaxations and increases the run times.

Further, we suggest the following changes for constraints (f) and (g):

$$\tilde{y}_{i,i'} \geq x_i^\omega - x_{i'}^\omega, \quad \forall \omega \in \Omega,\, i, i' \in I : i < i' \tag{4.2}$$

$$1 - \tilde{y}_{i,i'} \geq x_{i'}^\omega - x_i^\omega, \quad \forall \omega \in \Omega,\, i, i' \in I : i < i' \tag{4.3}$$

Note that we replace the variable $y_{i,i'}$ with $\tilde{y}_{i,i'}$, which is only defined for $i, i' \in I : i < i'$. We can set $\tilde{y}_{i,i'} = 0$ if $i \geq i'$ and drop them. If $x_i^\omega = 1$ and $x_{i'}^\omega = 0$, Eq. (4.2) forces $\tilde{y}_{i,i'} = 1$. Then Eq. (4.3) forces i to be preferred to i' for all ω. When the model is solved, the optimal $\tilde{y}_{i,i'}$ give us a two-by-two comparison of all variables. We can then build the priority list based on this information, because ($i \succ j$ and $j \succ k$) implies $i \succ k$. This statement is proved by noting that, by definition, $i \succ j$ means that if $x_i^\omega = 0$ in some scenario ω then $x_j^\omega = 0$ for the same scenario ω and there is at least one scenario ω for which $x_i^\omega = 1$ and $x_j^\omega = 0$. Similarly, $j \succ k$ means that if $x_j^\omega = 0$ in some scenario ω then $x_k^\omega = 0$ for the same scenario ω and there is at least one scenario ω for which $x_j^\omega = 1$ and $x_k^\omega = 0$. Combining the two statements yields the result immediately.

The new set of constraints described above are tight as they do not require a big-M constant. (In fact, $\tilde{y}_{i,i'}$ can even be relaxed to be in [0,1] for all pairs of projects.) Instead of Model (4.1), we can then solve the following problem as a more computationally efficient stochastic programming problem:

$$
\begin{aligned}
\max_{x,y} \quad & \sum_{\omega\in\Omega} q^{\omega} \sum_{i\in I} a_i^{\omega}\, x_i^{\omega} && \\
\text{s.t.} \quad & \sum_{i\in I} c_{i,t}^{\omega}\, x_i^{\omega} \le b_t^{\omega}, && t\in T, \omega\in\Omega \\
& y_{i,i'} \ge x_i^{\omega} - x_{i'}^{\omega}, && \omega\in\Omega, i<i', i,i'\in I \\
& 1 - y_{i,i'} \ge x_{i'}^{\omega} - x_i^{\omega}, && \omega\in\Omega, i<i', i,i'\in I \\
& x_i^{\omega} \in \{0,1\} && i\in I, \omega\in\Omega \\
& y_{i,i'} \in \{0,1\} && i\ne i', i,i'\in I
\end{aligned}
\tag{4.4}
$$

4.1.3 Implementation

We solve Koc et al. [46]'s model (4.1) and our suggested prioritizing model (4.4) using ILOG CPLEX version 12.1 for the full-size problem data given in Koc et al. [46]. Both problems hit the time limit which is set to 100,000 seconds. However, our suggested problem solution was at 0.13% of optimality, while Koc et al. [46]'s solution was at 37.20% of optimality. [46] does not provide run time statistics but only state the model was ultimately solved within 1% of optimality. Figure 4.1 shows the optimality gap and objective function values for both problems with respect to simplex iterations. As it is seen from the figure, revised stochastic problem quickly reduces the optimality gap to within 1% of optimality. Although Problem (4.1)'s initial lower bound (around 47) is bigger, it finds it later than the time when revised model improves the lower bound to above 60. Solving Model (4.4) gives us the y variables, from which we derive a priority list for the projects in seconds. We compare the two models for two subproblems provided in [46], when the number of projects is 10 and 15, respectively. For both subproblems we get the same objective function value and the same x^{ω} in all scenarios. There were some differences in the priority lists of the two models. The fact that there were some pairwise reversed priorities can be explained by the existence of ties between projects, which the computer breaks arbitrarily.

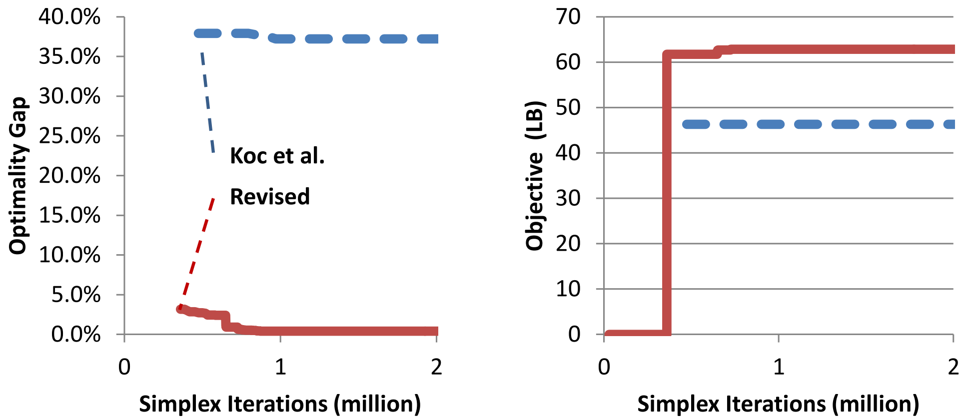


Figure 4.1: Comparison of Koc et al. [46]'s model with its revised model (4.4)

4.2 The Multi-Range Robust Optimization Model

4.2.1 High-Level Modeling

The stochastic programming problem provided by Koc et al. [46] does not provide any optimal solution within a reasonable time frame. While our stochastic model (4.4) performs significantly better, we feel that the run times still raise issues in terms of large-scale tractability of the stochastic approach. Therefore, in this section, we derive the multi-range robust counterpart of Problem (4.4). The formulation will be solved in the next section to demonstrate the potential of robust optimization in terms of solution time and quality.

The approach proposed by Düzgün and Thiele [37] enables us to incorporate all the possible values that uncertain parameters can take into the optimization problem, and thus addresses the limitations of the traditional robust optimization framework. Specifically here, an uncertain parameter will be allowed to take any of the pessimistic, most likely or optimistic values. We have two uncertainty ranges for each uncertain NPV and cost parameter: *low* and *high*. Figure 4.2 summarizes how we construct our low and high uncertainty ranges for the NPV parameters. For cost parameters, the place of optimistic and pessimistic values will be switched, so that the optimistic value for a cost will be the worst-case value of the low range.

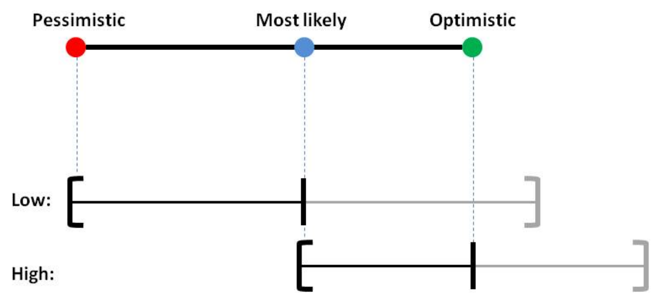


Figure 4.2: Construction of low and high ranges for the uncertain NPV parameters

The intervals are defined by using the fact that at optimality, the uncertain parameters in the robust optimization approach with two ranges will take one of four possible values:

i. The nominal value of the low range,

ii. The nominal value of the high range,

iii. The worst-case value of the low range,

iv. The worst-case value of the high range.

Because we only want three values, we will define the uncertainty intervals so that the nominal value of the low range coincides with the worst-case value of the high range. Again, it is not possible in traditional one-range robust optimization to consider three possible values for the data given. With the help of multi-range robust optimization approach, we are able to construct the uncertainty sets such that we can incorporate these multiple values and yet obtain a robust solution without having to consider many scenarios.

Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the uncertainty sets for NPV factors and cost factors, respectively. The robust problem that we are going to solve has the structure of Problem (5.1) below:

$$\begin{aligned} \max_{x} \quad & \min_{\mathbf{npv} \in \mathcal{P}_1} \mathbf{npv}\,\mathbf{x} \\ \text{s.t.} \quad & \max_{\mathbf{c} \in \mathcal{P}_2} \mathbf{c}\,\mathbf{x} \leq \mathbf{B} \\ & \mathbf{x} \in \{0,1\}^n \end{aligned} \tag{4.5}$$

4.2.2 Inner Optimization Problems

Inner minimization problem for NPVs.

We assign separate budgets of uncertainty for low-risk projects (with superscript $\mathcal{L}$) and medium-risk projects (with superscript $\mathcal{M}$) because projects in those groups have different probabilities of attaining their pessimistic, most likely and optimistic values. Superscript or subscript l denotes low-range coefficients, while superscript or subscript h denotes high-range coefficients. The rest of the notation is identical to that in Section 4.1.1.

$$\begin{aligned} \min_{u^l, u^h, y} \quad & \sum_{i=1}^{n} x_i \left[\overline{NPV}_i^l \, y_i^l - \widehat{NPV}_i^l \, u_i^l + \overline{NPV}_i^h \, y_i^h - \widehat{NPV}_i^h \, u_i^h \right] \\ \text{s.t.} \quad & u_i^l \leq y_i^l, && \forall i \in I, \\ & u_i^h \leq y_i^h, && \forall i \in I, \\ & y_i^l + y_i^h = 1, && \forall i \in I, \\ & \sum_{i \in \mathcal{L}} y_i^l \leq \Gamma_l^{\mathcal{L}}, \\ & \sum_{i \in \mathcal{M}} y_i^l \leq \Gamma_l^{\mathcal{M}}, \\ & \sum_{i \in \mathcal{L}} (u_i^l + u_i^h) \leq \Gamma^{\mathcal{L}}, \\ & \sum_{i \in \mathcal{M}} (u_i^l + u_i^h) \leq \Gamma^{\mathcal{M}}, \\ & y_i^j \in \{0,1\}, && \forall i \in I, \, \forall j \in \{l,h\}, \\ & u_i^l, u_i^h \geq 0, && \forall i \in I. \end{aligned} \tag{4.6}$$

Inner maximization problems for cost factors.

In year t, we have:

$$
\begin{aligned}
\max_{u^l,u^h,y} \quad & \sum_{i=1}^{n} x_i \left[\overline{c}^l_{i,t}\, y^l_i - \widehat{c}^l_{i,t}\, u^l_{i,t} + \overline{c}^h_{i,t}\, y^h_i - \widehat{c}^h_{i,t}\, u^h_{i,t} \right] & \\
\text{s.t.} \quad & u^l_{i,t} \le y^l_i, & \forall i \in I, \\
& u^h_{i,t} \le y^h_i, & \forall i \in I, \\
& y^l_i + y^h_i = 1, & \forall i \in I, \\
& \sum_{i \in \mathcal{L}} y^l_i \ge \Gamma^{\mathcal{L}}_l, & \\
& \sum_{i \in \mathcal{M}} y^l_i \ge \Gamma^{\mathcal{M}}_l, & \\
& \sum_{i \in \mathcal{L}} (u^l_{i,t} + u^h_{i,t}) \ge \Gamma^{\mathcal{L}}, & \\
& \sum_{i \in \mathcal{M}} (u^l_{i,t} + u^h_{i,t}) \ge \Gamma^{\mathcal{M}}, & \\
& y^j_i \in \{0,1\}, & \forall i \in I,\ \forall j \in \{l,h\}, \\
& u^l_{i,t}, u^h_{i,t} \ge 0, & \forall i \in I.
\end{aligned}
\tag{4.7}
$$

Note that we have greater-than-or-equal-to constraints for the uncertainty budgets because the inner problem is now a maximization problem. Thus, we will have exactly $\Gamma^{\mathcal{L}}_l$ and $\Gamma^{\mathcal{M}}_l$ values in the low range among low-risk and medium-risk projects, respectively. Similarly, exactly $\Gamma^{\mathcal{L}}$ low-risk and $\Gamma^{\mathcal{M}}$ medium-risk projects will take the worst case values in the range they fall into.

From Düzgün and Thiele [37], we know that the constraint sets of Problems (4.6) and (4.7) are totally unimodular. Therefore, we can relax the integrality of the y variables and still obtain an integer optimal solution, given that the right-hand-sides of the constraints are integer. This allows us to use strong duality and convert Problem (5.1) into one large maximization problem.

4.2.3 The Formulation

The **objective function** of our robust optimization problem comes from the objective function of the dual problem of Problem (4.6):

$$\max \qquad \sum_{i\in I} p_i - \sum_{i\in I}\left(z_i^l + z_i^h\right) - \Gamma_l^{\mathcal{L}}\,\gamma_l^{\mathcal{L}} - \Gamma_l^{\mathcal{M}}\,\gamma_l^{\mathcal{M}} - \Gamma^{\mathcal{L}}\,\gamma^{\mathcal{L}} - \Gamma^{\mathcal{M}}\,\gamma^{\mathcal{M}}$$

The constraints of the dual problem are added to the constraint set of our robust optimization problem. Dual constraints associated with variables y_i^l and y_i^h for low-risk and medium-risk projects are:

$$\begin{aligned}
p_i^l + p_i - \gamma_l^{\mathcal{L}} - z_i^l &\le \overline{NPV}_i^l\,x_i \quad && i\in\mathcal{L}\\
p_i^h + p_i - z_i^h &\le \overline{NPV}_i^h\,x_i \quad && i\in\mathcal{L}\\
p_i^l + p_i - \gamma_l^{\mathcal{M}} - z_i^l &\le \overline{NPV}_i^l\,x_i \quad && i\in\mathcal{M}\\
p_i^h + p_i - z_i^h &\le \overline{NPV}_i^h\,x_i \quad && i\in\mathcal{M}
\end{aligned}$$

Similarly, dual constraints associated with variables u_i^l and u_i^h for low-risk and medium-risk projects are:

$$\begin{aligned}
p_i^l + \gamma^{\mathcal{L}} &\le \widehat{NPV}_i^l\,x_i \quad && i\in\mathcal{L}\\
p_i^h + \gamma^{\mathcal{L}} &\le \widehat{NPV}_i^h\,x_i \quad && i\in\mathcal{L}\\
p_i^l + \gamma^{\mathcal{M}} &\le \widehat{NPV}_i^l\,x_i \quad && i\in\mathcal{M}\\
p_i^h + \gamma^{\mathcal{M}} &\le \widehat{NPV}_i^h\,x_i \quad && i\in\mathcal{M}
\end{aligned}$$

For the **uncertain cost parameters**, we have a maximization problem in the constraint set of Problem (5.1) but invoke strong duality and thus insert the dual problem of Problem (4.7) into our robust counterpart problem. The objective function of the dual of Problem (4.7) will form our new budget constraint for $t \in T$:

$$\sum_{i \in I} cp_{i,t} + \sum_{i \in I} \left(cz^l_{i,t} + cz^h_{i,t} \right) + c\Gamma^{\mathcal{L}}_l \, c\gamma^{\mathcal{L}}_{l,t} + c\Gamma^{\mathcal{M}}_l \, c\gamma^{\mathcal{M}}_{l,t} + c\Gamma^{\mathcal{L}} \, c\gamma^{\mathcal{L}}_t + c\Gamma^{\mathcal{M}} \, c\gamma^{\mathcal{M}}_t \leq B(t)$$

Then, we will have the dual constraints to be added to the robust counterpart problem. The dual constraints corresponding to y^l_i and y^h_i for low-risk and medium-risk projects in Problem (4.7) are:

$$\begin{aligned}
-cp^l_{i,t} + cp_{i,t} + c\gamma^{\mathcal{L}}_{l,t} + cz^l_{i,t} &\geq \overline{c}^l_{i,t} \, x_i && i \in \mathcal{L}, t \in T \\
-cp^h_{i,t} + cp_{i,t} + cz^h_{i,t} &\geq \overline{c}^h_{i,t} \, x_i && i \in \mathcal{L}, t \in T \\
-cp^l_{i,t} + cp_{i,t} + c\gamma^{\mathcal{M}}_{l,t} + cz^l_{i,t} &\geq \overline{c}^l_{i,t} \, x_i && i \in \mathcal{M}, t \in T \\
-cp^h_{i,t} + cp_{i,t} + cz^h_{i,t} &\geq \overline{c}^h_{i,t} \, x_i && i \in \mathcal{M}, t \in T
\end{aligned}$$

Similarly, the dual constraints associated with variables u^l_i and u^h_i in Problem (4.7) for low-risk and medium-risk projects are:

$$\begin{aligned}
cp^l_{i,t} + c\gamma^{\mathcal{L}}_t &\leq -\widehat{c}^l_{i,t} \, x_i && i \in \mathcal{L}, t \in T \\
cp^h_{i,t} + c\gamma^{\mathcal{L}}_t &\leq -\widehat{c}^h_{i,t} \, x_i && i \in \mathcal{L}, t \in T \\
cp^l_{i,t} + c\gamma^{\mathcal{M}}_t &\geq -\widehat{c}^l_{i,t} \, x_i && i \in \mathcal{M}, t \in T \\
cp^h_{i,t} + c\gamma^{\mathcal{M}}_t &\geq -\widehat{c}^h_{i,t} \, x_i && i \in \mathcal{M}, t \in T
\end{aligned}$$

In addition to these constraints, we have the constraints that were originally in the problem before reformulation and sign constraints:

$$\begin{aligned}
&x_i \in \{0,1\}^n, && i \in I, \\
&p^l_i, p^h_i, cp^l_{i,t}, cp^h_{i,t} \geq 0, && i \in I, t \in T, \\
&z^l_i, z^h_i, cz^l_{i,t}, cz^h_{i,t} \geq 0, && i \in I, t \in T, \\
&\gamma^{\mathcal{L}}_l, \gamma^{\mathcal{M}}_l, \gamma^{\mathcal{L}}_l, \gamma^{\mathcal{M}}_l \geq 0, && i \in I, t \in T,
\end{aligned}$$

The **complete formulation** is given by:

$$
\begin{aligned}
\max \quad & \sum_{i\in I} p_i - \sum_{i\in I}\left(z_i^l + z_i^h\right) - \Gamma_l^{\mathcal{L}}\,\gamma_l^{\mathcal{L}} - \Gamma_l^{\mathcal{M}}\,\gamma_l^{\mathcal{M}} - \Gamma^{\mathcal{L}}\,\gamma^{\mathcal{L}} - \Gamma^{\mathcal{M}}\,\gamma^{\mathcal{M}} \\
\text{s.t.} \quad & \max_{\mathbf{c}\in\mathcal{P}_2} \mathbf{c}'\mathbf{x} \le \mathbf{B} \\
& p_i^l + p_i - \gamma_l^{\mathcal{L}} - z_i^l \le \overline{NPV}_i^l\, x_i && i\in\mathcal{L} \\
& p_i^h + p_i - z_i^h \le \overline{NPV}_i^l\, x_i && i\in\mathcal{L} \\
& p_i^l + \gamma^{\mathcal{L}} \le \widehat{NPV}_i^l\, x_i && i\in\mathcal{L} \\
& p_i^h + \gamma^{\mathcal{L}} \le \widehat{NPV}_i^h\, x_i && i\in\mathcal{L} \\
& p_i^l + p_i - \gamma_l^{\mathcal{M}} - z_i^l \le \overline{NPV}_i^l\, x_i && i\in\mathcal{M} \\
& p_i^h + p_i - z_i^h \le \overline{NPV}_i^l\, x_i && i\in\mathcal{M} \\
& p_i^l + \gamma^{\mathcal{M}} \le \widehat{NPV}_i^l\, x_i && i\in\mathcal{M} \\
& p_i^h + \gamma^{\mathcal{M}} \le \widehat{NPV}_i^h\, x_i && i\in\mathcal{M} \\
& \mathbf{x}\in\{0,1\}^n.
\end{aligned}
\tag{4.8}
$$

Note that we no longer have any y binary variable establishing pairwise priorities because determining an appropriate priority order is straightforward once we have obtained the optimal solution: any order that ranks the selected ones above the non-selected ones will work. Our robust model is a deterministic model and finds a single portfolio unlike the stochastic programming model, which finds separate portfolios for different scenarios but a single ordering for all. Imposing a single priority order in that problem is, therefore, meaningful in the stochastic programming problem but redundant in the robust optimization one, since a priority can be inferred from the optimal solution. Therefore, for the multi-range problem we solve only the Knapsack problem without prioritizing projects (Model (5.1)) and obtain the priority list through post-processing.

4.3 Numerical Study

4.3.1 Setup

We follow the setup described in Koc et. al. [46] and have 26 low-risk projects and 15 medium-risk projects. In the stochastic programming approach, the cost and NPV of a low-risk project are assigned the pessimistic value with probability $\frac{1}{6}$, the optimistic value with probability $\frac{1}{6}$, and the most likely value with probability $\frac{4}{6}$. For medium-risk projects these three probabilities become $\frac{2}{6}$, $\frac{1}{6}$ and $\frac{3}{6}$, respectively. We use these probabilities to determine the budgets of uncertainty. On the average 4 or 5 projects out of 26 low-risk projects ($\frac{26}{6} = 4.33$) would take the pessimistic values. Similarly, 4 or 5 of them would take the optimistic values. 5 out of 15 medium-risk projects ($\frac{15 \cdot 2}{6} = 5$) would take the pessimistic values, 2 or 3 of them ($\frac{15}{6} = 2.5$) would take the optimistic values. Because the ranges are constructed so that the nominal value of the low range and the worst-case value of the high range coincide, we have one degree of freedom in setting the parameters.

We have 10 possible budgets: from \$2.5M to \$7M, in increments of \$0.5M. For each of these possible budgets, we solve Model (5.1) for three model settings, which can be seen on Table 4.1. The nominal model is the model where all NPV and cost components will take the *most likely* values.

Parameters		Model Setting		
Project	Model	Nominal	Robust1	Robust2
NPV	$\Gamma_l^{\mathcal{L}}$	26	8	13
	$\Gamma_l^{\mathcal{M}}$	15	5	7
	$\Gamma^{\mathcal{L}}$	0	18	13
	$\Gamma^{\mathcal{M}}$	0	13	11
Cost	$\Gamma_l^{\mathcal{L}}$	26	14	13
	$\Gamma_l^{\mathcal{M}}$	15	8	8
	$\Gamma^{\mathcal{L}}$	0	8	13
	$\Gamma^{\mathcal{M}}$	0	4	4

Table 4.1: Uncertainty budget combinations.

In the *Robust 1* combination, we have 8 low-risk projects in the low range, and the remaining

18 low-risk projects will be in the high range. We want 4 low-risk project to take the pessimistic values, in line with the probabilities mentioned above. Then, 4 of 8 projects in low range should be able to deviate from the nominal value. Also, 4 projects might take the optimistic values. It suffices to assume that 18-4 = 14 projects can deviate from the optimistic value (nominal value of the high range) and be at the lowest value of the high range, which is also the *most likely* value. Thus, we set $\Gamma^{\mathcal{L}} = 4 + 14 = 18$. Other values and the value in *Robust 2* are also assigned in a similar manner, recalling that we have one degree of freedom to set the budget parameters.

4.3.2 Results

Our focus is to determine whether the multi-range robust optimization approach has potential as a computational alternative to stochastic programming for this real-life problem. Figure 4.3 compares the objective function values for Nominal and Robust 1 settings. Right most column on the figure shows the expected value of the objective function over all budget scenarios. Circle on the rightmost column is the stochastic model solution reported by Koc et al. [46]. The triangle indicates the stochastic model solution (4.1) when the solver hit the time limit at 100,000 seconds. We see that our expected objective function value is very close to the given stochastic model's. We were able to reflect the expert knowledge given by the company and got robust solutions which are not very conservative. Moreover, robust optimization problem gives an optimal solution in less than a second, while in stochastic case, we could not get the optimal solution in a reasonable time frame. Tables 4.2, 4.3 and 4.4 display the objective function and model statistics for each budget values for three model settings Nominal, Robust 1 and Robust 2, respectively. The expected NPV is also given for each setting at the last row of the tables. Expected values are calculated using the probabilities of each budget and objective of that budget. We see that the nominal setting has the highest expected return, as expected. The Robust 1 and Robust 2 settings yield an expected NPV as 61.46% and 61.33%, respectively. [46] reports an expected NPV of 60.18%. We observe that our model solves to optimality in less than a second. Koc et. al. [46] does not report their model's statistics. Furthermore, if we do add constraints, such as constraints that create a priority list as part of the

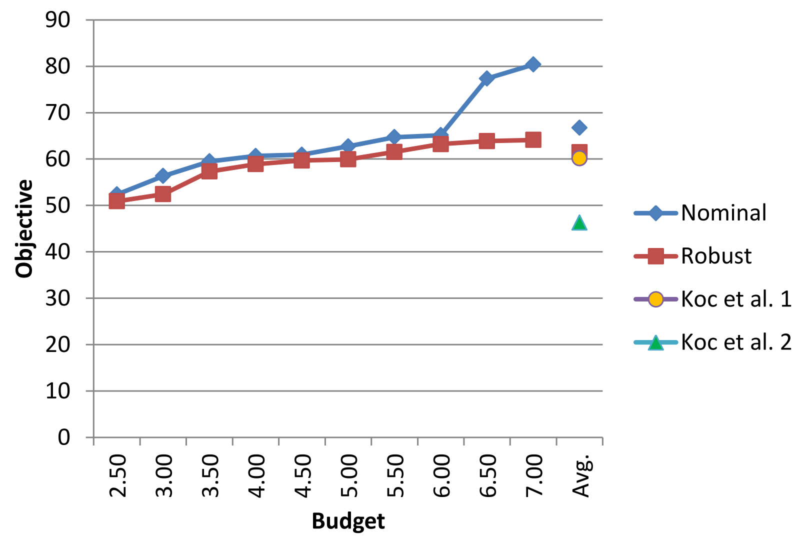


Figure 4.3: Objective function values of nominal and robust solutions for different budget scenarios.

Budget	Objective	Time (sec)	Iterations	Nodes
2.5	52.38	0.51	699	50
3.0	56.34	0.39	618	15
3.5	59.51	0.30	604	12
4.0	60.66	0.23	645	16
4.5	60.95	0.31	653	10
5.0	62.73	0.27	630	29
5.5	64.71	0.20	613	12
6.0	65.16	0.25	631	21
6.5	77.34	0.37	639	20
7.0	80.38	0.22	578	5
Average	**66.75**			

Table 4.2: Model results for *Nominal* uncertainty budget combinations

optimization problem, our solution time is only of the order of 1-2 seconds, as shown in Table 4.5.

Budget	Objective	Time (sec)	Iterations	Nodes
2.5	50.91	0.17	538	42
3.0	52.43	0.38	708	67
3.5	57.31	0.21	464	11
4.0	58.89	0.22	435	12
4.5	59.69	0.11	431	8
5.0	59.92	0.19	436	9
5.5	61.53	0.17	460	31
6.0	63.22	0.22	399	6
6.5	63.88	0.21	382	6
7.0	64.13	0.20	412	16
Average	**61.46**			

Table 4.3: Model results for *Robust 1* uncertainty budget combinations.

Budget	Objective	Time (sec)	Iterations	Nodes
2.5	50.82	0.27	636	44
3.0	52.31	0.40	1050	108
3.5	57.19	0.41	681	13
4.0	58.76	0.18	547	21
4.5	59.56	0.25	452	8
5.0	59.78	0.26	477	9
5.5	61.40	0.26	538	37
6.0	63.09	0.44	634	10
6.5	63.73	0.26	377	5
7.0	63.97	0.21	485	19
Average	**61.33**			

Table 4.4: Model results for *Robust 2* uncertainty budget combinations.

4.4 Robust Pricing

4.4.1 Introduction

Firms discount for many reasons. To increase sales and profit or to attract customers are among these reasons. Increasing profit might be in the form of reducing loss here. The relationship between price and demand of a good leads to classification of goods into substitutes or complements. If a company increases the price of a substitutable good, it is going to loose some of its customers to a competitor. In other words, an increase in price will result in an increase in demand for its substitute goods. Therefore, companies may choose to go to discounts on prices of these good. In

Budget	Objective	Time (sec)	Iterations	Nodes
2.5	50.82	1.16	1725	30
3.0	52.31	1.48	3354	81
3.5	57.19	0.75	1379	15
4.0	58.76	0.75	1667	17
4.5	59.56	0.54	1582	11
5.0	59.78	0.53	1367	8
5.5	61.40	2.42	2449	43
6.0	63.09	1.04	1472	19
6.5	63.73	1.11	1245	17
7.0	63.97	0.62	1089	6
Average	**61.33**			

Table 4.5: Model with prioritization results for *Robust 2* uncertainty budget combinations.

this chapter, we consider this pricing problem that firms face from a robust optimization point of view and formulate mathematical models that rely on our multi-range robust optimization, which was explained in Chapter 3.

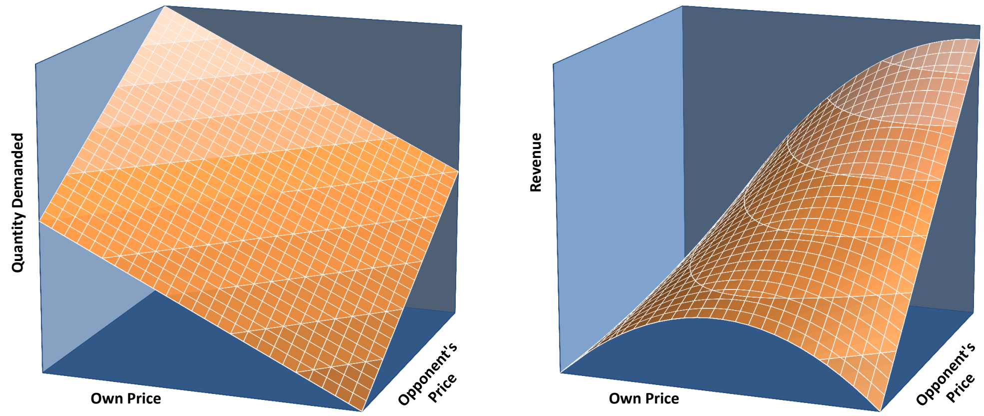
We can divide this chapter into two parts: In the first part there are n substitutable goods we are selling and we have a one competitor that sells the same goods. We do not know how much discount our competitor will do and our demand depends on our price and our competitor's price. We are trying to set our price such that our revenue is maximized. In the second part of the section, we are selling n substitutable goods and we have m competitors. One important point here is that we do not assume strategic response on the part of the competitors. We are interested in a worst case analysis.

4.4.2 N Goods, 1 Competitor

We mentioned that our goal is to maximize our revenue. Let linear market demand be represented by:

$$D_i = a_i - \beta_i p_i + \alpha_i q_i + \epsilon_i$$

where β_i and α_i are the sensitivity of the demand function for a given price for good i and q_i is our competitor's price of good i. D_i is demand we see for our good i and ϵ_i is the uncertainty on our



h1

Figure 4.4: Change in demand and revenue with respect to price change

demand that is affected by competitors price. In Figure (4.4), we see how our demand and revenue is affected by our price and our competitors price. Our objective function is to maximize our revenue, which is

$$\max_{p_i} \quad \sum_{i=1}^{n} D_i p_i = \max_{p_i} \sum_{i=1}^{n} (a_i - \beta_i p_i) p_i + \sum_{i=1}^{n} (\alpha_i q_i + \epsilon_i) p_i$$

Competitors' selling prices are uncertain. Assume that we expect the competitor would have $|K|$ possible values to discount and for each price setting, there is some uncertainty on how our demand will be affected. Competitors low price would decrease our sales. Then, our robust counterpart problem can be defined as:

$$\max_{p_i} \left[\sum_{i=1}^{n} (a_i - \beta_i p_i) p_i + \min_{y,z \in \mathcal{Q}} \sum_{i=1}^{n} p_i \sum_{k=1}^{|K|} (\alpha_i \, q_{i,k} \, y_{i,k} - \epsilon_{i,k} \, z_{i,k}) \right] \tag{4.9}$$

where $y \in \{0, 1\}$ and $0 \leq z \leq 1$. $q_{i,k}$ is the the selling price of good i in k^{th} price setting or range. $\epsilon_{i,k}$ indicates the uncertain effect of this price setting on demand of i.

Only one discount setting can be realized at the same time. Therefore we should have

$$\sum_{k=1}^{|K|} y_{i,k} = 1 \tag{4.10}$$

in our constraints set. Moreover, total deviation (uncertainty) is expected to be less than Γ, which is expressed by the equation

$$\sum_{i=1}^{n} \sum_{k=1}^{|K|} z_{i,k} \leq \Gamma \tag{4.11}$$

and we also assume that Γ_k products is expected to be sold at price $q_{i,k}$. That is:

$$\sum_{i=1}^{n} y_{i,k} \leq \Gamma_k \tag{4.12}$$

Equations (4.10), (4.11) and (4.12) forms the constraints set of $\mathcal{Q}$. We can formulate the inner minimization problem as:

$$\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} p_i \sum_{k=1}^{|K|} (\alpha_i \, q_{i,k} \, y_{i,k} - \epsilon_{i,k} \, z_{i,k}) & \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^{n} y_{i,k} \leq \Gamma_k, & \forall k \\
& \textstyle\sum_{i=1}^{n} \sum_{k=1}^{|K|} z_{i,k} \leq \Gamma & \\
& \textstyle\sum_{k=1}^{|K|} y_{i,k} = 1, & \forall i \\
& z_{i,k} \leq y_{i,k}, & \forall i, k \\
& y \in \{0, 1\} & \\
& 0 \leq z \leq 1 &
\end{aligned} \tag{4.13}$$

As we have proved in section (3), the constraint set of the problem is Totally Unimodular. We can use strong duality theorem to re-formulate our robust optimization problem.

We insert the dual of Problem (4.13) into the robust counterpart problem and get the following robust pricing model:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} a_i p_i - \sum_{i=1}^{n} \beta_i p_i^2 + \sum_{i=1}^{n} \eta_i - \sum_{i=1}^{n} \sum_{k=1}^{|K|} z_{i,k} - \sum_{k=1}^{|K|} \gamma_k \, \Gamma_k - \Gamma \, \gamma_0 \\
\text{s.t.} \quad & \pi_{i,k} + \gamma_0 \geq p_i \, \epsilon_{i,k}, && \forall i, k, \\
& \pi_{i,k} + \eta_i - \gamma_k - z_{i,k} \leq p_i \, \alpha_i \, q_{i,k}, && \forall i, k, \\
& p_i, \gamma^k, \gamma_0, \pi_{i,k}, z_{i,k} \geq 0 && \forall i, k.
\end{aligned} \tag{4.14}$$

4.4.3 N Goods, M Competitors

In the previous section, we were considering only one competitor. Here, we assume that we have M competitor and our goal is to set price of N substitutable goods such that it will maximize our revenue. Our demand function then becomes:

$$D_i = a_i - \beta_i p_i + \sum_{j=1}^{m} (\alpha_{i,j} q_{i,j} + \epsilon_{i,j})$$

Competitor j can decide on how much to discount on product i:

$$q_{i,j} = \sum_{k=1}^{|K|} q_{i,j,k} \, y_{i,j,k} \tag{4.15}$$

Thus, our robust counterpart problem in this setting is:

$$\max_{p_i} \left[\sum_{i=1}^{n} (a_i - \beta_i p_i) p_i + \min_{y,z \in \mathcal{Q}} \sum_{i=1}^{n} p_i \sum_{j=1}^{m} \sum_{k=1}^{|K|} (\alpha_{i,j} q_{i,j,k} \, y_{i,j,k} - \epsilon_{i,j,k} \, z_{i,j,k}) \right] \tag{4.16}$$

Normal price range of the product is also among these $|K|$ ranges. Therefore, we say that only one range (k) should be realized for all goods and for all competitors. That is:

$$\sum_{k=1}^{|K|} y_{i,j,k} = 1 \qquad \forall i, j \tag{4.17}$$

In order to avoid over conservativeness, we assume that a product i will be sold at price level k at most by $\Gamma_{i,k}^{Comp}$ competitors:

$$\sum_{j=1}^{m} y_{i,j,k} \leq \Gamma_{i,k}^{Comp} \qquad \forall i, k \tag{4.18}$$

We think that parameters $\Gamma_{i,k}^{Comp}$ can be computed using historical data on sold products. In addition to (4.18) and (4.17), we also have a budget constraint on the total deviation around the expected demand:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{|K|} z_{i,j,k} \leq \Gamma \tag{4.19}$$

All these constraints (4.17, 4.18, 4.19) constitutes our inner minimization problem (4.20) and they construct a totally unimodular constraint matrix.

$$\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} p_i \sum_{j=1}^{m} \sum_{k=1}^{|K|} (\alpha_{i,j} q_{i,j,k}\, y_{i,j,k} - \epsilon_{i,j,k}\, z_{i,j,k}) & \\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{m} y_{i,j,k} \leq \Gamma_{i,k}^{Comp}, & \forall i, k \\
& \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{|K|} z_{i,j,k} \leq \Gamma & \\
& \textstyle\sum_{k=1}^{|K|} y_{i,j,k} = 1, & \forall i, j \\
& z_{i,j,k} \leq y_{i,j,k}, & \forall i, k \\
& y \in \{0, 1\} & \\
& 0 \leq z \leq 1 &
\end{aligned} \tag{4.20}$$

With the replacement of inner minimization model in Problem (4.16) with its dual problem, we get our robust optimization problem (4.21):

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} \alpha_i p_i - \sum_{i=1}^{n} \beta_i p_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{m} \eta_{i,j} - \\
& \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{|K|} z_{i,j,k} - \sum_{i=1}^{n}\sum_{k=1}^{|K|} \gamma_{i,k}\, \Gamma_{i,k}^{Comp} - \Gamma\, \gamma_0 \\
\text{s.t.} \quad & \pi_{i,j,k} + \gamma_0 \geq p_i\, \epsilon_{i,j,k} && \forall i,j,k, \\
& \pi_{i,j,k} + \eta_{i,j} - \gamma_{i,k}^{Comp} - z_{i,j,k} \leq p_i\, \alpha_{i,j}\, q_{i,j,k}, && \forall i,j,k, \\
& p_i, \gamma_{i,k}^{Comp}, \gamma_0, \pi_{i,j,k}, z_{i,j,k} \geq 0 && \forall i,j,k.
\end{aligned} \tag{4.21}$$

4.5 Numerical Example

4.5.1 N Goods, 1 Competitor

For the N Goods, 1 Competitor setting, we have price levels for N products and we only consider 1 competitor. Our competitor can have k different price setting for a product, representing different sale or promotion levels. In this case, we have 10 products and each product has 4 price levels. We first randomly generated the market price of the products, then we generated 4 different sale prices. Not all the price levels are smaller then the market prices. We also set one lever that is higher than the market price level. We assume β_i is 1 for all products and we randomly generated α values between 0.5 and 2. We found a values by taking the derivative of the revenue function and equating it to zero after replacing competitor prices by the randomly generated market values.

We wrote the model in GAMS and solved it using the solver CONOPT. Figure (4.5) shows the histogram of revenues compared for the nominal case (when the competitor matches the market prices for all products, that is we assume known competitor prices) and the robust case. In robust case, we have a Γ value that indicates the number of products that will appear in a given price level. We see that both histogram plots are very close to each other. Nevertheless, robust solution has higher revenue in the lower percentiles. For example, robust solutions offers 0.3 and 0.2 percent

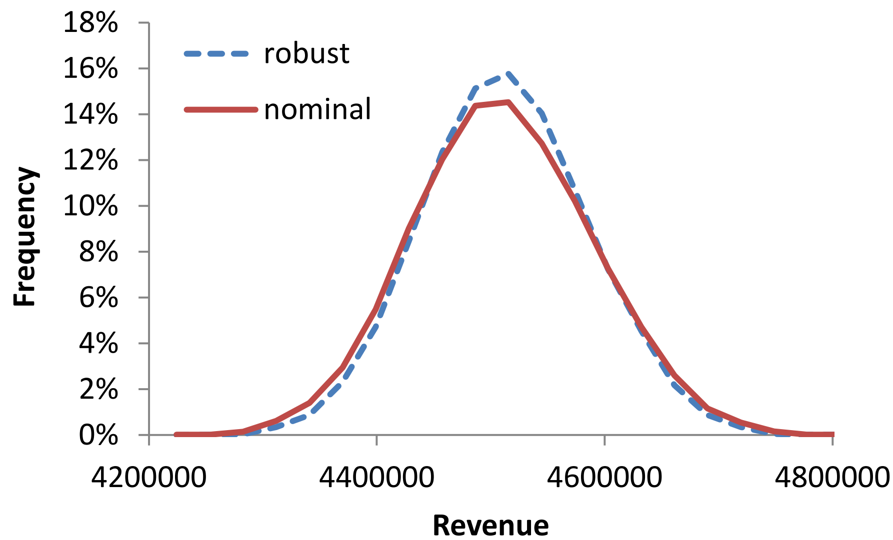


Figure 4.5: Histogram of revenues in N Goods 1 Competitor setting

higher revenue at 5th and 10th percentiles. This increase is important depending on the size of revenue. Moreover, note that this increase is the result of setting prices and it does not incur extra cost.

Figure (4.6) displays our and competitor's prices for each product. Small circles represents the price levels for the products. We see that we mostly set prices to their low values and most of the time we either match or be under the competitors price. For products 3 and 10 however, our prices are higher than the competitor's. We found the robust price that maximized our revenue, and sometimes, this may correspond to higher prices.

4.5.2 N Goods, M Competitors

In this case we have N goods and M competitors. Each competitor has different price levels for their each product. We have 10 goods, 10 competitors and 3 price levels. All parameters are randomly generated except a, which is found by taking the derivative of the revenue function and setting it to zero. Figure (4.7) shows the histogram of revenues. Robust problem offers 0.7%, 0.5% and 0.4%

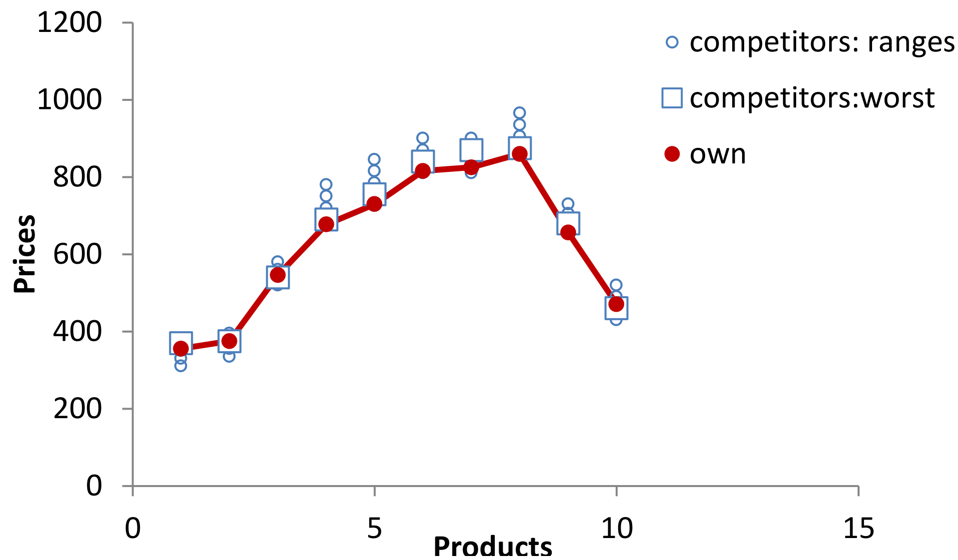


Figure 4.6: Price matching in N Goods 1 Competitor setting

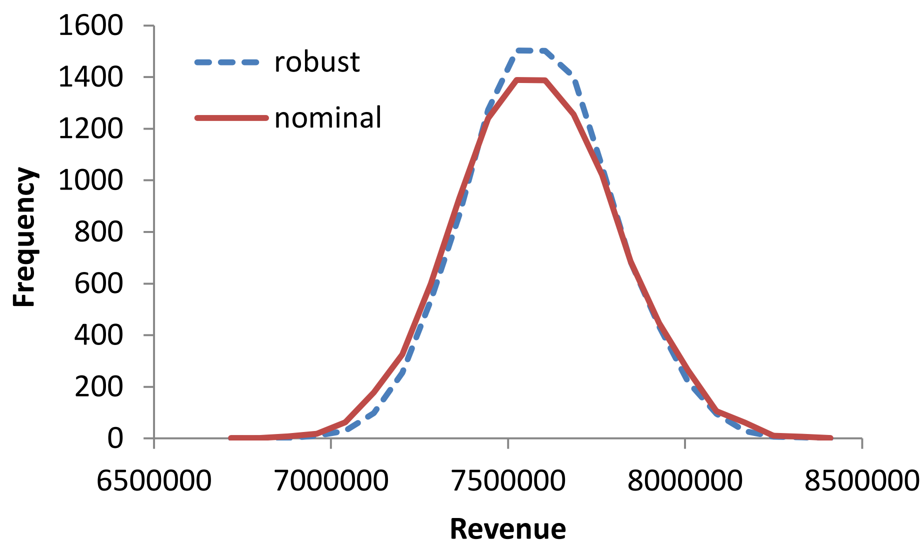


Figure 4.7: Histogram of revenues in N Goods M Competitor setting

higher revenue at 1st, 5th and 10th percentiles, respectively. Figure (4.8) displays the competitors's and ours price for the optimal robust solution. For clarity of the graph, we did not display the

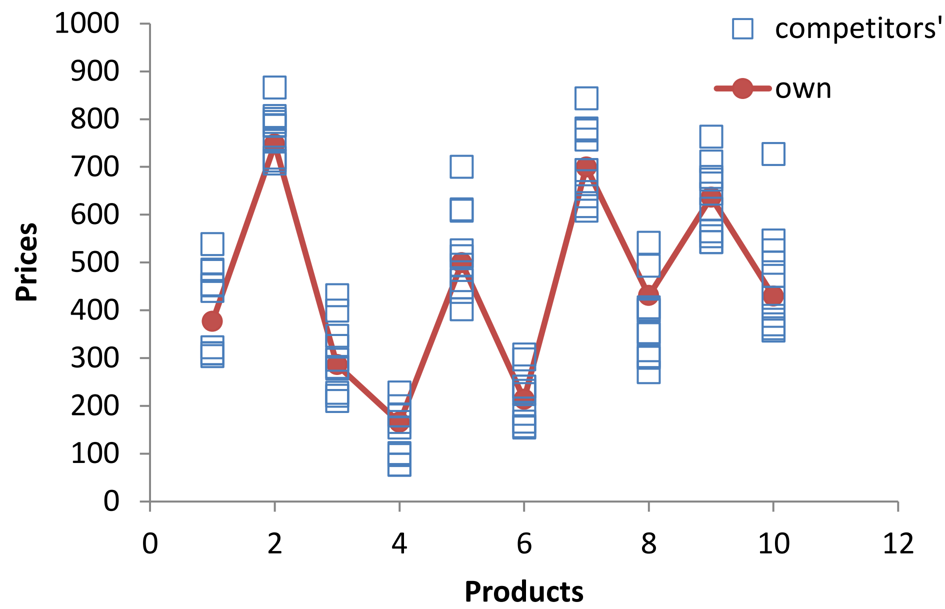


Figure 4.8: Price matching in N Goods M Competitor setting

possible price levels for each competitor and product. The squares represents the competitors' price and the circles are our price.

4.6 Conclusion

In this section, we show two possible applications of multi range robust optimization. First, we have shown how to implement multi-range robust optimization as a tractable alternative to stochastic programming, by selecting the budgets of uncertainty appropriately to match (the rounded values of) the expected number of times that the uncertain parameters will take their optimistic, most likely, pessimistic values. We have also shown how to improve solution time of the stochastic programming approach by using post-processing. Numerical results are very encouraging.

Second, we have shown how to apply multi range robust optimization on pricing decisions, when our demand depends on a linear function of our price and our competitors' price. We have shown two examples; where we have N goods and 1 competitor, and N goods and M competitors. In both cases, we were able to increase our revenue slightly compared to deterministic approach.

Chapter 5

Robust Optimization with Chance Constraints

5.1 Introduction

So far, we have used robust optimization techniques to represent the uncertainties in parameters; specifically, we have extended the robust optimization approach with polyhedral sets of Bertsimas and Sim [24] to the case where parameters can belong to disjoint ranges and the number of parameters that can belong to a type of range (e.g., low or high) is bounded. Intuitively, polyhedral uncertainty sets are more appealing in robust integer optimization because robust counterparts of linear problems remain linear (Bertsimas and Sim [23]), while ellipsoidal sets lead to nonlinear formulations (Ben-Tal and Nemirovski [9]), with clear drawbacks in an integer framework. On the other hand, it is legitimate to ask whether some of these sets offer a close connection to real-life randomness, and in particular probabilistic statements, which would further strengthen the relevance of the robust optimization methodology.

Providing an intuitive interpretation of uncertainty sets has always been of importance to operations researchers: for instance, Bertsimas and Sim [23] connects the choice of a key parameter in their approach called the budget of uncertainty with a probability of constraint violation, which has played a significant role in the adoption of their approach by practitioners. More recently, Ben-Tal et. al. [6] have described in detail a process where a "safe tractable approximation" of probabilistic constraints leads to a robust optimization approach where the uncertainty set is determined by the chosen approximation and the probability level.

Safe tractable approximations, the most famous of which is the Bernstein approximation, are motivated by the fact that incorporating a chance constraint to a problem usually creates significant computational difficulties if the random variables do not obey a Normal distribution, as it requires multivariate integration within the optimization problem. Ben-Tal et. al. [6]'s idea is that the probabilistic constraint should be replaced by a more tractable constraint that, when satisfied, guarantees that the original probabilistic statement is satisfied too (hence, is "safe"). While we will study the case of jointly Normal distribution for comparison purposes, a key assumption of our setup is that we do not know precisely the underlying distributions of the random variables, so we will study tractable bounds which approximate the chance constraints, using the safe tractable approximation framework. Our goal in the present paper is to investigate the theoretical and algorithmic insights we gain from this approach in the special case where decision variables are binary.

The contributions of this paper are as follows:

- We show that the safe tractable approximation (called Bernstein approximation) to binary optimization problems is equivalent to a deterministic problem with modified cost coefficients, which only depend on problem data and one extra coefficient.

- We consider two cases: (i) when the uncertain parameters obey a jointly Normal distribution, which allows us to demonstrate the insights we can gain in the simplest setting when we know, in closed form, both the distributions of the uncertain parameters and of the objective function, (ii) when we only know the first two moments and the support of the distributions of the uncertain parameters. Our conclusions are valid for both.

- We investigate an iterative approach to address the risk of over-conservatism of the safe tractable approximation approach, which arises because the methodology uses Markov's Inequality for non-negative random variables, admittedly an unsophisticated bound connecting tail percentile with expected value.

- We compare our approach in numerical experiments with the one proposed by Bertsimas and Sim [23], also for binary optimization problems with uncertain coefficients but for a different modeling of uncertainty when probability distributions are not known, and argue that, while solution quality is comparable, the solution times in our approach are substantially smaller.

The remainder of this chapter is structured as follows: in Section 5.2, we investigate the problem-specific formulations and properties that result from having binary decision variables in the one-range case, while Section 5.5 extends our framework to the case with multiple ranges. Finally, Section 5.6 contains concluding remarks.

5.2 The Safe Tractable Approximation

Consider the following maximization problem where the objective function parameters are uncertain:

$$\begin{aligned} \max \quad & \mathbf{c}'\mathbf{x} \\ \text{s.t.} \quad & x \in \mathcal{X} \subseteq \{0,1\}^n, \end{aligned} \tag{5.1}$$

Because the vector $\mathbf{c}$ is not known precisely, our goal here will be to maximize the greatest parameter A such that:

$$P\left(\sum_{i=1}^{n} c_i x_i < A\right) \leq \epsilon, \tag{5.2}$$

for $0 < \epsilon < 1/2$. We will assume at first that the random coefficients are independent, and will relax this assumption in Section 5.5.

When distributions are continuous, A can be interpreted as the ϵ-quantile of $\mathbf{c}'\mathbf{x}$. We use a strict inequality in Eq. (5.2) to better align ourselves with the approach description in Ben-Tal et. al. [6].

5.2.1 Special Case: Normally Distributed Parameters

Exact Reformulation

Theorem 5.1 (Exact Reformulation for Gaussian uncertainty) *Assume $c \sim \mathcal{N}(\mu, \Sigma)$. Then Problem (5.1) can be reformulated as:*

$$\begin{aligned} \max \quad & \mu^T x + \Phi^{-1}(\epsilon)||\Sigma^{\frac{1}{2}} x||_2 \\ \textit{s.t.} \quad & x \in \mathcal{X} \subseteq \{0,1\}^n. \end{aligned} \tag{5.3}$$

Proof. Since $c \sim \mathcal{N}(\mu, \Sigma)$, we have $c'x \sim \mathcal{N}(\mu^T x, x^T \Sigma x)$. Eq. (5.2) can then be reformulated as:

$$\Phi\left(\frac{A - \mu^T x}{\sqrt{x^T \Sigma x}}\right) \leq \epsilon.$$

This yields:

$$\mu^T x - A \geq -\Phi^{-1}(\epsilon)||\Sigma^{\frac{1}{2}} x||_2. \tag{5.4}$$

The greatest such A is then:

$$A_{\max} = \mu^T x + \Phi^{-1}(\epsilon)||\Sigma^{\frac{1}{2}} x||_2,$$

which yields Problem (5.3). This is a nonlinear binary problem, for which we discuss solution techniques in Section 5.2.1.

Bernstein Approximation

We can use the exact reformulation (5.3) to gain insights into the quality of the Bernstein approximation framework described in Ben-Tal et. al. [6]. Specifically, we have:

Theorem 5.2 (Bernstein Approximation for Gaussian uncertainty) *A lower bound on the optimal objective of Problem (5.1) is provided by the optimal objective of:*

$$
\begin{aligned}
\max_{\theta \geq 0} \frac{\ln \epsilon}{\theta} + \max_{x} \quad & \sum_{i=1}^{n} \left(\mu_i - \frac{1}{2}\theta\sigma_i^2 \right) x_i \\
\text{s.t.} \quad & x \in \mathcal{X} \subseteq \{0,1\}^n,
\end{aligned}
\tag{5.5}
$$

which, at θ given, is a binary optimization problem.

Proof. In line with Ben-Tal et. al. [6], Eq. (5.2) can be written as, with $\theta > 0$:

$$
P\left(-\theta \sum_{i=1}^{n} c_i x_i > -\theta A \right) = P\left(\exp\left\{ -\theta \sum_{i=1}^{n} c_i x_i \right\} > \exp\{-\theta A\} \right) \leq \epsilon. \tag{5.6}
$$

Since the exponential function is a nonnegative and nondecreasing function, we can invoke Markov's Inequality, leading to:

$$
P\left(\exp\left\{ -\theta \sum_{i=1}^{n} c_i x_i \right\} > \exp\{-\theta A\} \right) \leq \frac{E[\exp\{-\theta \sum_{i=1}^{n} c_i x_i\}]}{\exp\{-\theta A\}}. \tag{5.7}
$$

Under the assumption that the random parameters are independent, the right-hand side of Eq. (5.7) can be reformulated as:

$$
\exp\{\theta A\} \prod_{i=1}^{n} E[\exp\{-\theta c_i x_i\}] \quad = \exp\{\theta A\} \prod_{i=1}^{n} \exp\left\{ -\theta \mu_i x_i + \frac{1}{2}\theta^2 \sigma_i^2 x_i \right\},
$$

where we have used the expression of the moment generating function for Gaussian random variables and the fact that x_i is binary for all i, so that $x_i^2 = x_i$ for all i.

Therefore, it is sufficient for Eq. (5.2) to be satisfied to have:

$$
\theta A + \sum_{i=1}^{n} \left(-\theta \mu_i x_i + \frac{1}{2}\theta^2 \sigma_i^2 x_i \right) \leq \ln \epsilon. \tag{5.8}
$$

Reinjecting the upper bound on A yields Problem (5.5).

Comments:

- The robust model (5.5) can be interpreted as a deterministic problem with modified cost coefficients with only one extra parameter, θ, and no new constraints.

- The coefficients in Problem (5.5) decrease from their nominal value by an amount proportional to the parameter variance, rather than standard deviation.

- The advantage of Problem (5.5) is that it is linear and thus more tractable than Problem (5.3) due to its structure. It is legitimate, however, to ask how good (tight) of an approximation this leads to, since Markov's inequality is a very simple bound. We explore this topic in Section 5.2.3.

Solution Approaches

We can solve Problem (5.5) in several different ways:

i. as a mixed integer nonlinear problem.

ii. as a sequence of integer linear problems, by solving iteratively and updating θ values. We start (step 0) with $\theta = \theta_0$ (small but positive); at each step, we solve Problem (5.5) at θ given, obtaining the optimal solution x^k. Then, using the first derivative condition at x^k given and using the fact that $f(\cdot, x^k)$ is concave, we derive the next θ^{k+1} value in closed form:

$$\theta^{k+1} = \sqrt{\frac{-2\ln\epsilon}{\sum_i \sigma_i^2 x_i^k}}.$$

We inject θ^{k+1} back into Problem (5.5) and find the new solutions x^{k+1}. We continue until we get the same solution in two consecutive iterations. While we cannot formally prove that this scheme converges to the global optimal solution, it converges very rapidly to the solution we get when we solve the problem as a mixed integer nonlinear problem and as a piecewise linear problem.

iii. as a piecewise linear approximation to the original problem, by considering a piecewise linear approximation to $\dfrac{\ln(\epsilon)}{\theta}$ and modeling the full problem as a mixed integer linear problem. Assume we have m linear sections and s_j is the slope of section $j = 1, \ldots, m$. (We can increase the tightness of the approximation by increasing the number of pieces, m.) Model (5.9), explained below, is a mixed integer approximation to Problem (5.5).

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{m} s_j u_j + \frac{\ln(\epsilon)}{\overline{\theta}_{min}} + \sum_{i=1}^{n} \mu_i x_i - \frac{1}{2}\sum_{i=1}^{n} y_i \sigma_i^2 \\
\text{s.t.} \quad & x \in \mathcal{X} \subseteq \{0,1\}^n, \\
& y_i - \theta \geq -M(1 - x_i), && \forall i \\
& \theta = \sum_{j=1}^{m} u_j + \overline{\theta}_{min} \\
& 0 \leq u_j \leq \overline{\theta}_j - \overline{\theta}_{j-1}, && \forall j \\
& \theta \geq 0, \\
& y_i \geq 0 && \forall i.
\end{aligned}
\tag{5.9}
$$

$\overline{\theta}_j$ indicates the breakpoints of linear sections. $\overline{\theta}_{min}$ is the smallest value θ can take. We replaced $\theta \cdot x_i$ by y_i in the objective function to linearize. M in the constraints indicates a large value. For instance, we can assign $M = \sqrt{\dfrac{-2\ln(\epsilon)}{\sigma_{min}}}$ with σ_{min} the smallest variance in the data set. When $x_i = 1$, y_i will take the value of θ. When $x_i = 0$, y_i will take value 0 since we are maximizing and y has a negative sign in the objective function.

Example. We solved Problem (5.5) as a nonlinear problem, as a piecewise mixed integer problem (Problem (5.9)) and iteratively using randomly generated data with different sizes for a knapsack problem. The data is generated using normal or uniform generation functions with different mean and standard deviations. Table (5.1) shows the sizes of each different data sets and their respective solution time for each solution method. For all data sets the solutions converged to the same (x, θ) at each solution method. We used CPLEX for the piecewise MIP and iterative MIP models and BARON for nonlinear model, We observe that, in every instance we generated, iterative model finds

	n	Nonlinear	Piecewise	Iterative
Data Set 1	80	0.74	0.19	0.05
Data Set 2	100	2.12	0.27	0.06
Data Set 3	100	0.58	0.22	0.16
Data Set 4	100	2.89	0.61	0.33
Data Set 5	200	3.03	0.23	0.17
Data Set 6	300	1431	0.50	0.25
Data Set 7	400	81.18	0.34	0.19
Data Set 8	500	3696.10	0.50	0.18

Table 5.1: Average solution time in seconds

the optimal θ in the second iteration and total time elapsed for both iterations is less than the solution time for both the nonlinear and piecewise models.

5.2.2 General Case: Formulation Based on Moment Information

We now are interested in deriving a deterministic tractable counterpart to the binary optimization problem with binary optimization when only a limited amount of information is known: the mean, distribution and support of each uncertain parameter. We use the linear semi-infinite optimization approach of Bertsimas and Popescu ([21], [22]) in order to find bounds on the right-hand side of Eq. (5.7) after invoking the independence of the coefficients. Therefore, we are interested in a (tight) bound for $E[\exp\{-\theta c\}]$ for all probability distributions of given mean, variance and support.

Theorem 5.3 *The random parameter c comes from a symmetric uncertainty set with $\bar{c} = \mu, \hat{c} = m\,\sigma$ for $m > 0$. Then,*

$$\max_{f\in\pi} E_f[\exp\{-\theta c\}] = \exp\{-\theta\bar{c}\}\exp\{-\theta\hat{c}\}\left(1 + \theta\hat{c} + \frac{(m^2+1)}{m^2}\frac{\theta^2\hat{c}^2}{2}\right). \tag{5.10}$$

Proof. The expected value of c with probability distribution function $f(c)$ is given by

$$E[c] = \int_{c^-}^{c^+} cf(c)dc$$

Then, we have

$$E[\exp\{-\theta c\}] = \int_{c^-}^{c^+} \exp\{-\theta c\} f(c) dc$$

The upper bound problem of $E[\exp\{-\theta c\}]$ can be formulated as the following optimization problem:

$$\begin{aligned} \max \quad & \textstyle\int_{c^-}^{c^+} \exp\{-\theta c\} f(c) dc \\ \text{s.t.} \quad & \textstyle\int_{c^-}^{c^+} f(c) dc = 1 \\ & \textstyle\int_{c^-}^{c^+} c f(c) dc = \mu \\ & \textstyle\int_{c^-}^{c^+} c^2 f(c) dc = \mu^2 + \sigma^2 \\ & f(c) \geq 0, \quad \forall c \in [c^-, c^+] \end{aligned} \tag{5.11}$$

μ and $\mu^2 + \sigma^2$ are the first two moments of the distribution. We write the dual of Problem (5.11):

$$\begin{aligned} \min \quad & \alpha + \mu\beta + (\mu^2 + \sigma^2)\gamma \\ \text{s.t.} \quad & \alpha + c\beta + c^2\gamma \geq \exp\{-\theta c\} \quad \forall c \in [c^-, c^+] \end{aligned} \tag{5.12}$$

This is a minimization problem with "greater-than-or-equal-to" inequality constraints. We look for a feasible solution that will make our bound as tight as possible. Let the constraint will be tight at c^- and $\tilde{c}$, where $\tilde{c} \in [c^-, c^+]$ and $\tilde{c} = \bar{c} + a\hat{c}$ where $-1 \leq a \leq 1$. The slopes of the two functions $\alpha + c\beta + c^2\gamma$ and $\exp\{-\theta c\}$ will be equal either at c^- or at $\tilde{c}$. Thus:

$$\alpha + c^-\beta + (c^-)^2\gamma = \exp\{-\theta c^-\}$$

$$\alpha + \tilde{c}\beta + (\tilde{c})^2\gamma = \exp\{-\theta \tilde{c}\}$$

Case 1: The tangency condition at c^- is written as:

$$\beta = -\theta \exp\{-\theta c^-\} - 2c^-\gamma$$

We write α and β in terms of γ and θ and inject them in the objective function to get the objective function when the constraint is tight at lower bound;

$$Obj_{c^-} = \exp\{-\theta c^-\} - \theta\hat{c}\exp\{-\theta c^-\} + \frac{m^2+1}{m^2}\hat{c}^2\gamma$$

Case 2: When the functions are tangent at $\tilde{c}$, we obtain:

$$Obj_{\tilde{c}} = \exp\{-\theta\tilde{c}\} + \theta\hat{c}\exp\{-\theta\tilde{c}\} + \frac{m^2a^2+1}{m^2}\hat{c}^2\gamma$$

We get the following objective function when we insert the value of γ in terms of c^- and $\tilde{c}$;

$$Obj = \exp\{-\theta\bar{c}\}\exp\{-\theta a\hat{c}\}\{1 + \theta a\hat{c} + \frac{m^2a^2+1}{m^2}\hat{c}^2\gamma\}$$

Since we are minimizing the dual problem (5.12), we take an a value that minimizes the function above, which is $a = 1$.

The objective function at the lower point of the range will provide the tightest bound for our probability, yielding Eq. (5.10).

Let define the function F as:

$$F_\theta(m, \widehat{c}) = \left(1 + \theta\hat{c} + \frac{(m^2+1)}{m^2}\frac{\theta^2\hat{c}^2}{2}\right), \tag{5.13}$$

so that:

$$\max_{f\in\pi} E_f[\exp\{-\theta c\}] = \exp\{-\theta c^+\}\, F_\theta(m, \widehat{c}).$$

Theorem 5.4 (Robust Problem with Bernstein Approximation) *The robust counterpart of Problem (5.1) is:*

$$\begin{aligned} \max_{\theta\geq 0}\frac{\ln\epsilon}{\theta} + \quad \max \quad & \sum_{i=1}^{n}\left(c_i^+ - \frac{1}{\theta}\ln\, F_\theta(m, \widehat{c}_i)\right)x_i \\ \textit{s.t.} \quad & x \in \mathcal{X} \subseteq \{0,1\}^n. \end{aligned} \tag{5.14}$$

where F is defined in Eq. (5.13) and $c_i^+ = \overline{c}_i + \widehat{c}_i$ is the upper bound of the uncertainty range.

Proof. Follows directly from injecting Eq. (5.10) of Theorem 5.3 into Eq. (5.7):

$$\exp\{\theta A\} \prod_{i=1}^{n} E[\exp\{-\theta c_i x_i\}] \leq \exp\{\theta A\} \prod_{i=1}^{n} \exp\{-\theta c_i^+ \, x_i\} \, F_\theta(m, \widehat{c}_i \, x_i) \leq \epsilon. \tag{5.15}$$

Our new objective function becomes (using that the x_i are binary and $F_\theta(m, 0) = 1$ for all m):

$$f(\theta) = \frac{\ln \epsilon}{\theta} + \frac{1}{\theta} \sum_{i}^{n} \left[\theta c_i^+ - \ln \, F_\theta(m, \widehat{c}_i)\right] x_i.$$

Problem (5.14) is a mixed integer nonlinear problem and it is a hard problem to solve in a reasonable time frame. It suffers from computational difficulties when we try to optimize in x and θ simultaneously, so we solve Problem (5.14) as a line search problem by computing the optimal objective for iterated values of θ. An appealing feature of this problem is that for a given θ, the problem becomes a mixed integer linear problem and it provides the insight that the robust optimization problem is a nominal problem with modified objective coefficients, specifically, the nominal coefficients are shifted by an amount proportional to $F_\theta(m, \widehat{c}_i)$ for all i.

5.2.3 An Alternative Formulation for the Knapsack Problem

We mentioned that when the uncertain parameters comes from Normal distribution, we can write the exact formulation of the uncertainty as in (5.3). In order to test the quality of our approximation method, after solving Problem (5.5) we computed the objective function of Problem (5.3) and compared the objective function values. The table below show the objective function values with respect to each model and the percentage difference between the values. We see that although our approximation has a close value, due to approximation technique, we might be more conservative than what we actually want to be. Therefore, we provide an alternative formulation for the problem and have a better control on the conservativeness of our solutions. When the underlying problem is a knapsack problem with a single constraint, interpreted as a project selection problem with a

	Approximation	**Exact Formulation**	**% Difference**
Data Set 1	5447	5597	2.68
Data Set 2	22068	22380	1.40
Data Set 3	28855	29199	1.18
Data Set 4	29772	30122	1.16

Table 5.2: Objective function values of Model (5.5)–(B&S) and (5.3)–(Exact Formulation)

budget constraint, we can approach the problem from a different perspective and try to minimize the required investment spending. In this case we will have the chance constraint in the constraint set. The threshold A we do not want to exceed will be given. This allows us to study the quality of the Bernstein approximation and adjust the values of ϵ_M in the mathematical formulation, so that the real protection threshold ϵ is achieved without over-conservatism.

Our problem becomes:

$$\begin{aligned} \min \quad & B \\ \text{s.t.} \quad & P\left(\sum_{i=1}^{n} c_i x_i < A\right) \leq \epsilon \\ & \sum_{i=1}^{n} CD_i x_i \leq B \\ & x_i \in \{0,1\}, \forall i. \end{aligned} \tag{5.16}$$

The main conclusion is that we can use a higher (sometimes substantially so) uncertainty parameter ϵ_M in the mathematical model to achieve a good protection level (the actual probability of falling below the threshold A, i.e., the actual ϵ). This motivates the following procedure: solve the mathematical problem, compute the resulting actual ϵ (protection level), update the model parameter $\epsilon_M \geq \epsilon$ (decreasing it if the actual ϵ is higher than desired).

Gaussian case

For the case when we assume the NPV obey a jointly Gaussian distribution, Problem (5.16) can be approximated (using the Bernstein framework) as:

$$\begin{aligned}
\min \quad & B \\
\text{s.t.} \quad & \frac{\ln \epsilon}{\theta} + \sum_{i=1}^{n} \mu_i x_i - \frac{1}{2}\theta \sum_{i=1}^{n} \sigma_i^2 x_i \geq A \\
& \sum_{i=1}^{n} CD_i x_i \leq B \\
& x_i \in \{0,1\} \\
& \theta \geq 0.
\end{aligned} \tag{5.17}$$

The solution methods applied to Problem (5.5) can be applied to Problem (5.17), as well.

If the distribution is Normal, we can compute the actual (realized) protection level after solving Problem (5.17) to optimality as follows:

$$\epsilon_R = \Phi\left(\frac{A - \sum_i^n \mu_i x_i}{\sqrt{\sum_i^n \sigma_i^2 x_i}}\right).$$

According to the actual tolerances we compute, we might need to iterate over ϵ to reduce the over conservativeness of the model. For example, Table 5.3 shows how different ϵ_M values affect the optimal solution and what real tolerances we obtain for that given ϵ_M. Bernstein approximation framework is protecting the constraint more than desired and it yields over-conservatism. If we set $\epsilon_M = 0.05$ the realized tolerance is 0.00605, which is far smaller than what the decision-maker was trying to plan for. For tolerances 0.05 or 0.01, $\epsilon_M = 0.3$ and 0.05 can be used, respectively. By using a larger ϵ_M, we get smaller objective functions, which means that we are using less budget. As you can see in Figure 5.1, as we become less conservative, as in case $\epsilon_M = 0.3$, we minimize the required budget. Revenues we get from the $\epsilon_M = 0.3$ solution has a mean of 1042.9 and standard deviation 63.2, while the $\epsilon_M = 0.05$ solution hase 1074.91 mean and 62.8 standard deviation.

ϵ_M	0.0004	0.006	0.05	0.08	0.1	0.3	0.4
ϵ_R	0.00001	0.00048	0.00605	0.01085	0.01441	0.04286	0.05925
i=1	1	0	1	1	1	1	1
i=2	1	0	0	1	0	1	1
i=3	1	0	1	0	1	0	0
i=4	0	1	0	1	0	0	0
i=5	1	1	0	1	0	1	0
i=6	1	1	1	0	1	0	1
i=7	0	0	0	0	0	0	0
i=8	0	1	1	0	0	1	0
i=9	0	0	0	0	0	0	0
i=10	0	0	0	0	1	0	1
Obj.	516	445	413	410	403	390	445

Table 5.3: Tolerance computations for different ϵ_M values: realized tolerances ϵ_R and optimal solutions.

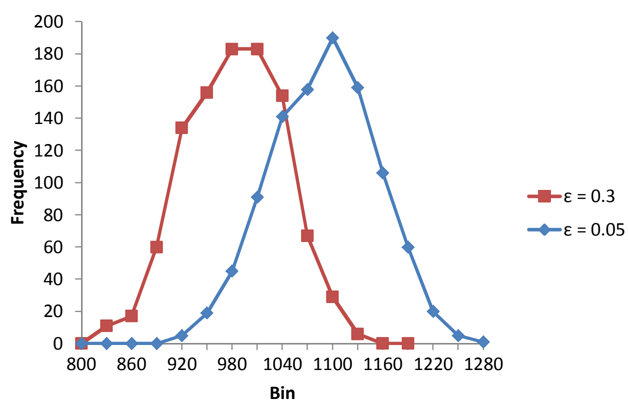


Figure 5.1: Simulated solutions, $\sum_{i=1}^{n} c_i x_i$ for $\epsilon = 0.3$ and $\epsilon = 0.05$

General case

We reformulate the problem based on moment information (Problem (5.14)) for this case:

$$
\begin{aligned}
\min \quad & B \\
\text{s.t.} \quad & \frac{\ln \epsilon}{\theta} + \sum_{i}^{n} \left(c_i^{+} - \frac{1}{\theta} \ln\, F_\theta(m, \widehat{c}_i) \right) x_i \geq A \\
& \sum_{i=1}^{n} CD_i x_i \leq B \\
& x_i \in \{0, 1\} \\
& \theta \geq 0.
\end{aligned}
\tag{5.18}
$$

In this case, we do not know the distribution that would help us calculate the real tolerances. For this reason, we use the probability bound (5.15) to see whether the solution we get from Problem (5.18) satisfies the tolerance we wanted.

Using the solution we get from Problem (5.18), we can compute the mean $\sum_i^n \bar{c}_i x_i$ and measure of uncertainty $\sum_i^n \hat{c}_i x_i$, so we can use the bound below to find an upper bound for the realized probability:

$$E_f[\exp\{-\theta c\}] = \exp\left\{-\theta \sum_{i=1}^n \bar{c}_i x_i\right\} \exp\left\{-\theta \sum_{i=1}^n \hat{c}_i x_i\right\} F_\theta \left(m, \sum_{i=1}^n \widehat{c}_i x_i\right).$$

where we have a new value for m, which is;

$$m = \frac{\sum_i^n \hat{c}_i x_i}{\sqrt{\sum_i^n \sigma_i^2 x_i}}.$$

Thus, an upper bound for the realized protection level is

$$\epsilon_{Up} = \exp\{\theta\, A\}\, E_f[\exp\{-\theta c\}]$$

Table 5.4 shows that the bound we get for the corresponding solution lead to smaller realized tolerances than the decision-maker was willing to have. To reduce the over-conservativeness of the model, we can set higher ϵ_M values than the desired tolerance level and reduce them iteratively.

5.3 Comparison with Bertsimas-Sim model

5.3.1 Setup

Our robust optimization model (5.14) has the appealing feature of being a deterministic linear problem with modified objective coefficients; however, the approach proposed by Bertsimas and Sim

ϵ_M	0.05	0.1	0.15	0.175	0.2
ϵ_{Up}	0.0092	0.0178	0.0394	0.0473	0.0711
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1
4	0	0	1	1	1
5	1	1	1	1	0
6	1	1	0	0	1
7	0	0	0	0	0
8	0	0	0	0	0
9	1	1	1	1	0
10	0	0	0	0	0
Obj.	348	348	346	346	338

Table 5.4: Tolerance computations for different ϵ values for formulation based on moment information.

[23] also prove, for a different modeling of the uncertainty based on uncertainty sets, that robust binary optimization problems with uncertain cost coefficients can also be solved either as a mixed-integer linear problem (with new constraints and variables) or as a series of binary optimization problems with modified cost coefficients. Therefore, it is natural to wonder about the relative performance of our approach compared to the Bertsimas and Sim framework.

Bertsimas and Sim [23] model the uncertain coefficients as uncertain parameters in the range forecast $[\bar{c}_i - \widehat{c}_i,\, \bar{c}_i + \widehat{c}_i]$ for each i and define $\Gamma \in \{0, \ldots, n\}$, also called *budget of uncertainty*, as a measure of the decision-maker's conservatism by capturing the number of coefficients that can take their worst-case values simultaneously. The robust counterpart of the problem in their framework is:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} \bar{c}_i\, x_i \; - \Gamma z_0 - \sum_{i=1}^{n} z_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} cd_i\, x_i \leq B \\
& \mathbf{x} \in \mathcal{X} \\
& z_i + z_0 \geq \hat{c}_i\, x_i, \qquad\qquad \forall i, \\
& z_i, z_0 \geq 0 \qquad\qquad\qquad \forall i.
\end{aligned}
\tag{5.19}
$$

We tested the models with 6 sets of randomly generated data using Normal or Uniform distribution. Size of each data set can be seen on Table (5.5).

5.3.2 Results

We ran model (5.19) for all Γ values from 0 to n for each data set. We ran model (5.14) for 500 iterations where θ values start at 0.01 and are increased by 0.001 in each iteration. Unlike Problem (5.5), objective function of Problem (5.14) is unbounded for a given x solution (see Figure 5.4). Therefore, finding an *optimal* θ is not as easy as in Problem (5.5). We modeled the problems in the mathematical modeling software GAMS and used IPM ILOG CPLEX 12.2 as our solver. Table 5.5 shows the average solution times of each problem. Problem (5.14) – which we call the D&T model – takes a shorter time to solve the problem in each data set. This is not surprising since Problem (5.19) – which we call the B&S model – increases the problem size whereas Problem (5.14) does not. Solving Problem (5.19) as a series of binary problems increases solution time substantially further and those numbers are omitted here.

	n	**B&S**	**D&T**
Data Set 1	100	0.10	0.07
Data Set 2	100	0.08	0.07
Data Set 3	300	55.59	18.00
Data Set 4	500	78.27	14.36
Data Set 5	1000	176.26	15.64
Data Set 6	100000	Time Limit	119.07

Table 5.5: Average solution time of Models (5.19)–(B&S) and (5.14)–(Non-Gaussian) with respect to different data sets

We simulated 10,000 instances where each revenue component independently deviates from the nominal value to lower bound with probability 0.4. Figures 5.2 and 5.3 compare the simulation results for different solutions on a histogram and cumulative probability distribution. In Figure 5.2 we see that the B&S solution reduces the downside risk the most; however, it is also the most conservative solution. The D&T solution also reduces the downside risk to some degree but it is not conservative. As we look at the cumulative distributions of the solutions in Figure 5.3, we

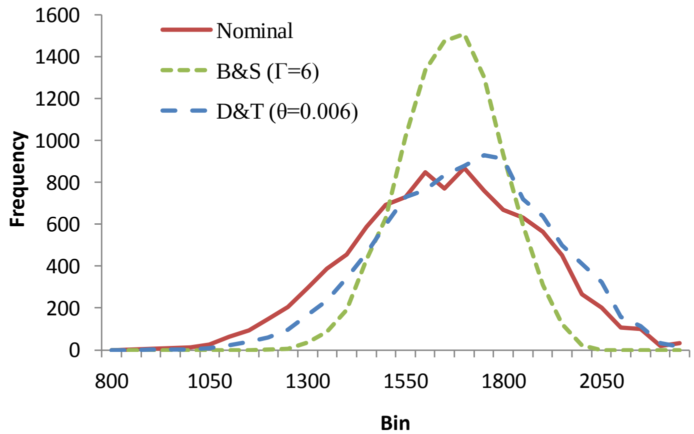


Figure 5.2: Histogram of Revenues

see that even if the B&S solution had a histogram with small variance, it does not dominate the nominal solution for all values. The D&T solution dominates the nominal solution and the break point suggests that it is better than B&S solution 70% of the time.

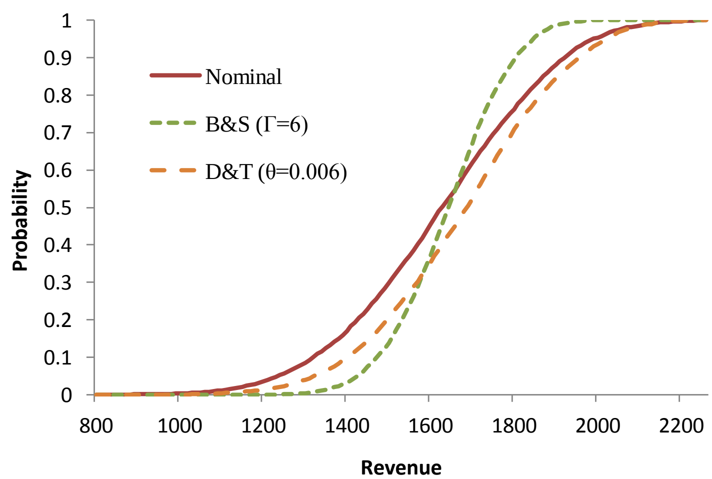


Figure 5.3: Cumulative Probability Distributions

We now analyze the sensitivity on θ. Figure 5.4 shows the change in objective function as a function of θ for Model (5.14). If we look at the sensitivity analysis for θ, we see that it changes the objective function value significantly when it is small. Numerical results show that the solution is not very sensitive to the changes in θ. We observed that we get only 2 or 3 different solutions in 500 iterations. Therefore, the decision maker is advised to search for different solutions and chose the one that dominates the nominal solution and other solutions. In the examples we solve, we see that very small θ values gives the nominal solution, and very large θ values do not always dominate all the other solutions. A sensitivity analysis would be helpful to find the solution that is not conservative yet robust.

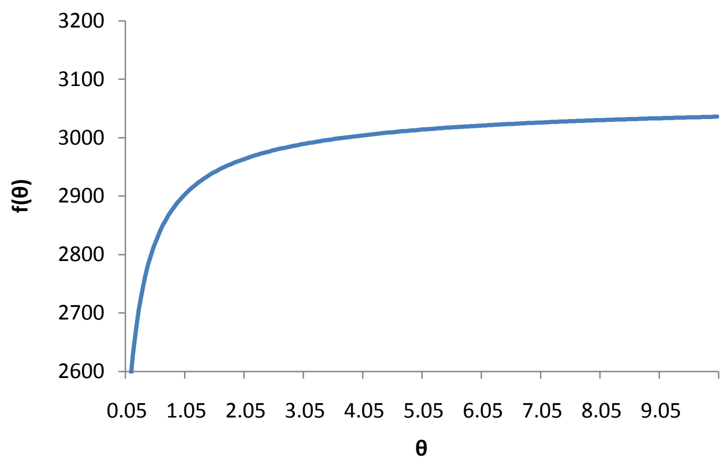


Figure 5.4: Sensitivity Analysis for θ

5.4 Chance Constraints for Correlated Data

So far we have assumed that the uncertain data are independent. We now relax this assumption. Let parameter c_i be defined as $c_i = \bar{c}_i + \sum_j d_{i,j} z_j$ where z_j's are independent with mean 0 and standard deviation 1, and d represents the square root of the covariance matrix of c. This yields:

$$\sum_{i=1}^{n} c_i x_i = \sum_{i=1}^{n} \bar{c}_i x_i + \sum_{j=1}^{n} (\sum_{i=1}^{n} d_{i,j}\, x_i)\, z_j. \tag{5.20}$$

We assume that independent random variables z_i fall within a given range $[-b_i, b_i]$. Both robust optimization models (with and without Gaussian assumption) can easily extended to the case where we have data correlation.

5.4.1 Correlated Data with Gaussian Assumption

We reformulate the right hand side of Eq. (5.7):

$$\frac{E[\exp\{-\theta\sum_{i=1}^{n} c_i x_i\}]}{\exp\{-\theta A\}} = \exp\{\theta A\}\prod_{i=1}^{n} E[\exp\{-\theta x_i(\bar{c}_i + \sum_j d_{i,j} z_j)\}] \tag{5.21}$$

$$= \exp\{\theta A\}\prod_{i=1}^{n} \exp\{-\theta x_i \bar{c}_i\}\prod_j E[\exp -\theta x_i\, d_{i,j}\, z_j] \tag{5.22}$$

$$\leq \exp\{\theta A\}\prod_{i=1}^{n} \exp\{-\theta x_i \bar{c}_i\}\prod_j \exp\{0 + \frac{1}{2}\theta^2\, x_i\, d_{i,j}^2\} \tag{5.23}$$

$$= \theta A - \sum_i \theta x_i \bar{c}_i + \sum_i \sum_j \left(\frac{1}{2}\theta^2\, x_i\, d_{i,j}^2\right) \leq \ln \epsilon \tag{5.24}$$

Eq. (5.24) is due to the $(0, d_{i,j}^2)$ mean and variance of random variable $d_{i,j}\, z_i$.

Then the objective function $f(\theta, x)$ becomes:

$$f(\theta, x) = \frac{\ln \epsilon}{\theta} + \sum_i \bar{c}_i\, x_i - \frac{\theta}{2}\sum_i \sum_j d_{i,j}\, x_i. \tag{5.25}$$

5.4.2 Correlated Data without Gaussian Assumption

Theorem 5.3 shows that:

$$\max_{f\in\pi} E_f[\exp\{-\theta c\}] = \exp\{-\theta\bar{c}\}\exp\{-\theta\hat{c}\}\left(1 + \theta\hat{c} + \frac{(m^2+1)}{m^2}\frac{\theta^2\hat{c}^2}{2}\right).$$

where c is a random variable that comes from a symmetric uncertainty set with mean $\bar{c}$, range $[c^-, c^+]$ and $\hat{c} = \bar{c} - c^-$. Consider Eq. (5.22). From Theorem 5.3, we know that

$$\max_{f\in\pi} E_f[\exp\{-\theta dz\}] = \exp\{-\theta\bar{dz}\}\exp\{-\theta\hat{dz}\}\left(1 + \theta\hat{dz} + \frac{(m^2+1)}{m^2}\frac{\theta^2\hat{dz}^2}{2}\right)$$

where $\bar{dz} = 0$, $\hat{dz} = db$ and $m = \dfrac{b}{d}$. In other words, we have

$$\max_{f \in \pi} E_f[\exp\{-\theta dz\}] = \exp\{-\theta db\} \left(1 + \theta db + (b^2 + d^2)\frac{\theta^2 d^2}{2} \right).$$

Let

$$F_\theta(\frac{b}{d}, dz) = \left(1 + \theta db + (b^2 + d^2)\frac{\theta^2 d^2}{2} \right)$$

If we continue to Equation (5.22)

$$\frac{E[\exp\{-\theta \sum_{i=1}^n c_i x_i\}]}{\exp\{-\theta A\}} = \exp\{\theta A\} \prod_{i=1}^n \exp\{-\theta x_i \bar{c}_i\} \prod_j E[\exp -\theta x_i\, d_{i,j}\, z_j] \tag{5.26}$$

$$\leq \exp\{\theta A\} \prod_{i=1}^n \exp\{-\theta x_i \bar{c}_i\} \prod_j \exp\{-\theta d_{ij}\, b_j x_i\} F_\theta(\frac{b_i}{d_{ij}}, d_{ij} z_j) \tag{5.27}$$

$$= \theta A - \theta \sum_{i=1}^n x_i \bar{c}_i + \sum_{i=1}^n \sum_{j=1}^n \left[-\theta d_{ij} b_j + \ln \left(F_\theta(\frac{b_i}{d_{ij}}, d_{ij} z_j) \right) \right] x_i \tag{5.28}$$

Then, the objective function of the robust problem when data are correlated becomes:

$$f(\theta, x) = \frac{\ln \epsilon}{\theta} + \sum_{i=1}^n \bar{c}_i\, x_i + \sum_{i=1}^n \sum_{j=1}^n \left[d_{i,j}\, b_j - \frac{1}{\theta} F_\theta(\frac{b_i}{d_{ij}}, d_{ij} z_j) \right] x_i \tag{5.29}$$

5.5 Chance Constraints for Multi-Range Uncertainty

Now, we will extend our model to multiple ranges of uncertainty. We keep the same notation as in the previous chapter; in particular, range k for coefficient i is denoted $[c_i^{k-},\ c_i^{k+}]$. The probability of the i-th coefficient falling into the k-th range (high, low, medium) is denoted p_i^k.

5.5.1 Special Case: Normally Distributed Parameters

Assume that the parameters are independent and Normally distributed. Consider the Markov's Inequality $\exp\{\theta A\}\prod_{i=1}^{n} E[\exp\{-\theta c_i x_i\}] \le \epsilon$ but this time we have

$$E[\exp\{-\theta c_i x_i\}] = \sum_{k=1}^{K} p_i^k E[\exp\{-\theta c_i^k x_i\}|c_i = c_i^k], \tag{5.30}$$

where the conditioning occurs over the K possible ranges. This can be bounded by (using x_i binary):

$$E[\exp\{-\theta c_i x_i\}] \le \sum_{k=1}^{K} p_i^k \exp\left(-\theta\mu_i^k x_i + \frac{1}{2}\theta^2\sigma_{i,k}^2 x_i\right).$$

We substitute this bound in place of $E[\exp\{-\theta c_i x_i\}]$ and get a new equation;

$$\theta A + \sum_{i}^{n} \ln\left[\sum_{k=1}^{K} p_i^k \exp\left(-\theta\mu_i^k x_i + \frac{1}{2}\theta^2\sigma_{i,k}^2 x_i\right)\right] \le \ln\epsilon$$

Our robust optimization model becomes:

$$\begin{aligned} \max \quad & \frac{\ln\epsilon}{\theta} - \frac{1}{\theta}\sum_{i=1}^{n} x_i \ln\left[\sum_{k=1}^{K} p_i^k \exp\left(-\theta\mu_i^k + \frac{1}{2}\theta^2\sigma_{i,k}^2\right)\right] \\ \text{s.t.} \quad & x \in \mathcal{X} \subseteq \{0,1\}^n, \\ & \theta \ge 0. \end{aligned} \tag{5.31}$$

Again, at θ given, the robust model is a nominal model with modified objective coefficients.

Figure 5.5 below shows the graph of coefficients when $i = 1, 2, 3$ as a function of θ. For all coefficients, we find θ_i that minimizes the coefficient in front of x_i. This yields n values of θ. We solve Model (5.31) for all these θ values and determine the θ that maximizes our model, which is denoted $\tilde{\theta}$. If the solution (in x) did not change between $\tilde{\theta}$ and its 2 neighbor θ values, then we found the optimal solution. If the solution changes, we resolve the model using the average of 2 consecutive θ values as our new θ. We continue to resolve the model using the average of θ that maximizes the model and its 2 consecutive values until we obtain the same solution.

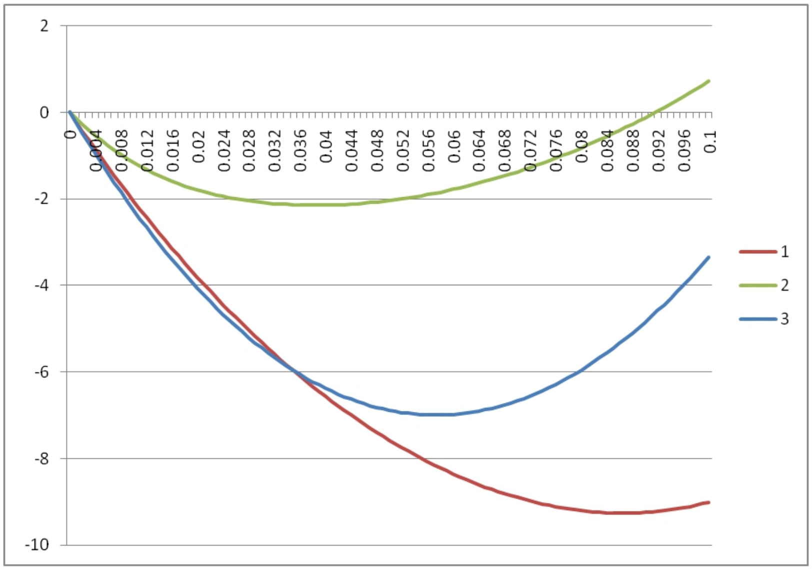


Figure 5.5: Coefficient of x_i when $i = 1, 2, 3$.

5.5.2 General Case: Formulation Based on Moment Information

We can easily extend inequality (5.30) to find bounds using Theorem 5.3. If we combine the bound found in Theorem 5.3 with Eq. (5.30), we obtain:

$$E[\exp\{-\theta c_i x_i\}] \leq \sum_{k=1}^{K} p_i^k \exp\{-\theta \overline{c}_i^k\} \, F(m, \widehat{c}_i^k).$$

Now, we insert this equation into Markov's Inequality:

$$\exp\{\theta A\} \prod_{i=1}^{n} \left(\sum_{k=1}^{K} p_i^k \exp\{-\theta \overline{c}_i^k\} \, F(m, \widehat{c}_i^k) \right) \leq \epsilon,$$

or equivalently, using x_i binary for all i:

$$\begin{aligned} \max \quad & \frac{\ln \epsilon}{\theta} - \frac{1}{\theta} \sum_{i}^{n} x_i \ln \left(\sum_{k=1}^{K} p_i^k \exp\{-\theta \overline{c}_i^k\} \, F(m, \widehat{c}_i^k) \right) \\ \text{s.t.} \quad & x \in \mathcal{X} \subseteq \{0, 1\}^n. \end{aligned} \tag{5.32}$$

Histogram 5.7 illustrates the benefit of using a finer description of uncertainty, as allowed by a

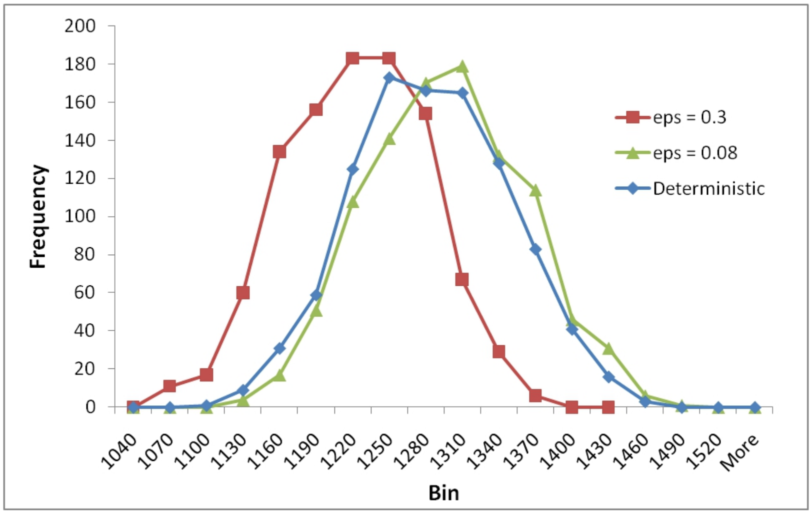


Figure 5.6: Comparison of histograms for robust and nominal models.

two-range robust optimization model, rather than a single-range one.

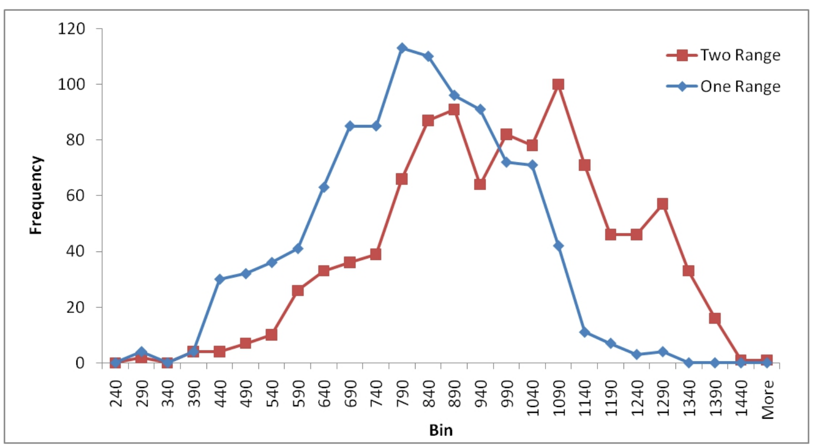


Figure 5.7: Comparison of histograms for one-range and two-range models.

5.6 Conclusions

In this chapter, we have investigated the connection between robust optimization and probabilistic models when decision variables are binary, incorporating various amounts of distributional information into the problem formulation. The approach further motivates the use of robust optimization problems with a linear structure; we have shown that the robust counterparts are deterministic problems with modified objective coefficients, which depend on a new parameter introduced in the Bernstein approximation. We could easily incorporated the case when there is a correlation between projects. The comparison between our approach and the robust discrete optimization model of Bertsimas and Sim shows that our solution quality is comparable to theirs while our approach is significantly faster.

Bibliography

[1] Elodie Adida and Georgia Perakis. A robust optimization approach to dynamic pricing and inventory control with no backorders. *Mathematical Programming, Ser. B*, 107:97–129, 2006.

[2] Shabbir Ahmed, Mohit Tawarmalani, and Nikolaos V. Sahinidis. A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, 100:355–377, 2004.

[3] Norman Baker and James Freeland. Recent advances in R&D benefit measurement and project selection methods. *Management Science*, 21(10), 1975.

[4] Aharon Ben-Tal, Dimitris Bertsimas, and David B. Brown. A soft robust model for optimization under ambiguity. *Operations Research*, 2010. forthcoming.

[5] Aharon Ben-Tal, Stephen Boyd, and Arkadi Nemirovski. Extending scope of robust optimization: Comprehensive robust counterparts of uncertain problems. *Mathematical Programming, Ser. B*, 107:63–89, 2006.

[6] Aharon Ben-Tal, Laurent El-Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2009.

[7] Aharon Ben-Tal, Boaz Golany, Arkadi Nemirovski, and Jean-Philippe Vial. Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing and Service Operations Management*, 7(3):248–271, 2005.

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

[9] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

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

[11] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.

[12] Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization: Methodology and applications. *Mathematical Programming*, 92:453–480, 2002.

[13] Dimitri P Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3. edition, 2005.

[14] Dimitri P Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 3. edition, 2007.

[15] Dimitris Bertsimas and David B. Brown. Constructing uncertainty sets for robust linear optimization. *Operations Research*, 57(6):1483–1495, 2009.

[16] Dimitris Bertsimas and Constantine Caramanis. Finite adaptability in linear optimization. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[17] Dimitris Bertsimas and Vineet Goyal. On the power and limitations of affine policies in two-stage adaptive optimization. *Submitted to Math. Programming*, 2009.

[18] Dimitris Bertsimas and Vineet Goyal. On the power of robust solutions in two-stage stochastic and adaptive optimization problems. *Submitted to Mathematics of Operations Research*, 2009.

[19] Dimitris Bertsimas and Dessislava Pachamanova. Robust multiperiod portfolio management with transaction costs. *Computers and Operations Research*, 35(1):3–17, 2008.

[20] Dimitris Bertsimas, Dessislava Pachamanova, and Melvyn Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32:510–516, 2004.

[21] Dimitris Bertsimas and Ioana Popescu. On the relation between option and stock prices: A convex optimization approach. *Operations Research*, 50(2):358–374, 2002.

[22] Dimitris Bertsimas and Ioana Popescu. Optimal inequalities in probability theory: A convex optimization approach. *Society for Industrial and Applied Mathematics*, 15(3):780–804, 2005.

[23] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical Programming B*, 98:49–71, 2003.

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

[25] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization under ellipsoidal uncertainty sets. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[26] Dimitris Bertsimas and Aurélie Thiele. Robust and data-driven optimization: Modern decision-making under uncertainty. *Tutorials in Operations Research*, pages 95–122, 2006.

[27] Dimitris Bertsimas and Aurélie Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168, 2006.

[28] Daniel Bienstock and Nuri Ozbay. Computing robust base-stock levels. *Discrete Optimization*, 5:389–414, 2008.

[29] John R. Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Series in Operations Research, Springer-Verlag, New York, 1997.

[30] Claus C. Carøe and Jørgen Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83:451–464, 1998.

[31] Xin Chen, Melvyn Sim, and Peng Sun. A robust optimization perspective of stochastic programming. *Operations Research*, 55(6):1058–1071, 2007.

[32] Xin Chen, Melvyn Sim, Peng Sun, and Jiawei Zhang. A linear decision-based approximation approach to stochastic programming. *Operations Research*, 56(2):344–357, 2008.

[33] Xin Chen and Jiawei Zhang. Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482, 2009.

[34] Paul D. Childs and Alexander J. Triantis. Dynamic R&D investment policies. *Management Science*, 45(10):1359–1377, 1999.

[35] George B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955.

[36] Jitka Dupačová. Minimax stochastic programs with nonseparable penalties. In *Optimization techniques (proc. of the 9th IFIP Conference on Optimization Techniques, Warsaw, 1979), Part 1, Lecture Notes in Control and Information Sciences*, volume 22, pages 157–163. Springer Berlin / Heidelberg, 1980.

[37] Ruken Düzgün and Aurélie Thiele. Robust R&D project management. Technical report, Lehigh University, Bethlehem, PA, 2010.

[38] Martin Dyer and Leen Stougie. Computational complexity of stochastic programming problems. *Mathematical Programming*, 106:423–432, 2006.

[39] Harel Eilat, Boaz Golany, and Avraham Shtub. Constructing and evaluating balanced portfolios of R&D projects with interactions: A dea based methodology. *European Journal of Operational Research*, 172:1018–1039, 2006.

[40] Laurent El-Ghaoui and Hervé Lebret. Robust solutions to least-square problems to uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[41] Laurent El-Ghaoui, Francois Oustry, and Hervé Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9:33–52, 1998.

[42] Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 2010. forthcoming.

[43] Peter Kall and Stein W.Wallace. *Stochastic Programming*. Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons Ltd. Chichester, 2. edition, 1994.

[44] Ban Kawas and Aurélie Thiele. A robust optimization approach to log-robust portfolio management. *OR Spectrum*, 2010. forthcoming.

[45] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer-Verlag, Berlin, Heidelberg, Germany, 2004.

[46] Ali Koc, David P. Morton, Elmira Popodva, Ernie Kee Stephen M. Hess, and Drew Richards. Prioritizing project selection. *The Engineering Economist*, 54:267–297, 2009.

[47] Manuel Laguna. Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty. *Management Science*, 44(11):101–110, 1998.

[48] Gilbert Laporte and Francois V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13:133–142, 1993.

[49] Jonathan D. Linton, Steven T. Walsh, and Joseph Morabito. Analysis, ranking and selection of r&d projects in a portfolio. *R&D Management*, 32(2):139–148, 2002.

[50] Scott A. Malcolm and Stavros A. Zenios. Robust optimization for power systems capacity expansion under uncertainty. *The Journal of the Operational Research Society*, 45(9):1040–1049, 1994.

[51] Joseph P. Martino. *Research and Development Project Selection*. Wiley Series in Engineering & Technology Management, New York, 1995.

[52] Gokhan Metan and Aurélie Thiele. Scenario design in robust inventory management. Technical report, Lehigh University, Bethlehem, PA, 2008.

[53] John M. Mulvey, Robert J. Vanderbei, and Stavros A. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43(2):264–281, 1995.

[54] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Dicrete Mathematics and Optimization, USA, 1988.

[55] Lewis Ntaimo and Suvrajeet Sen. A branch-and-cut algorithm for two-stage stochastic mixed-binary programs with continuous first-stage variables. *International Journal of Computational Science and Engineering*, 3(3):232–241, 2007.

[56] Fernando Ordoñez and Jiamin Zhao. Robust capacity expansion of network flows. *Networks*, 50(2):136–145, 2007.

[57] Warren B. Powell. *Approximate Dynamic Programming: solving the curses of dimentionality*. Wiley-Interscience Series in Probability and Statistics, John Wiley & Sons, Inc. Hoboken, New Jersey, 2007.

[58] András Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, Boston, 1995.

[59] Andrzej Ruszczyński and Alexander Shapiro. *Stochastic Programming*, volume 10. Handbooks in Operations Research and Management Science, North-Holland, Amsterdam, 2003.

[60] Chuen-Teck See and Melvyn Sim. Robust approximation to multi-period inventory management. *Operations Research*, 2010. forthcoming.

[61] Wenqing See and Melvyn Sim. Goal-driven optimization. *Operations Research*, 57(2):342–357, 2009.

[62] Suvrajeet Sen and H. D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006.

[63] Alexander Shapiro. Worst-case distribution analysis of stochastic programs. *Mathematical Programming, Ser. B*, 107:91–96, 2006.

[64] Alexander Shapiro and Anton Kleywegt. Minimax analysis of stochastic problems. *Optimization Methods and Software*, 17:523–542, 2002.

[65] Alexander Shapiro and Arkadi Nemirovski. On complexity of stochastic programming problems. In *Continuous Optimization*, volume 99, pages 111–146. Springer US, 2005.

[66] Allen L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.

[67] Christian Stummer and Kurt Heidenberger. Interactive r&d portfolio analysis with project interdependencies and time profiles of multiple objectives. *IEEE Transactions On Engineering Management*, 50(2):175–183, 2003.

[68] Aurélie Thiele. Robust stochastic programming with uncertain probabilities. *IMA Journal of Management Mathematics*, 19(3):289–321, 2008.

[69] Aurélie Thiele, Tara Terry, and Marina Epelman. Robust linear optimization with recourse. Technical report, Lehigh University and University of Michigan, 2009.

[70] Christiana Vassiadou-Zeniou and Stavros A. Zenios. Robust optimization models for managing callable bond portfolios. *European Journal of Operational Research*, 91(2):264–273, 1996.

[71] Wei Ye and Fernando Ordoñez. Robust optimization models for energy-limited wireless sensor networks under distance uncertainty. *IEEE Transactions on Wireless Communications*, 7(6):2161–2169, 2008.

Vita

Ruken Duzgun was born in Hatay, Turkey. She received her B.S. degree from Middle East Technical University in Ankara, Turkey, in 2007. She has a Master's degree on Management Science (2010) from Lehigh University and she is currently a doctoral candidate at the Industrial Engineering Department at Lehigh University and expects to graduate on May 2012. Her research interests include robust optimization, mixed integer programming and chance constraints. She is a Rossin Doctoral Fellow since 2010. She contributed to Turkish Students Club and Lehigh University INFORMS Student Chapter as a board member and she is a student member of INFORMS, Women in OR/MS (WORMS) Society and The New York Academy of Science. She will join Revenue Management group at Marriott International Inc. after graduation.