

Robust Linear Optimization With Recourse

Aurélie Thiele*, Tara Terry† and Marina Epelman‡

March, 2010

Abstract

We propose an approach to two-stage linear optimization with recourse that does not involve a probabilistic description of the uncertainty and allows the decision-maker to adjust the degree of conservativeness of the model, while preserving its linear properties. We model uncertain parameters as belonging to a polyhedral uncertainty set and minimize the sum of the first-stage costs and the worst-case second-stage costs over that set, i.e., taking a robust optimization approach. The decision-maker's conservatism is taken into account through a budget of uncertainty, which determines the size of the uncertainty set around the nominal values of the uncertain parameters. We establish that the robust problem is a linear programming problem with a potentially very large number of constraints, and describe how a cutting plane algorithm can be used in the two-stage setting. We test the robust modeling approach on two examples of problems, one with simple, and one with general recourse, to illustrate the structure and performance of robust policies as well as to evaluate the performance of the cutting plane algorithm.

1 Introduction

Linear optimization with recourse was first introduced by Dantzig in [17] as a mathematical framework for sequential decision-making under uncertainty. In that setting, the decision-maker must make some decisions before discovering the actual value taken by the uncertain parameters, but has the opportunity to take further action once uncertainty has been revealed, with the objective of minimizing total expected cost. This framework later became known as stochastic programming and is described in detail in the monographs by Birge and Louveaux [13] and Kall and Wallace [25]. However, as early as the mid-1960s, researchers such as Dupačová, née Žáčková, [37] recognized the practical limitations of the expected-value paradigm, which requires the exact knowledge of the underlying probability distributions. The fact that such probabilities are very hard to estimate in practice motivated the development of a *minimax approach*, where the decision-maker minimizes the maximal expected cost over a family of probability distributions. It has received significant attention in the stochastic programming literature, for instance from Dupačová, [18, 19, 20, 37],

*Department of Industrial and Systems Engineering, Lehigh University, aurelie.thiele@lehigh.edu. Supported in part by NSF Grant DMI-0540143, a P.C. Rossin Assistant Professorship and an IBM Faculty Award.

†Industrial and Operations Engineering Department, University of Michigan, tlterry@umich.edu. Supported in part by NSF Grant CCF-0306240 and Elizabeth C. Crosby Research Award.

‡Industrial and Operations Engineering Department, University of Michigan, mepelman@umich.edu. Supported in part by NSF Grant CCF-0306240 and Elizabeth C. Crosby Research Award.

whose work laid the foundation for subsequent research efforts. Other early references include Jagannathan [24], who studied stochastic linear programming with simple recourse when the first two moments of the distributions are known, and Birge and Wets [14], who focused on bounding and approximating stochastic problems. More recently, Shapiro and Ahmed [31] and Shapiro and Kleywegt [32] have investigated further the theoretical properties of minimax stochastic optimization, while Takriti and Ahmed described in [35] an application to electricity contracts. The main drawback of the stochastic minimax approach is that the solution methods proposed in the literature (a stochastic gradient technique in Ermoliev et al. [23], a bundle method in Breton and El Hachem [15], a cutting-plane algorithm in Riis and Andersen [29], to name a few), all require finding explicitly the worst-case probability distribution for the current candidate solution at each step of the algorithm, and hence suffer from dimensionality problems. These are particularly acute here as stochastic programming often yields large-scale formulations. Although Shapiro and Ahmed [31] and Shapiro and Kleywegt [32] have studied specific classes of problems for which the minimax framework leads to traditional stochastic formulations, no such approach has been developed to date for the general case. Furthermore, while the field of stochastic programming has seen in recent years a number of algorithmic advances, e.g., sampling methods (Shapiro [30]), the problem with recourse remains significantly more difficult to solve than its deterministic linear counterpart, and does not allow for easy insights into the impact of uncertainty on the optimal values of the decision variables.

Therefore, the need arises to develop an approach to linear optimization with recourse that does not rely on a probabilistic description of the uncertainty, yields insights into the way uncertainty affects the optimal solution, and remains solvable in practice, even if not tractable in the theoretical sense. The purpose of this paper is to present such an approach based on robust optimization. While robust optimization has been previously used in stochastic programming as a method to incorporate cost variability in the objective function (Takriti and Ahmed [36]), we consider here a different methodology, which was developed independently under the same name. What we refer to as robust optimization addresses data uncertainty in mathematical programming problems by finding the optimal solution for the worst-case instances of unknown but bounded parameters.

Traditionally, the term “robust optimization” refers to an approach to dealing with parameter uncertainty in one-stage optimization problems (i.e., problems without recourse). This approach was pioneered in 1973 by Soyster [33], who proposed a model that guarantees feasibility for all instances of the parameters within a convex set. However, the geometry of this uncertainty set he considered to assure tractability of the resulting model lead to very conservative solutions, in the sense that they are too far from optimality in the nominal model to be of practical interest for real-life implementation. This issue of over-conservatism hindered the adoption of robust techniques in optimization problems until the mid-1990s, when Ben-Tal and Nemirovski [3, 4, 5], El Ghaoui and Lebret [21] and El Ghaoui et al. [22] started investigating tractable robust counterparts of linear, semidefinite and other important convex optimization problems. They focused on ellipsoidal uncertainty sets, which allow for important insights into the robust framework but increase the complexity of the problem considered, e.g., yield second-order cone problems as the robust counterpart of linear models. In contrast, Bertsimas and Sim [8] study polyhedral uncertainty sets, which do not change the class of the problem at hand, and explicitly quantify the trade-off between performance and conservatism in terms of probabilistic bounds of constraint violation. An advantage of their approach is that it can be easily extended to integer and mixed-integer programming problems (Bertsimas and Sim [7]). While robust optimization has been applied in the references above as a way to address parameter uncertainty, Bertsimas and Thiele [9] use this framework to

model random variables and address uncertainty on the underlying distributions in a multi-period inventory problem. Their approach highlights the potential of robust optimization for dynamic decision-making in the presence of randomness.

A first step towards applying robust techniques in stochastic programming with recourse is taken by Ben-Tal et al. [2], who coin the terms “adjustable decision variables” as a synonym for second-stage, or recourse, decisions, and “adjustable robust optimization” for robust counterparts of problems with recourse. Unfortunately, the resulting adjustable robust optimization problems are frequently computationally intractable, which leads them to restrict the recourse variables to affine functions of the uncertain data. Most of the following literature imposes some similar type of restriction on recourse variables to obtain a theoretically tractable optimization problem that approximates the adjustable robust counterpart (e.g., “deflected-linear” functions considered by Chen and Sim in [16] and references therein), or identifies special cases when the robust adjustable model at hand is tractable. Takeda et al. [34] discuss conditions for tractability of convex robust problems with recourse; Ordóñez and Zhao [28], for example, consider a tractable special case of network design under uncertain demand. Atamtürk and Zhang in [1] also propose a robust adjustable model for two-stage network flow and design under uncertain demand, but do not invoke affinely adjustable decision variables. Similarly, in this paper we do not impose any limitations on the structure of the functional dependence of the recourse on the uncertain parameters. The model presented here is broader in scope than the network design problem considered in [1], in that we develop a robust approach for general two-stage stochastic linear problems with uncertainty on the right-hand side. We believe that the framework proposed in this paper offers a new perspective on linear programming with recourse that combines the decision-maker’s degree of conservatism and the uncertainty on the probability distributions.

The algorithm proposed here can be seen as a variation of Benders’ decomposition, or delayed constraint generation (see Section 3 for details). Previous extensions of decomposition to robust optimization problems include work by Bienstock and Özbay [12] and Bienstock [11]. In these papers the authors consider specific applications of single-stage, non-adjustable, robust framework (determining a robust basestock level under uncertain demands, and robust portfolio optimization with uncertain returns, respectively), and apply a generic decomposition algorithm that alternates between solving a restricted master problem that includes a limited subset of possible data realizations to determine a solution, and an *adversarial problem* which finds, exactly or approximately, the worst-case data realization for the solution found. The newly identified data realization is then added to the restricted master problem, and the process is repeated. The papers focus on the specifics of solving the master and adversarial problems for the applications and uncertainty sets considered. While working on this manuscript, we became aware of the recent work by Mutapcic and Boyd [27], which applies the same paradigm and its variations to robust counterparts of single-stage convex problems and presents several examples. In contrast to the ordinary one-stage robust problems considered in the above papers, in this paper we deal with robust problems *with recourse*, i.e., two-stage problems, which present additional challenges (specifically in solving the adversarial problem). We discuss exact solution methods of adversarial problems arising in problems with recourse under right-hand-side uncertainty in problems with simple as well as general recourse.

Specifically, we make the following contributions:

1. We address right-hand-side uncertainty in linear programming problems with recourse by considering an adjustable (two-stage) robust formulation with a polyhedral uncertainty set. The level of conservatism of the optimal solution is flexibly adjusted by setting a parameter

called the “budget of uncertainty” to an appropriate value.

2. We propose a cutting-plane method, based on Kelley’s method, for solving adjustable robust linear programs of this type. This method is similar to, but less computationally demanding than, Benders’ decomposition algorithm for their stochastic counterparts.
3. We provide techniques for finding worst-case realizations of the uncertain parameters within the polyhedral uncertainty set for problems with simple and general recourse, and demonstrate their performance via computational experiments.
4. Finally, using the above methods we obtain solutions to robust formulations of two example problems, which yield insight into structure and performance of such solutions under uncertainty.

The organization of the paper is as follows. In Section 2, we define the model of uncertainty and present the main ideas underlying the robust approach. Section 3 presents a cutting-plane algorithm based on Kelley’s method for solving robust linear programs with recourse, along with a comparison with the Benders’ decomposition algorithm for solving stochastic programming problems. Sections 4 and 5 discuss how one can compute objective function values and subgradients in the implementation of our algorithm for problems with simple and general recourse, respectively, focusing on solving the adversarial problem. Sections 6 and 7 present results of computational experiments on problems with these two types of recourse. Finally, Section 8 contains some concluding remarks.

2 Problem Overview

2.1 Optimization with Recourse

The focus of this paper is on two-stage linear optimization with right-hand side uncertainty, which was first described by Dantzig in [17]. The deterministic problem can be formulated as:

$$\begin{aligned}
 \min \quad & \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \\
 \text{s.t.} \quad & \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{y} = \mathbf{b} \\
 & \mathbf{x} \in S, \mathbf{y} \geq \mathbf{0}
 \end{aligned} \tag{1}$$

with the following notation:

- \mathbf{x} : the first-stage decision variables,
- \mathbf{y} : the second-stage decision variables,
- \mathbf{c} : the first-stage costs,
- \mathbf{d} : the second-stage costs,
- \mathbf{A} : the first-stage coefficient matrix,
- \mathbf{B} : the second-stage coefficient matrix,
- \mathbf{b} : the requirement vector.

Here S is a polyhedron described by a finite list of inequalities.

In many applications, the requirement vector is uncertain and the decision-maker implements the first-stage (“here-and-now”) variables without knowing the actual requirements, but chooses the second-stage (“wait-and-see”) variables only after the uncertainty has been revealed. This has traditionally been modeled using stochastic programming techniques, i.e., by assuming that the

requirements obey a known probability distribution and minimizing the expected cost of the problem. In mathematical terms, if we define the recourse function, once the first-stage decisions have been implemented and the realization of the uncertainty is known, as:

$$Q(\mathbf{x}, \mathbf{b}) = \min_{\mathbf{y}} \mathbf{d}^T \mathbf{y} \quad \text{s.t.} \quad \begin{aligned} \mathbf{B}\mathbf{y} &= \mathbf{b} - \mathbf{A}\mathbf{x} \\ \mathbf{y} &\geq \mathbf{0}, \end{aligned} \quad (2)$$

the stochastic counterpart of problem (1) can be formulated as a nonlinear problem:

$$\min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} + \mathbf{E}_{\mathbf{b}}[Q(\mathbf{x}, \mathbf{b})] \quad \text{s.t.} \quad \mathbf{x} \in S. \quad (3)$$

If the uncertainty is discrete, consisting of Ω possible requirement vectors each occurring with probability π_{ω} , $\omega = 1, \dots, \Omega$, problem (3) can be written as a linear programming problem:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}_{\omega}} \quad & \mathbf{c}^T \mathbf{x} + \sum_{\omega=1}^{\Omega} \pi_{\omega} \cdot \mathbf{d}^T \mathbf{y}_{\omega} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}_{\omega} = \mathbf{b}_{\omega}, \quad \omega = 1, \dots, \Omega \\ & \mathbf{x} \in S, \mathbf{y}_{\omega} \geq \mathbf{0}, \quad \omega = 1, \dots, \Omega. \end{aligned} \quad (4)$$

However, a realistic description of the uncertainty generally requires a high number of scenarios. Therefore, the deterministic equivalent (4) is often a large-scale problem, which necessitates the use of special-structure algorithms such as decomposition methods or Monte-Carlo simulations (see Birge and Louveaux [13] and Kall and Wallace [25] for an introduction to these techniques). Problem (4) can thus be considerably harder to solve than problem (1), although both are linear. The difficulty in estimating probability distributions accurately also hinders the practical implementation of these techniques.

2.2 The Adjustable Robust Approach

In contrast with the stochastic programming framework, robust optimization models uncertain parameters using *uncertainty sets* rather than probability distributions. The objective is then to minimize the worst-case cost in that set. Specifically, let \mathcal{B} be the uncertainty set of the requirement vector. The robust problem with recourse, or adjustable robust problem, is formulated as:

$$\min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} + \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{b}) \quad \text{s.t.} \quad \mathbf{x} \in S. \quad (5)$$

If $\mathcal{B} = \{\bar{\mathbf{b}}\}$, where $\bar{\mathbf{b}}$ is a known mean/nominal value of the requirement vector, problem (5) is the “nominal” problem. As \mathcal{B} expands around $\bar{\mathbf{b}}$, the decision-maker protects the system against more realizations of the uncertain parameters, and the solution becomes more robust, but also more conservative. If the decision-maker does not take uncertainty into account, he might incur very large costs once the uncertainty has been revealed. On the other hand, if he includes every possible outcome in his model, he will protect the system against realizations that would indeed be detrimental to his profit, but are also very unlikely to happen. The question of choosing uncertainty sets that yield a good trade-off between performance and conservatism is central to robust optimization.

Following the approach developed by Bertsimas and Sim [7, 8] and Bertsimas and Thiele [9], we focus on polyhedral uncertainty sets and model the uncertain parameter b_i , $i = 1, \dots, m$, as a parameter of known mean, or nominal value, \bar{b}_i and belonging to the interval $[\bar{b}_i - \widehat{b}_i, \bar{b}_i + \widehat{b}_i]$. Equivalently,

$$b_i = \bar{b}_i + \widehat{b}_i z_i, \quad |z_i| \leq 1, \quad \forall i.$$

To avoid overprotecting the system, we impose the constraint

$$\sum_{i=1}^m |z_i| \leq \Gamma,$$

which bounds the total scaled deviation of the parameters from their nominal values. Such a constraint was first proposed by Bertsimas and Sim [8] in the context of linear programming with uncertain coefficients. The parameter Γ , which we assume to be integer, is called the *budget of uncertainty*. $\Gamma = 0$ yields the nominal problem and, hence, does not incorporate uncertainty at all, while $\Gamma = m$ corresponds to interval-based uncertainty sets and leads to the most conservative case. In summary, we will consider the following uncertainty set:

$$\mathcal{B} = \left\{ \mathbf{b} : b_i = \bar{b}_i + \widehat{b}_i z_i, \quad i = 1, \dots, m, \quad \mathbf{z} \in \mathcal{Z} \right\}, \quad (6)$$

with

$$\mathcal{Z} = \left\{ \mathbf{z} : \sum_{i=1}^m |z_i| \leq \Gamma, \quad |z_i| \leq 1, \quad i = 1, \dots, m \right\}. \quad (7)$$

In the remainder of the paper, we investigate how problem (5) can be solved for the polyhedral set defined in equations (6)–(7), and how the robust approach can help us gain insights into the impact of the uncertainty on the optimal solution.

2.3 Assumptions and Preliminaries

Consider the problem (5) with $Q(\mathbf{x}, \mathbf{b})$ defined by (2). We assume that $Q(\mathbf{x}, \mathbf{b}) > -\infty$ for all $\mathbf{x} \in S$ and $\mathbf{b} \in \mathcal{B}$ (i.e., problem (5) has *sufficiently expensive recourse* in stochastic programming parlance), and thus the dual problem of (2) is feasible. By strong duality, we can write:

$$Q(\mathbf{x}, \mathbf{b}) = \max_{\mathbf{p}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p} \quad \text{s.t.} \quad \mathbf{B}^T \mathbf{p} \leq \mathbf{d}. \quad (8)$$

Thus, problem (5) is equivalent to:

$$\min_{\mathbf{x} \in S} \left[\mathbf{c}^T \mathbf{x} + \max_{\mathbf{b} \in \mathcal{B}, \mathbf{p} : \mathbf{B}^T \mathbf{p} \leq \mathbf{d}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p} \right]. \quad (9)$$

Proposition 2.1. *For any $\mathbf{b} \in \mathcal{B}$, $Q(\mathbf{x}, \mathbf{b})$ is a convex function in \mathbf{x} . For any $\mathbf{x} \in S$, $Q(\mathbf{x}, \mathbf{b})$ is a convex function in \mathbf{b} .*

Proof. The proof of the first claim is familiar from the stochastic programming literature: let $\mathbf{b} \in \mathcal{B}$ be fixed, let $\mathbf{x}_1, \mathbf{x}_2 \in S$, and $\lambda \in [0, 1]$. If, for either of the values \mathbf{x}_1 or \mathbf{x}_2 , problem (2) (the primal recourse problem) is infeasible, the relationship

$$Q(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2, \mathbf{b}) \leq \lambda Q(\mathbf{x}_1, \mathbf{b}) + (1 - \lambda) Q(\mathbf{x}_2, \mathbf{b})$$

holds trivially. Otherwise, $Q(\mathbf{x}_i, \mathbf{b})$ is finite (due to the sufficiently expensive recourse assumption), with the optimal value of problem (2) attained at \mathbf{y}_i , for $i = 1, 2$. Then $\mathbf{y} = \lambda\mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2$ is feasible for problem (2) with $\mathbf{x} = \lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$, and so

$$Q(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2, \mathbf{b}) \leq \mathbf{d}^T(\lambda\mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2) = \lambda\mathbf{d}^T\mathbf{y}_1 + (1 - \lambda)\mathbf{d}^T\mathbf{y}_2 = \lambda Q(\mathbf{x}_1, \mathbf{b}) + (1 - \lambda)Q(\mathbf{x}_2, \mathbf{b}),$$

proving the first claim.

The second claim can be easily established by analyzing the dual recourse problem (8). \square

Throughout the rest of the paper, we make the following assumption for ease of presentation.

Assumption 2.2. *Problem (5) possesses relatively complete recourse, i.e., problem (2) is feasible for all $\mathbf{x} \in S$ and $\mathbf{b} \in \mathcal{B}$.*

Together with the earlier assumption on feasibility of (8), Assumption 2.2 implies that $Q(\mathbf{x}, \mathbf{b})$ is finite for all $\mathbf{x} \in S$ and $\mathbf{b} \in \mathcal{B}$.

3 A Cutting-Plane Approach: Kelley's Algorithm

In this section, we describe a cutting-plane algorithm for solving problem (5) based on Kelley's algorithm, originally proposed in [26]. Kelley's algorithm is designed to minimize a linear objective function over a compact convex feasible region that is complex (possibly described by an infinite number of constraints) or given only by a separation oracle, i.e., a subroutine that, given a point in the variable space, either correctly asserts that the point is feasible or returns the normal vector and intercept of some hyperplane that strictly separates the point from the feasible region. At each iteration, the algorithm maintains a polyhedral outer approximation of the feasible region; the objective function is minimized over the approximation, and if the resultant solution is infeasible, adds a linear inequality (cut) derived from the separating hyperplane to the description of the polyhedral approximation of the feasible region, thus improving the approximation. For problems with feasible regions described by a (possibly infinite) family of differentiable convex inequality constraints, cuts can be generated using gradients of the violated constraints; subgradients can be used for convex non-differentiable functions. Recall that vector \mathbf{g} is a *subgradient* of a convex function $f(\mathbf{x})$ at $\tilde{\mathbf{x}}$ (denoted $\mathbf{g} \in \partial f(\tilde{\mathbf{x}})$) if the following subgradient inequality is satisfied:

$$f(\mathbf{x}) \geq f(\tilde{\mathbf{x}}) + \mathbf{g}^T(\mathbf{x} - \tilde{\mathbf{x}}) \quad \forall \mathbf{x}.$$

Problem (5) has a simple feasible region S , but the objective function $\mathbf{c}^T\mathbf{x} + \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{b})$ is complex. Let us define

$$Q(\mathbf{x}) = \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{b}) = \max_{\mathbf{b} \in \mathcal{B}, \mathbf{p}: \mathbf{B}^T\mathbf{p} \leq \mathbf{d}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T\mathbf{p}. \quad (10)$$

(Following [12], we can view the problem $\max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{b})$ as the *adversarial* problem, since its goal is to find the worst-case requirement vector for the first-stage decision \mathbf{x} .) Then (5) can be re-written as

$$\begin{aligned} \min_{\mathbf{x}, \alpha} \quad & \mathbf{c}^T\mathbf{x} + \alpha \\ \text{s.t.} \quad & \alpha \geq Q(\mathbf{x}) \\ & \mathbf{x} \in S. \end{aligned} \quad (11)$$

In light of the first claim of Proposition 2.1, the function $Q(x)$ is convex, making problem (11) suitable for application of Kelley’s algorithm. In our implementation of the algorithm, we maintain a piecewise linear lower approximation of the function $Q(x)$. The approximation is improved by adding cuts derived using values and subgradients of this function. In addition, we will maintain a lower bound and upper bound on the optimal objective function value; let these be denoted L and U , respectively.

Computing the value of function $Q(\mathbf{x}) = \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{b})$ is not an easy task, in general: since $Q(\mathbf{x}, \mathbf{b})$ is convex in \mathbf{b} for any \mathbf{x} , it requires maximization of a convex function. We postpone the discussion of this issue until Sections 4 and 5, for simple and general recourse cases, respectively. However, once the “worst-case” right-hand side for a given value of the first-stage decision is found, the subgradient of (10) can be found easily.

Lemma 3.1. *Let $\tilde{\mathbf{b}} \in \arg \max_{\mathbf{b} \in \mathcal{B}} Q(\tilde{\mathbf{x}}, \mathbf{b})$. Furthermore, let $\tilde{\mathbf{p}}$ be an optimal solution of (8) with $(\mathbf{x}, \mathbf{b}) = (\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$. Then $-\mathbf{A}^T \tilde{\mathbf{p}} \in \partial Q(\tilde{\mathbf{x}})$.*

Proof. For an arbitrary \mathbf{x} ,

$$\begin{aligned} Q(\mathbf{x}) &\geq Q(\mathbf{x}, \tilde{\mathbf{b}}) \geq (\tilde{\mathbf{b}} - \mathbf{A}\mathbf{x})^T \tilde{\mathbf{p}} \\ &= Q(\tilde{\mathbf{x}}, \tilde{\mathbf{b}}) + (-\mathbf{A}^T \tilde{\mathbf{p}})^T (\mathbf{x} - \tilde{\mathbf{x}}) \\ &= Q(\tilde{\mathbf{x}}) + (-\mathbf{A}^T \tilde{\mathbf{p}})^T (\mathbf{x} - \tilde{\mathbf{x}}), \end{aligned}$$

proving the claim. □

We refer to the vector $\tilde{\mathbf{p}}$ as in Lemma 3.1 as the dual recourse vector (for $\tilde{\mathbf{x}}$).

We can now specify Kelley’s algorithm for problem (11):

Algorithm 3.2 (Kelley’s Algorithm for Robust Linear Program with Recourse).

Initialization *Let $Q_0(\mathbf{x})$ be the initial piecewise linear lower approximation of $Q(\mathbf{x})$. Set $L = -\infty$ and $U = \infty$; $t = 0$.*

Iteration t *Given L, U , and $Q_t(\mathbf{x})$,*

Step 1 *Solve $\min_{\mathbf{x}, \alpha} \{\mathbf{c}^T \mathbf{x} + \alpha : \mathbf{x} \in S, \alpha \geq Q_t(\mathbf{x})\}$:*

$$\begin{aligned} \min_{\mathbf{x}, \alpha} \quad & \mathbf{c}^T \mathbf{x} + \alpha \\ \text{s.t.} \quad & \alpha + \mathbf{p}_l^T \mathbf{A}\mathbf{x} \geq \mathbf{b}_l^T \mathbf{p}_l, \quad l = 1, \dots, t-1 \\ & \alpha \geq Q_0(\mathbf{x}) \\ & \mathbf{x} \in S. \end{aligned} \tag{12}$$

Let (\mathbf{x}_t, α_t) be an optimal solution and let $L = \mathbf{c}^T \mathbf{x}_t + \alpha_t = \mathbf{c}^T \mathbf{x}_t + Q_t(\mathbf{x}_t)$.

Step 2 *Compute $Q(\mathbf{x}_t)$, let \mathbf{b}_t and \mathbf{p}_t be the corresponding worst-case right-hand-side and dual recourse vector, respectively. Let $U = \min\{U, \mathbf{c}^T \mathbf{x}_t + Q(\mathbf{x}_t)\}$. If $U - L$ is sufficiently small, stop and return \mathbf{x}_t as an approximate solution.*

Step 3 *Define*

$$Q_{t+1}(\mathbf{x}) = \max\{Q_t(\mathbf{x}), \mathbf{p}_t^T (\mathbf{b}_t - \mathbf{A}\mathbf{x})\}.$$

Step 4 Set $t \leftarrow t + 1$.

Observe that, for a given \mathbf{x} , the function $\max_{\mathbf{b} \in \mathcal{B}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p}$ is convex in \mathbf{p} , and therefore, under Assumption 2.2, problem (11) can be re-written as the following *master problem*:

$$\begin{aligned} \min_{\mathbf{x}, \alpha} \quad & \mathbf{c}^T \mathbf{x} + \alpha \\ \text{s.t.} \quad & \alpha \geq \max_{\mathbf{b} \in \mathcal{B}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p}_k, \quad k = 1, \dots, K \\ & \mathbf{x} \in S, \end{aligned} \tag{13}$$

where \mathbf{p}_k , $k = 1, \dots, K$ are the extreme points of $\{\mathbf{p} : \mathbf{B}^T \mathbf{p} \leq \mathbf{d}\}$. Algorithm 3.2 can be seen as a variant of Bender's decomposition for problem (13), with relaxed master problem (12). Bender's decomposition is typically presented for the stochastic programming problem (4) [6]. (Bertsimas and Tsitsiklis provide an introduction to these techniques in [10]; the reader is also referred to Birge and Louveaux [13] and Kall and Wallace [25] for an extensive treatment of these methods in the context of stochastic optimization.) The corresponding master problem can be written as:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} + \sum_{\omega=1}^{\Omega} \pi_{\omega} Z_{\omega} \\ \text{s.t.} \quad & Z_{\omega} \geq \mathbf{p}_k^T (\mathbf{b}_{\omega} - \mathbf{A}\mathbf{x}) \quad \forall k, \omega \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{14}$$

where $\omega = 1, \dots, \Omega$ are the scenarios. Here, at each iteration a relaxed master problem is solved to obtain a first-stage solution $\tilde{\mathbf{x}}$ and the corresponding value of the recourse function \tilde{Z}_{ω} when scenario ω is realized. To then check if this solution is optimal for the full master problem or to apply a cut to the expected recourse function $\sum_{\omega=1}^{\Omega} \pi_{\omega} Q(\mathbf{x}, \mathbf{b}_{\omega})$, one needs to solve the recourse problem (8) for each scenario $\omega = 1, \dots, \Omega$. While these problems are similar to each other and each can be solved efficiently by applying, for instance, the dual simplex method, the large number of subproblems is a drawback in accurately solving the stochastic programming counterpart of problem (1) in many real-life settings. In contrast, Algorithm 3.2, which applies a similar technique to the adjustable robust counterpart of problem (1), involves solving only one subproblem per iteration — a computationally simpler task so long the relevant subproblem can be easily identified. The methods for identifying the subproblem are discussed in Sections 4 and 5.

We note that if Assumption 2.2 does not hold, the algorithm can be modified by inclusion of *feasibility cuts*, and can be viewed as a variant of delayed constraint generation for solving the following modification of (13):

$$\begin{aligned} \min_{\mathbf{x}, \alpha} \quad & \mathbf{c}^T \mathbf{x} + \alpha \\ \text{s.t.} \quad & \alpha \geq \max_{\mathbf{b} \in \mathcal{B}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p}_k, \quad k = 1, \dots, K \\ & 0 \geq \max_{\mathbf{b} \in \mathcal{B}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{d}_l, \quad l = 1, \dots, L \\ & \mathbf{x} \in S, \end{aligned}$$

where \mathbf{p}_k , $k = 1, \dots, K$ are the extreme points, and \mathbf{d}_l , $l = 1, \dots, L$ are the extreme rays of $\{\mathbf{p} : \mathbf{B}^T \mathbf{p} \leq \mathbf{d}\}$.

To obtain further perspective on Algorithm 3.2, recall that $Q(\mathbf{x}, \mathbf{b})$ is convex in \mathbf{b} . Then, if the

uncertainty set \mathcal{B} is compact, problem (11) can be re-written as follows:

$$\begin{aligned} \min_{\mathbf{x}, \beta} \quad & \mathbf{c}^T \mathbf{x} + \alpha \\ \text{s.t.} \quad & \alpha \geq Q(\mathbf{x}, \mathbf{b}_e), \mathbf{b}_e \in E(\mathcal{B}) \\ & \mathbf{x} \in S, \end{aligned}$$

where $E(\mathcal{B})$ denotes the set of extreme points of \mathcal{B} (or of the convex hull of \mathcal{B}). Furthermore, using Assumption 2.2 again, an equivalent formulation is

$$\begin{aligned} \min_{\mathbf{x}, \beta} \quad & \mathbf{c}^T \mathbf{x} + \alpha \\ \text{s.t.} \quad & \alpha \geq (\mathbf{b}_e - \mathbf{A}\mathbf{x})^T \mathbf{p}_k, k = 1, \dots, K, \mathbf{b}_e \in E(\mathcal{B}) \\ & \mathbf{x} \in S. \end{aligned} \tag{15}$$

At each iteration, Kelley's algorithm adds one of the constraints of (15) as a cut. If the uncertainty set \mathcal{B} is polyhedral (e.g., if it is specified by equations (6)–(7)), (15) has a finite number of constraints, thus assuring convergence of Algorithm 3.2 after a finite number of iterations. In the worst case, Algorithm 3.2 may need to generate cuts corresponding to all non-redundant constraints of (15) to find an optimal solution (depending on the structure of the problem, there may be a connection between dual recourse vectors \mathbf{p} that can correspond to a particular value of $\mathbf{b} \in E(\mathcal{B})$ for all $\mathbf{x} \in S$, rendering some constraints of (15) redundant). In practice, however, algorithms of this type often perform better than this worst-case analysis suggests, finding approximate solutions after a small number of iterations. Sections 6 and 7 assess the number of iterations needed to solve instances of problem (15) to optimality for two example problems.

To conclude this section, we present a direct reformulation of the adjustable robust linear program (5) with \mathcal{B} given by (6)–(7) as a large-scale linear programming problem. In the proof, we refer to the set

$$\mathcal{Z}' = \left\{ \mathbf{z}' : \sum_{i=1}^m z'_i \leq \Gamma, 0 \leq z'_i \leq 1, i = 1, \dots, m \right\}. \tag{16}$$

Theorem 3.3. *If a robust linear program (5) with general recourse and \mathcal{B} given by (6)–(7) satisfies Assumption 2.2, it can be written as*

$$\begin{aligned} \min_{\mathbf{x}, \alpha, \mu, \lambda} \quad & \mathbf{c}^T \mathbf{x} + \alpha \\ \text{s.t.} \quad & \alpha \geq (\bar{\mathbf{b}} - \mathbf{A}\mathbf{x})^T \mathbf{p}_k + \lambda_k \Gamma + \sum_{i=1}^m \mu_{ik} \quad \forall k \\ & \lambda_k + \mu_{ik} \geq \hat{b}_i |p_{ik}| \quad \forall i, k \\ & \lambda_k, \mu_{ik} \geq 0 \quad \forall i, k \\ & \mathbf{x} \in S, \end{aligned} \tag{17}$$

where $\mathbf{p}_1, \dots, \mathbf{p}_K$ are (all) the extreme points of the set $\{\mathbf{p} : \mathbf{B}^T \mathbf{p} \leq \mathbf{d}\}$.

Proof. Recall that problem (5) can be written in the form (13).

Given \mathbf{x} and \mathbf{p} ,

$$\max_{\mathbf{b} \in \mathcal{B}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p} = (\bar{\mathbf{b}} - \mathbf{A}\mathbf{x})^T \mathbf{p} + \max_{\mathbf{z}' \in \mathcal{Z}'} \sum_{i=1}^m |p_i| \hat{b}_i z'_i,$$

with \mathcal{Z}' defined by (16). Since \mathcal{Z}' is nonempty and bounded, strong duality holds, which yields:

$$\begin{aligned} \max_{\mathbf{b} \in \mathcal{B}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p} = (\bar{\mathbf{b}} - \mathbf{A}\mathbf{x})^T \mathbf{p} + \min_{\lambda, \mu} \lambda \Gamma + \sum_{i=1}^m \mu_i \\ \text{s.t. } \lambda + \mu_i \geq \widehat{b}_i |p_i|, \quad i = 1, \dots, m \\ \lambda, \mu_i \geq 0, \quad i = 1, \dots, m. \end{aligned} \quad (18)$$

Combining this with formulation (13) yields problem (17). \square

The above approach to solving the robust problem is only practical if the set of extreme points of $\{\mathbf{p} : \mathbf{B}^T \mathbf{p} \leq \mathbf{d}\}$ is known and is relatively small in size.

4 Analysis of Robust Linear Programs with Simple Recourse

An important special case of linear programs with recourse are those with *simple recourse*, where the decision-maker is able to address excess or shortage for each of the requirements independently. For instance, he might pay a unit shortage penalty s_i for falling short of the uncertain target b_i or a unit holding cost h_i for exceeding the uncertain target b_i , for each i . We describe an application to multi-item newsvendor problems in Section 6.

The deterministic model with simple recourse can be formulated as:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} + \mathbf{s}^T \mathbf{y}^- + \mathbf{h}^T \mathbf{y}^+ \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{y}^- - \mathbf{y}^+ = \mathbf{b} \\ & \mathbf{x} \in S, \mathbf{y}^-, \mathbf{y}^+ \geq \mathbf{0}, \end{aligned}$$

and the recourse function of equation (2) becomes:

$$\begin{aligned} Q(\mathbf{x}, \mathbf{b}) = \min \quad & \mathbf{s}^T \mathbf{y}^- + \mathbf{h}^T \mathbf{y}^+ \\ \text{s.t.} \quad & \mathbf{y}^- - \mathbf{y}^+ = \mathbf{b} - \mathbf{A}\mathbf{x} \\ & \mathbf{y}^-, \mathbf{y}^+ \geq \mathbf{0}. \end{aligned} \quad (19)$$

Note that Assumption 2.2 holds for (19). We will further require that $\mathbf{s} + \mathbf{h} \geq \mathbf{0}$ to ensure sufficiently expensive recourse. It is straightforward to see that $Q(\mathbf{x}, \mathbf{b})$ is available in closed form:

$$Q(\mathbf{x}, \mathbf{b}) = \sum_{i=1}^m [s_i \cdot \max\{0, b_i - (\mathbf{A}\mathbf{x})_i\} + h_i \cdot \max\{0, (\mathbf{A}\mathbf{x})_i - b_i\}].$$

However, we will focus on problem (19) to develop an efficient method for computing $Q(\mathbf{x})$. We obtain an equivalent characterization of the recourse function by invoking strong duality:

$$\begin{aligned} Q(\mathbf{x}, \mathbf{b}) = \max \quad & (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p} \\ \text{s.t.} \quad & -\mathbf{h} \leq \mathbf{p} \leq \mathbf{s}. \end{aligned} \quad (20)$$

Therefore, in this section we will be developing methods to solve:

$$\min_{\mathbf{x} \in S} \left[\mathbf{c}^T \mathbf{x} + \max_{\mathbf{b} \in \mathcal{B}, -\mathbf{h} \leq \mathbf{p} \leq \mathbf{s}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p} \right], \quad (21)$$

where \mathcal{B} has been defined in equations (6)–(7).

The following theorem provides a simple method for computing $Q(\mathbf{x})$ in problems with simple recourse.

Theorem 4.1. Given \mathbf{x} , define for $i = 1, \dots, m$,

$$\Delta_i = \max \left\{ (\bar{b}_i + \hat{b}_i - (\mathbf{A}\mathbf{x})_i)s_i, ((\mathbf{A}\mathbf{x})_i - \bar{b}_i + \hat{b}_i)h_i \right\} - \max \left\{ (\bar{b}_i - (\mathbf{A}\mathbf{x})_i)s_i, ((\mathbf{A}\mathbf{x})_i - \bar{b}_i)h_i \right\}.$$

Let \mathcal{I} be the set of indices corresponding to the Γ largest Δ_i 's (ties can be broken arbitrarily). Then $\mathcal{Q}(\mathbf{x}) = \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{b})$ with $Q(\mathbf{x}, \mathbf{b})$ given by (20) and \mathcal{B} given by (6)–(7) verifies:

$$\begin{aligned} \mathcal{Q}(\mathbf{x}) = \sum_{i \in \mathcal{I}} \max \left\{ (\bar{b}_i + \hat{b}_i - (\mathbf{A}\mathbf{x})_i)s_i, ((\mathbf{A}\mathbf{x})_i - \bar{b}_i + \hat{b}_i)h_i \right\} \\ + \sum_{i \notin \mathcal{I}} \max \left\{ (\bar{b}_i - (\mathbf{A}\mathbf{x})_i)s_i, ((\mathbf{A}\mathbf{x})_i - \bar{b}_i)h_i \right\}. \end{aligned} \quad (22)$$

Proof. We note that for any first-stage decision vector \mathbf{x} :

$$\begin{aligned} \mathcal{Q}(\mathbf{x}) &= \max_{\mathbf{b} \in \mathcal{B}} \max_{-\mathbf{h} \leq \mathbf{p} \leq \mathbf{s}} (\mathbf{b} - \mathbf{A}\mathbf{x})^T \mathbf{p}, \\ &= \max_{\mathbf{b} \in \mathcal{B}} \sum_{i=1}^m \max \left\{ (b_i - (\mathbf{A}\mathbf{x})_i)s_i, ((\mathbf{A}\mathbf{x})_i - b_i)h_i \right\} \\ &= \sum_{i=1}^m \max \left\{ (\bar{b}_i - (\mathbf{A}\mathbf{x})_i)s_i, ((\mathbf{A}\mathbf{x})_i - \bar{b}_i)h_i \right\} + \max_{\mathbf{z}' \in \mathcal{Z}'} \sum_{i=1}^m \Delta_i z'_i, \end{aligned} \quad (23)$$

where \mathcal{Z}' is defined in (16). The last equality is obtained by observing that the expression in (23) is convex in \mathbf{b} , and thus the worst-case value of \mathbf{b} that attains the maximum can be found at an extreme point of \mathcal{B} . The extreme points of \mathcal{B} can be enumerated by letting Γ components of \mathbf{b} deviate up or down (to their highest or lowest values), while keeping the remaining components at their nominal values. Whether the worst case is reached when b_i deviates up or down (to its highest or lowest value) is captured by the value of Δ_i . It then follows that $\max_{\mathbf{z}' \in \mathcal{Z}'} \sum_{i=1}^m \Delta_i z'_i$ is equal to $\sum_{i \in \mathcal{I}} \Delta_i$, where \mathcal{I} is as specified in the theorem. \square

Corollary 4.2. Given \mathbf{x} , corresponding worst-case demand \mathbf{b} can be determined as follows: for $i = 1, \dots, m$

$$b_i = \begin{cases} \bar{b}_i + \hat{b}_i & \text{if } i \in \mathcal{I} \text{ and } (\bar{b}_i + \hat{b}_i - (\mathbf{A}\mathbf{x})_i)s_i \geq ((\mathbf{A}\mathbf{x})_i - \bar{b}_i + \hat{b}_i)h_i \\ \bar{b}_i - \hat{b}_i & \text{if } i \in \mathcal{I} \text{ and } (\bar{b}_i + \hat{b}_i - (\mathbf{A}\mathbf{x})_i)s_i < ((\mathbf{A}\mathbf{x})_i - \bar{b}_i + \hat{b}_i)h_i \\ \bar{b}_i & \text{if } i \notin \mathcal{I}. \end{cases} \quad (24)$$

Corresponding dual recourse vector \mathbf{p} can be determined as follows: for $i = 1, \dots, m$

$$p_i = \begin{cases} s_i & \text{if } i \in \mathcal{I} \text{ and } (\bar{b}_i + \hat{b}_i - (\mathbf{A}\mathbf{x})_i)s_i \geq ((\mathbf{A}\mathbf{x})_i - \bar{b}_i + \hat{b}_i)h_i \\ -h_i & \text{if } i \in \mathcal{I} \text{ and } (\bar{b}_i + \hat{b}_i - (\mathbf{A}\mathbf{x})_i)s_i < ((\mathbf{A}\mathbf{x})_i - \bar{b}_i + \hat{b}_i)h_i \\ s_i & \text{if } i \notin \mathcal{I} \text{ and } (\bar{b}_i - (\mathbf{A}\mathbf{x})_i)s_i \geq ((\mathbf{A}\mathbf{x})_i - \bar{b}_i)h_i \\ -h_i & \text{if } i \notin \mathcal{I} \text{ and } (\bar{b}_i - (\mathbf{A}\mathbf{x})_i)s_i < ((\mathbf{A}\mathbf{x})_i - \bar{b}_i)h_i. \end{cases} \quad (25)$$

The subgradient of $\mathcal{Q}(\mathbf{x})$ can now be computed as in Lemma 3.1, allowing the implementation of Kelley's algorithm 3.2.

5 Analysis of Robust Linear Programs with General Recourse

In this section we return to the analysis of the robust linear program with general recourse function (2). Without any assumptions on the structure of recourse matrix \mathbf{B} , evaluation of $Q(\mathbf{x})$ of (10) can no longer be done in closed form, as was the case with simple recourse. We present a general approach for computing the value of $Q(\mathbf{x})$, along with corresponding worst-case value of \mathbf{b} and dual recourse variable \mathbf{p} , via a mixed integer programming problem, under the assumption of relatively complete recourse.

Theorem 5.1. *Suppose Assumption 2.2 holds for problems (5). Given \mathbf{x} , $Q(\mathbf{x}) = \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{b})$ with $Q(\mathbf{x}, \mathbf{b})$ given by (8) and \mathcal{B} given by (6)–(7) can be computed as*

$$\begin{aligned}
 Q(\mathbf{x}) = \max_{\mathbf{p}^+, \mathbf{p}^-, \mathbf{q}^+, \mathbf{q}^-, \mathbf{r}^+, \mathbf{r}^-} & (\bar{\mathbf{b}} - \mathbf{A}\mathbf{x})^T(\mathbf{p}^+ - \mathbf{p}^-) + \widehat{\mathbf{b}}^T(\mathbf{q}^+ + \mathbf{q}^-) \\
 \text{s.t.} & \mathbf{B}^T(\mathbf{p}^+ - \mathbf{p}^-) \leq \mathbf{d} \\
 & \mathbf{0} \leq \mathbf{q}^+ \leq \mathbf{p}^+ \\
 & \mathbf{0} \leq \mathbf{q}^- \leq \mathbf{p}^- \\
 & \mathbf{q}^+ \leq M\mathbf{r}^+ \\
 & \mathbf{q}^- \leq M\mathbf{r}^- \\
 & \mathbf{e}^T(\mathbf{r}^+ + \mathbf{r}^-) \leq \Gamma \\
 & \mathbf{r}^+ + \mathbf{r}^- \leq \mathbf{e} \\
 & \mathbf{r}^+, \mathbf{r}^- \in \{0, 1\}^m \\
 & \mathbf{p}^+, \mathbf{p}^- \geq \mathbf{0},
 \end{aligned} \tag{26}$$

where \mathbf{e} is the vector of all ones and M is a sufficiently large positive number.

Proof. Introducing variables $\mathbf{p}^+, \mathbf{p}^- \geq \mathbf{0}$ and recalling definition of \mathcal{B} , problem (10) can be rewritten as

$$\begin{aligned}
 Q(\mathbf{x}) = \max_{\mathbf{p}^+, \mathbf{p}^-} & \left[(\bar{\mathbf{b}} - \mathbf{A}\mathbf{x})^T(\mathbf{p}^+ - \mathbf{p}^-) + \max_{\mathbf{z} \in \mathcal{Z}} \sum_{i=1}^m \widehat{b}_i(p_i^+ - p_i^-)z_i \right] \\
 \text{s.t.} & \mathbf{B}^T(\mathbf{p}^+ - \mathbf{p}^-) \leq \mathbf{d} \\
 & \mathbf{p}^+, \mathbf{p}^- \geq \mathbf{0},
 \end{aligned} \tag{27}$$

where \mathcal{Z} was defined in equation (7). Note that for a given \mathbf{p}^+ and \mathbf{p}^- , the inner maximization problem or (27) is a linear program in variables \mathbf{z} , and its optimal solution can be found at one of the extreme points of the set \mathcal{Z} , which have the form $z_i \in \{-1, 0, 1\} \forall i$ and $\sum_{i=1}^m |z_i| = \Gamma$. Therefore, we can re-write the inner maximization problem of (27) as the following integer program in variables $\mathbf{r}^+, \mathbf{r}^-$:

$$\begin{aligned}
 \max_{\mathbf{r}^+, \mathbf{r}^-} & \sum_{i=1}^m \widehat{b}_i(p_i^+ - p_i^-)(r_i^+ - r_i^-) \\
 \text{s.t.} & \mathbf{e}^T(\mathbf{r}^+ + \mathbf{r}^-) \leq \Gamma \\
 & \mathbf{r}^+ + \mathbf{r}^- \leq \mathbf{e} \\
 & \mathbf{r}^+, \mathbf{r}^- \in \{0, 1\}^m.
 \end{aligned}$$

Substituting the above into (27), we obtain the following equivalent formulation:

$$\begin{aligned}
Q(\mathbf{x}) = \max_{\mathbf{p}^\pm, \mathbf{r}^\pm} & (\bar{\mathbf{b}} - \mathbf{A}\mathbf{x})^T(\mathbf{p}^+ - \mathbf{p}^-) + \sum_{i=1}^m \widehat{b}_i(p_i^+ - p_i^-)(r_i^+ - r_i^-) \\
\text{s.t.} & \mathbf{B}^T(\mathbf{p}^+ - \mathbf{p}^-) \leq \mathbf{d} \\
& \mathbf{e}^T(\mathbf{r}^+ + \mathbf{r}^-) \leq \Gamma \\
& \mathbf{r}^+ + \mathbf{r}^- \leq \mathbf{e} \\
& \mathbf{r}^+, \mathbf{r}^- \in \{0, 1\}^m \\
& \mathbf{p}^+, \mathbf{p}^- \geq \mathbf{0}.
\end{aligned} \tag{28}$$

Finally, to linearize the objective function of (28), we introduce variables $q_i^+ = p_i^+ r_i^+$ and $q_i^- = p_i^- r_i^- \forall i$ (note that we can assume $p_i^+ r_i^- = p_i^- r_i^+ = 0 \forall i$ without loss of generality). Making the substitution in (28) and adding appropriate forcing constraints results in problem (26). \square

Corollary 5.2. *Given \mathbf{x} , let $(\mathbf{p}^\pm, \mathbf{r}^\pm, \mathbf{q}^\pm)$ solve (26). Then corresponding worst-case value of \mathbf{b} can be determined as follows: $b_i = \bar{b}_i + b_i(r_i^+ - r_i^-)$, $i = 1, \dots, m$, and corresponding dual recourse vector \mathbf{p} can be determined as $\mathbf{p} = \mathbf{p}^+ - \mathbf{p}^-$.*

The subgradient of $Q(\mathbf{x})$ can now be computed as in Lemma 3.1.

6 Computational Results: Newsvendor Problem

In this section we test the robust methodology on a multi-item newsvendor problem. The decision-maker orders perishable items subject to a capacity constraint, faces uncertain demand, and incurs surplus and shortage costs for each item at the end of the time period. His goal is to minimize total cost. We use the following notation:

- n : the number of items,
- c_i : the unit ordering cost of item i ,
- h_i : the unit holding cost of item i ,
- s_i : the unit shortage cost of item i ,
- b_i : the demand for item i ,
- A : the purchasing budget.

The deterministic problem can be formulated as:

$$\begin{aligned}
\min & \mathbf{c}^T \mathbf{x} + \sum_{i=1}^n \max \{s_i(b_i - x_i), h_i(x_i - b_i)\} \\
\text{s.t.} & \mathbf{c}^T \mathbf{x} \leq A \\
& \mathbf{x} \geq \mathbf{0}
\end{aligned}$$

or equivalently, as:

$$\begin{aligned}
\min & \mathbf{c}^T \mathbf{x} + \mathbf{s}^T \mathbf{y}^- + \mathbf{h}^T \mathbf{y}^+ \\
\text{s.t.} & \mathbf{x} + \mathbf{y}^- - \mathbf{y}^+ = \mathbf{b}, \\
& \mathbf{c}^T \mathbf{x} \leq A, \\
& \mathbf{x} \geq \mathbf{0}.
\end{aligned} \tag{29}$$

Problem (29) is an example of a linear programming problem with simple recourse and therefore can be analyzed using the techniques described in Sections 3 and 4. We consider a case with $n = 50$ items and budget $A = 100n$, with ordering cost $c_i = 1$, nominal demand $\bar{b}_i = 8 + 2i$, and maximum deviation of the demand from its nominal value $\hat{b}_i = 0.5 \cdot \bar{b}_i$, for each $i = 1, \dots, 50$. We consider two different structures for the surplus and shortage penalties, resulting in two instances of problem (29). In the first instance, items with larger nominal demand (and thus wider demand variability by the above definitions of \bar{b} and \hat{b}) have larger surplus and shortage penalties than items with smaller nominal demand. In the second instance, surplus and shortage penalties follow the opposite pattern. In particular, the penalties for item i are shown in the table below:

| | Shortage penalty s_i | Surplus penalty h_i |
|------------|------------------------|-----------------------|
| Instance 1 | $2i$ | i |
| Instance 2 | $2(n + 1 - i)$ | $(n + 1 - i)$ |

We applied Algorithm 3.2 to problem (29). Since the recourse value in this problem is always nonnegative, we set $\mathcal{Q}_0(\mathbf{x}) \equiv 0$ in the initialization step. Step 2 of the algorithm was carried out as discussed in Theorem 4.1 and Corollary 4.2. Finally, we terminated the algorithm when $L = U$, solving the robust problem to optimality.

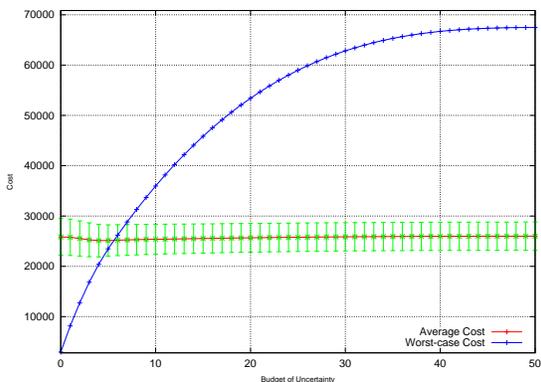
6.1 Analysis of Problem Solutions

To understand the effect of the budget of uncertainty Γ utilized by the decision-maker in the selection of the uncertainty set \mathcal{B} , we solved both instances of the newsvendor problem for values of Γ ranging from 0 to 50. Figure 1 summarizes our findings.

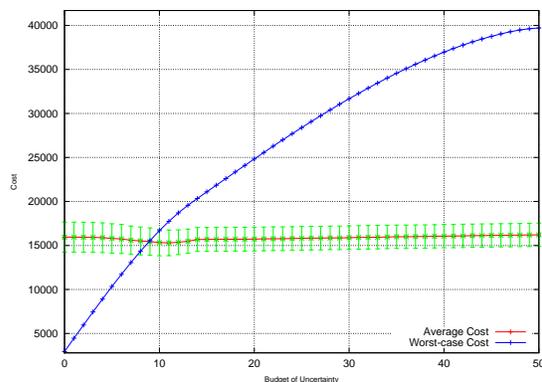
Figures 1(a) and 1(b) show the worst-case costs of the two instances (i.e., optimal objective value of (5)) as a function of Γ , which, as expected, increase as the solution becomes more conservative (the blue curves).

The decision-maker's choice of the uncertainty set (and in particular \bar{b}_i 's and \hat{b}_i 's) can reflect his knowledge or assumptions on the possible realizations of demands, including any information on the probability distribution when the demands are perceived by the decision maker as random variables. On the other hand, the goal of the robust approach is to be able to provide a solution that performs reasonably well even if the distribution is not known with any great deal of precision. To assess the performance of robust solutions, we considered an example in which demands are indeed random, each following an independent Normal distribution with mean \bar{b}_i and standard deviation $0.4 \cdot \hat{b}_i$ for each i (recall that $\hat{b}_i = 0.5 \cdot \bar{b}_i$). We created a sample of 5000 realizations of the demands. The resulting average costs of robust solutions are depicted in Figure 1 (the red curves in Figures 1(a) and 1(b) and, on a different scale, in Figures 1(c) and 1(d); the error bars reflect the sample standard deviations). For both instances of the newsvendor problem, we observe from Figures 1(c) and 1(d) that the average cost first decreases with Γ , as incorporating a small amount of uncertainty in the model yields more robust solutions, reaches its minimum, and starts increasing with Γ as the solution becomes overly conservative for the typical demand realization. In Figure 1(c), the best trade-off is reached at $\Gamma = 5$, and the average cost of corresponding robust solution achieves savings of 2.82% over solution obtained for $\Gamma = 0$ (i.e., solution targeted to satisfy the nominal demand $\bar{\mathbf{b}}$), while Figure 1(d) has an best trade-off at $\Gamma = 11$ and savings of 4.09%; both are consistent with the guidelines provided by Bertsimas and Sim in [8], namely, that the budget

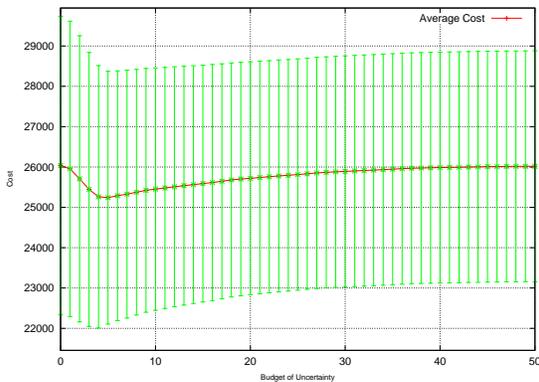
of uncertainty should be of the order of \sqrt{n} (here, $\sqrt{50} \approx 7.1$). Note that for the first instance, the worst case for $\Gamma = 5$ corresponds to the situation where the demand for the last 5 items (items 46 to 50) is equal to its highest value and demand for the other items is equal to its nominal value, which makes sense as the last 5 items have the largest shortage and holding penalties. In the second instance, the worst-case instance for $\Gamma = 11$ consists of demand for 11 products with mid-range penalties equal to its highest value.



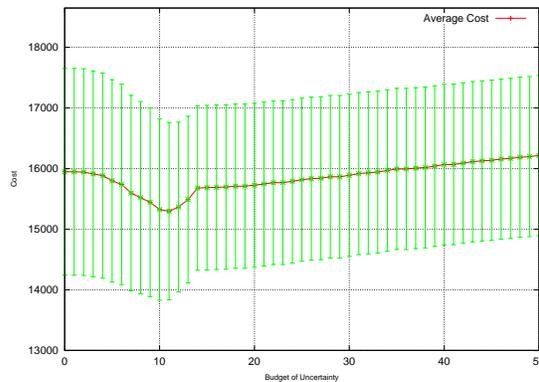
(a) Instance 1: worst-case cost vs. Γ .



(b) Instance 2: worst-case cost vs. Γ .



(c) Instance 1: average cost vs. Γ .



(d) Instance 2: average cost vs. Γ .

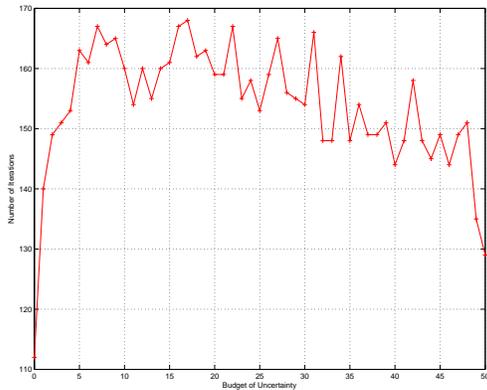
Figure 1: The impact of the budget of uncertainty on worst-case and average costs for two instances of the newsvendor problem.

6.2 Algorithmic Performance

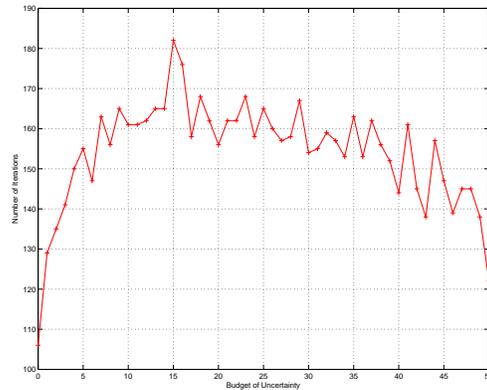
The algorithm was implemented using AMPL/CPLEX v.12.1 on a Linux machine with Intel Core Quad CPU at 2.50GHz.

Figure 2 illustrates the effect of the budget of uncertainty on both the number of iterations and the

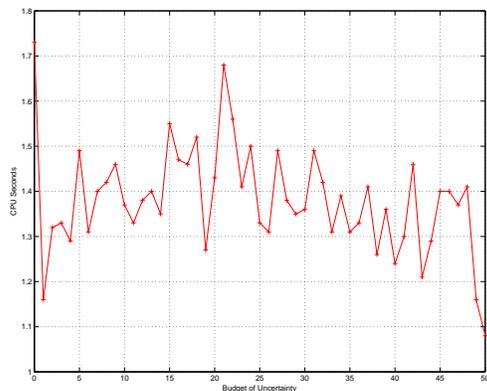
running time, in CPU seconds, of Algorithm 3.2. Neither the number of iterations nor the running time showed any particular dependence on Γ (although problems with very small and very large values of Γ appear easier to solve, perhaps due to relatively smaller numbers of extreme points of \mathcal{B} , and thus small number of non-redundant constraints in problem (15)). The maximum number of iterations needed for either problem instance was around 180, while the maximum running time was under 2 seconds.



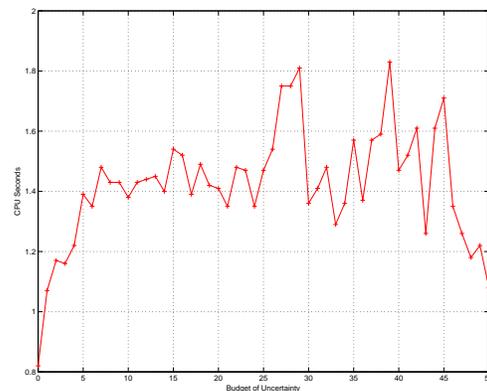
(a) Instance 1: Iteration count vs. Γ .



(b) Instance 2: Iteration count vs. Γ .



(c) Instance 1: Run time (CPU seconds) vs. Γ .

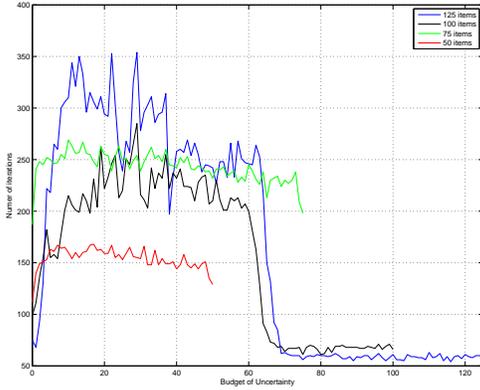


(d) Instance 2: Run time (CPU seconds) vs. Γ .

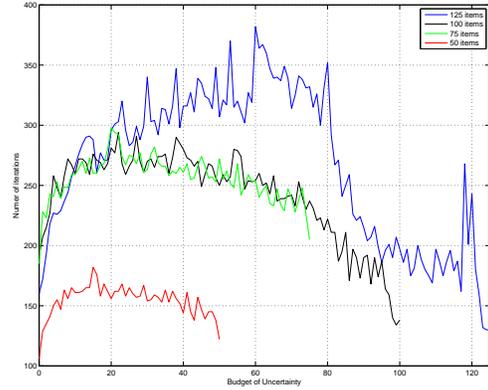
Figure 2: The impact of the budget of uncertainty on the number of iterations and run time (sec) for two instances of the multi-item newsvendor problem.

We next investigated how the size of the problem affects the number of iterations required by the algorithm, as well as its running time. We compared instances of the newsvendor problem with $n = 50, 75, 100, 125$ and $A = 100n$. One again, two cost structures were considered, and the problems were solved for all values of $\Gamma = 0, 1, \dots, n$.

Overall, the number of iterations required, as well as the running time of the algorithm, increase with the increase in problem size. (The number of iterations for $n = 125$ is above 360 for some



(a) Instance 1: Iteration count vs. Γ .



(b) Instance 2: Iteration count vs. Γ .

Figure 3: The impact of problem size on the number of iterations for two instances of the multi-item newsvendor problem.

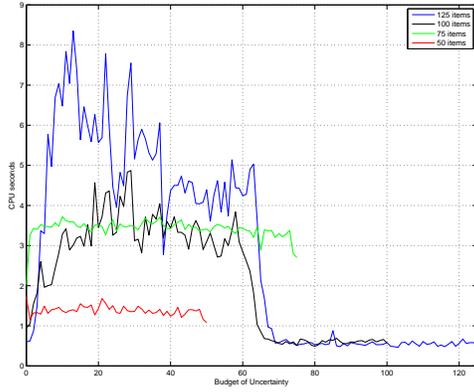
values of Γ in both instances.) Figures 3(a) and 3(b) illustrate the dependence of the number of iterations performed on the structure of the problem, reflected in this case by the difference in the cost vectors in the two instances. The number of constraints in problem (15) is the same in the two instances for the same values of n and Γ . However, in instance 1, the number of iterations required decreases significantly for larger values of n and Γ . This is likely due to the fact that the number of *non-redundant* constraints is significantly smaller for those problems (not surprising given the relationship between worst-case demand vectors \mathbf{b} and dual recourse vectors \mathbf{p} highlighted in Corollary 4.2).

Figures 4(a) and 4(b) demonstrate corresponding increases in the running time of the algorithm for increasing problem sizes. This overall increase can be attributed to two factors: the increase in the number of iterations, as well as the increase in the complexity of each iteration, as the number of variables in the subproblem solved, as well as the size of vector to be sorted to solve the adversarial problem, increases. Figures 4(c) and 4(d) depict the ratio between the running time and the number of iterations required for each instance solved. It appears that the increased complexity of each iteration contributes significantly to the increase in running times.

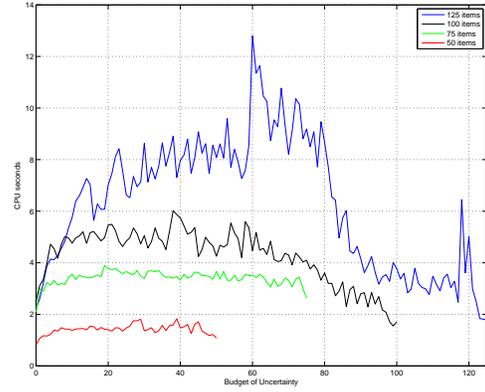
It should be noted that we did not make any special efforts to implement the individual steps of our algorithm efficiently. E.g., we did not take advantage (beyond what is done automatically by the CPLEX solver under default settings) of the fact that the linear program being solved at each iteration differs from the one solved in the previous iteration by just one additional constraint, and hence is likely easily handled by dual simplex algorithm. Such savings could potentially improve the running time of the algorithm, but not change the number of iterations required.

7 Computational Results: Production Planning Problem

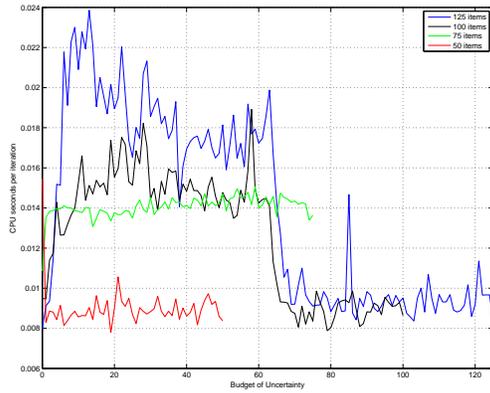
Here, we consider a production planning example where the demand is uncertain but must be met. Once demand has been revealed, the decision-maker has the option to buy additional raw materials



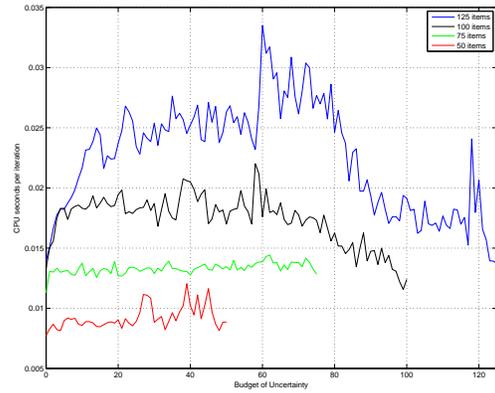
(a) Instance 1: Run time (CPU seconds) vs. Γ .



(b) Instance 2: Run time (CPU seconds) vs. Γ .



(c) Instance 1: CPU seconds per iteration vs. Γ .



(d) Instance 2: CPU seconds per iteration vs. Γ .

Figure 4: The impact of problem size on run time (sec) for two instances of the multi-item newsven-
dor problem.

at a higher cost and re-run the production process, so that demand for all products is satisfied. The goal is to minimize the ordering costs of raw materials and the production costs of finished products, as well as the inventory (or disposal) costs on the materials and products remaining at the end of the time period. We introduce the following notation:

m : the number of raw materials,
 n : the number of finished products,
 \mathbf{c} : the first-stage unit cost of the raw materials,
 \mathbf{d} : the second-stage unit cost of the raw materials,
 \mathbf{f} : the first-stage unit production cost,
 \mathbf{g} : the second-stage unit production cost,
 \mathbf{h} : the unit inventory cost of unused raw materials,
 \mathbf{k} : the unit inventory cost of unsold finished products,
 \mathbf{A} : the productivity matrix,
 \mathbf{b} : the demand for the finished products,
 \mathbf{x} : the raw materials purchased in the first stage,
 \mathbf{y} : the raw materials purchased in the second stage,
 \mathbf{u} : the products produced in the first stage,
 \mathbf{v} : the products produced in the second stage,

We assume that all coefficients of the matrix \mathbf{A} are nonnegative, as are all the costs. The deterministic problem can be formulated as:

$$\begin{aligned}
\min_{\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}} \quad & \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} + \mathbf{f}^T \mathbf{u} + \mathbf{g}^T \mathbf{v} + \mathbf{h}^T (\mathbf{x} + \mathbf{y} - \mathbf{A}(\mathbf{u} + \mathbf{v})) + \mathbf{k}^T (\mathbf{u} + \mathbf{v} - \mathbf{b}) \\
\text{s.t.} \quad & \mathbf{A} \mathbf{u} \leq \mathbf{x} \\
& \mathbf{A} \mathbf{u} + \mathbf{A} \mathbf{v} \leq \mathbf{x} + \mathbf{y} \\
& \mathbf{u} + \mathbf{v} \geq \mathbf{b} \\
& \mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v} \geq \mathbf{0}.
\end{aligned} \tag{30}$$

Note that, according to the formulation, raw materials can be purchased in the first stage of the time period and used in production in the second stage. This problem satisfies assumption of relatively complete recourse.

7.1 Analysis of Production Planning Problem

The robust production planning problem is as follows:

$$\begin{aligned}
\min_{\mathbf{x}, \mathbf{u}} \quad & \mathbf{c}^T \mathbf{x} + \mathbf{f}^T \mathbf{u} + \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{u}, \mathbf{b}) \\
\text{s.t.} \quad & \mathbf{A} \mathbf{u} \leq \mathbf{x} \\
& \mathbf{x}, \mathbf{u} \geq \mathbf{0},
\end{aligned} \tag{31}$$

where the recourse function, $Q(\mathbf{x}, \mathbf{u}, \mathbf{b})$ is given by

$$\begin{aligned}
Q(\mathbf{x}, \mathbf{u}, \mathbf{b}) = \bar{c}(\mathbf{x}, \mathbf{u}) - \mathbf{k}^T \mathbf{b} + \min_{\mathbf{y}, \mathbf{v}} \quad & (\mathbf{d} + \mathbf{h})^T \mathbf{y} + (\mathbf{g} - \mathbf{A}^T \mathbf{h} + \mathbf{k})^T \mathbf{v} \\
\text{s.t.} \quad & \mathbf{y} - \mathbf{A} \mathbf{v} \geq \mathbf{A} \mathbf{u} - \mathbf{x} \\
& \mathbf{v} \geq \mathbf{b} - \mathbf{u} \\
& \mathbf{y}, \mathbf{v} \geq \mathbf{0},
\end{aligned} \tag{32}$$

where $\bar{c}(\mathbf{x}, \mathbf{u}) = \mathbf{h}^T \mathbf{x} - \mathbf{h}^T \mathbf{A} \mathbf{u} + \mathbf{k}^T \mathbf{u}$. Since this form of the recourse function is slightly different from expression (2) in that the constraints are in inequality form, we begin by deriving a modification of the integer program of Theorem 5.1.

The dual of the recourse function is:

$$Q(\mathbf{x}, \mathbf{u}, \mathbf{b}) = \bar{c}(\mathbf{x}, \mathbf{u}) - \mathbf{k}^T \bar{\mathbf{b}} + \max_{\mathbf{q}, \mathbf{p}} (\mathbf{A}\mathbf{u} - \mathbf{x})^T \mathbf{q} + (\bar{\mathbf{b}} - \mathbf{u})^T \mathbf{p} \quad (33)$$

$$\text{s.t. } \mathbf{0} \leq \mathbf{q} \leq \mathbf{d} + \mathbf{h}$$

$$\mathbf{0} \leq \mathbf{p} \leq \mathbf{A}^T \mathbf{q} + \mathbf{g} - \mathbf{A}^T \mathbf{h} + \mathbf{k}.$$

To obtain $\mathcal{Q}(\mathbf{x}, \mathbf{u}) = \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{u}, \mathbf{b})$, where $Q(\mathbf{x}, \mathbf{u}, \mathbf{b})$ is defined by (33), we can write:

$$\mathcal{Q}(\mathbf{x}, \mathbf{u}) = \bar{c}(\mathbf{x}, \mathbf{u}) - \mathbf{k}^T \bar{\mathbf{b}} + \max_{\mathbf{q}, \mathbf{p}} \left((\mathbf{A}\mathbf{u} - \mathbf{x})^T \mathbf{q} + (\bar{\mathbf{b}} - \mathbf{u})^T \mathbf{p} + \max_{\mathbf{z} \in \mathcal{Z}} \sum_{i=1}^n (\hat{b}_i(p_i - k_i)z_i) \right)$$

$$\text{s.t. } \mathbf{0} \leq \mathbf{q} \leq \mathbf{d} + \mathbf{h}$$

$$\mathbf{0} \leq \mathbf{p} \leq \mathbf{A}^T \mathbf{q} + \mathbf{g} - \mathbf{A}^T \mathbf{h} + \mathbf{k}, \quad (34)$$

where \mathcal{Z} was defined in equation (7). Applying the same logic as in Theorem 5.1, for a given \mathbf{q} and \mathbf{p} , we can re-write the inner maximization problem of (34) as an integer programming problem in variables \mathbf{r}^+ and \mathbf{r}^- :

$$\max_{\mathbf{r}^+, \mathbf{r}^-} \sum_{i=1}^m \hat{b}_i(p_i - k_i)(r_i^+ - r_i^-)$$

$$\text{s.t. } \mathbf{e}^T(\mathbf{r}^+ + \mathbf{r}^-) \leq \Gamma, \quad (35)$$

$$\mathbf{r}^+ + \mathbf{r}^- \leq \mathbf{e},$$

$$\mathbf{r}^+, \mathbf{r}^- \in \{0, 1\}^m.$$

Substituting (35) into (34), we obtain:

$$\mathcal{Q}(\mathbf{x}, \mathbf{u}) = \bar{c}(\mathbf{x}, \mathbf{u}) - \mathbf{k}^T \bar{\mathbf{b}} + \max_{\mathbf{q}, \mathbf{p}, \mathbf{r}^\pm} (\mathbf{A}\mathbf{u} - \mathbf{x})^T \mathbf{q} + (\bar{\mathbf{b}} - \mathbf{u})^T \mathbf{p} + \sum_{i=1}^n (\hat{b}_i(p_i - k_i)(r_i^+ - r_i^-))$$

$$\text{s.t. } \mathbf{0} \leq \mathbf{q} \leq \mathbf{d} + \mathbf{h}, \quad (36)$$

$$\mathbf{0} \leq \mathbf{p} \leq \mathbf{A}^T \mathbf{q} + \mathbf{g} - \mathbf{A}^T \mathbf{h} + \mathbf{k},$$

$$\mathbf{e}^T(\mathbf{r}^+ + \mathbf{r}^-) \leq \Gamma,$$

$$\mathbf{r}^+ + \mathbf{r}^- \leq \mathbf{e},$$

$$\mathbf{r}^+, \mathbf{r}^- \in \{0, 1\}^n.$$

To remove the nonlinearity from the objective function, we introduce variables $s_i^+ = p_i r_i^+$ and $s_i^- = p_i r_i^- \forall i$. Substituting the above into (36) we conclude that we can calculate the value of $\mathcal{Q}(\mathbf{x}, \mathbf{u})$ by solving the following IP:

$$\bar{c}(\mathbf{x}, \mathbf{u}) - \mathbf{k}^T \bar{\mathbf{b}} + \max_{\mathbf{q}, \mathbf{p}, \mathbf{s}^\pm, \mathbf{r}^\pm} (\mathbf{A}\mathbf{u} - \mathbf{x})^T \mathbf{q} + (\bar{\mathbf{b}} - \mathbf{u})^T \mathbf{p} + \hat{\mathbf{b}}^T(\mathbf{s}^+ - \mathbf{s}^-) - \sum_{i=1}^n k_i \hat{b}_i(r_i^+ - r_i^-)$$

$$\text{s.t. } \mathbf{0} \leq \mathbf{q} \leq \mathbf{d} + \mathbf{h}$$

$$\mathbf{0} \leq \mathbf{p} \leq \mathbf{A}^T \mathbf{q} + \mathbf{g} - \mathbf{A}^T \mathbf{h} + \mathbf{k}$$

$$\mathbf{0} \leq \mathbf{s}^+ \leq \mathbf{p}^+$$

$$\mathbf{0} \leq \mathbf{s}^- \leq \mathbf{p}^- \quad (37)$$

$$\mathbf{s}^+ \leq M\mathbf{r}^+$$

$$\mathbf{s}^- \leq M\mathbf{r}^-$$

$$\mathbf{e}^T(\mathbf{r}^+ + \mathbf{r}^-) \leq \Gamma$$

$$\mathbf{r}^+ + \mathbf{r}^- \leq \mathbf{1}$$

$$\mathbf{r}^+, \mathbf{r}^- \in \{0, 1\}^n,$$

where M is a sufficiently large positive number.

Lastly, given (\mathbf{x}, \mathbf{u}) , let $(\mathbf{q}, \mathbf{p}, \mathbf{s}^\pm, \mathbf{r}^\pm)$ solve (37). The corresponding dual recourse vector is (\mathbf{q}, \mathbf{p}) , and corresponding worst-case value of \mathbf{b} can be determined as $b_i = \bar{b}_i + \hat{b}_i(r_i^+ - r_i^-)$, $i = 1, \dots, m$. The subgradient of $\mathbf{c}^T \mathbf{x} + \mathbf{f}^T \mathbf{u} + \max_{\mathbf{b} \in \mathcal{B}} Q(\mathbf{x}, \mathbf{u}, \mathbf{b})$ can now be computed as in Lemma 3.1, and is equal to

$$\begin{bmatrix} \mathbf{c} - \mathbf{q} + \mathbf{h} \\ \mathbf{f} + \mathbf{A}^T \mathbf{q} - \mathbf{p} - \mathbf{A}^T \mathbf{h} + \mathbf{k} \end{bmatrix}.$$

7.2 Computational Results

We considered an example of the production planning problem with $m = 2$ raw materials and $n = 30$ finished products with the following data:

- Purchasing and inventory costs of raw materials: $c_j = 100$, $d_j = 150$, $h_j = 20$ for $j = 1, 2$;
- Production and inventory costs of products: $f_i = 530$, $g_i = 750$, $k_i = 50$ for $i = 1, \dots, 30$;
- Components of productivity matrix \mathbf{A} were independently drawn from the integer-valued uniform distribution $U[0, 15]$;
- Product demand ranges: $\bar{b}_i = 10$ and $\hat{b}_i = 5$ for $i = 1, \dots, 30$.

In our example, $\mathbf{g} > \mathbf{A}^T \mathbf{h} + \mathbf{k}$, ensuring that any production \mathbf{v} in the second stage is performed solely in order to satisfy demand not filled by first-stage production \mathbf{u} (see (32)).

To initialize the algorithm, one can select arbitrary feasible values of first-stage variables (\mathbf{x}, \mathbf{u}) , compute the value and subgradient of $Q(\mathbf{x}, \mathbf{u})$, and use the resulting linear function as $Q_0(\mathbf{x}, \mathbf{u})$ (we used conservative values of $\mathbf{u} = \bar{\mathbf{b}} + \hat{\mathbf{b}}$ and $\mathbf{x} = \mathbf{A}\mathbf{u}$).

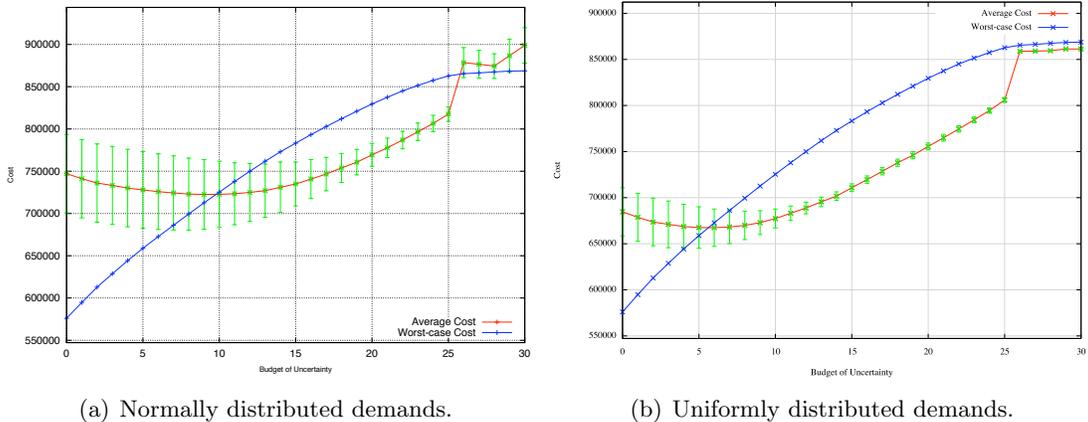


Figure 5: The impact of the budget of uncertainty on worst-case and average cost of the production planning problem, under Normal and Uniform demand distributions.

In this experiment, we considered the situation in which the decision maker assumes the demands for individual products to be random, and has information about the average demands for the various products, as well as some sense of the variabilities of these demands, but does not have a

specific probability model he trusts. Given his information, he fixes the uncertainty set by selecting the values of \bar{b}_i and \hat{b}_i specified above, and considers a range of values of the uncertainty budget Γ .

The blue curve in Figure 5(a) (and in Figure 5(b)) shows the worst-case cost of this problem as a function of budget of uncertainty $\Gamma \in [0, 30]$. To assess the expected performance of robust solutions, we considered two possible distributions of demand: one in which demands for individual products follow independent Normal distributions with mean $\bar{b}_i = 10$ and standard deviation $\hat{b}_i = 5$ (truncated at 0, to avoid negative values), and one in which demands follow independent continuous Uniform distributions on the interval $[\bar{b}_i - \hat{b}_i, \bar{b}_i + \hat{b}_i] = [5, 15]$. (Recall that the decision-maker does not have the full knowledge of the distribution when selecting the uncertainty set.) We generated independent samples of 5000 realizations of the demands for each distribution and plotted the resulting average cost in Figures 5(a) and 5(b) for the Normal and Uniform distributions, respectively. Just as in the newsvendor problem of the previous section, the average cost first decreases with Γ , as incorporating uncertainty into the model yields more robust solutions, reaches its minimum, and then increases with Γ as the solutions become overly conservative. The minimum average cost in Figure 5(a) occurs at $\Gamma = 9$, and by implementing the corresponding ordering and production planning solution, the decision-maker can achieve savings of 3.3% over the solution obtained for $\Gamma = 0$ (i.e., the solution targeted to satisfy the demand $\bar{\mathbf{b}}$); in Figure 5(b) the minimum occurs at $\Gamma = 6$, with savings of 2.5%. These outcomes make intuitive sense, since demands in the sample generated from the Normal distribution exhibit higher variability than in the sample generated from the Uniform distribution. This example illustrates the advantage of the robust optimization approach in situations when precise estimates of probability distributions of uncertain parameters are unavailable, or turn out to be inaccurate. Observe that implementing any of the robust solutions obtained by setting Γ anywhere in the range between 5 and 10 yields a robust solution that would perform well (as measured by the expected cost) regardless of whether the demands follow a Normal or Uniform distribution. We performed a number of additional experiments with demands sampled from a variety of distributions. Results presented in this section are typical of all these experiments, with the minimum average cost occurring at $\Gamma \in [5, 15]$, which indicates that in the production planning problem a fair amount of uncertainty needs to be considered to obtain solutions that perform well in expectation.

It is informative to consider the amounts of raw materials purchased and production done in the first and second stages. The first-stage purchasing and production decisions (\mathbf{x} and \mathbf{u} , respectively) are made according to the solution calculated by solving the robust problem (31). Once the first-stage decisions are made and implemented, actual demand is revealed and the second-stage decisions (\mathbf{y} and \mathbf{v}) tune themselves to the realized demands.

Figures 6(a) and 6(b) plot the total amount of raw materials purchased and products produced, respectively, in the first and second stage as fractions of the total purchasing/production that occurs under the worst-case demand outcome. For $\Gamma = 0$, we start with first-stage purchasing/production levels targeted to satisfy nominal demand. As Γ increases, we split purchasing and production between the two stages. However, for higher values of Γ worst-case demand realizations tend to have values higher than nominal, and thus we see the second-stage purchasing and production decreasing.

Figure 7(a) displays the sample averages of the fractions of total amounts of raw materials purchased in the first and second stage when first-stage decisions are obtained by solving the robust formulation for various values of Γ , and the demands are normally distributed. (Figure 7(b) captures similar

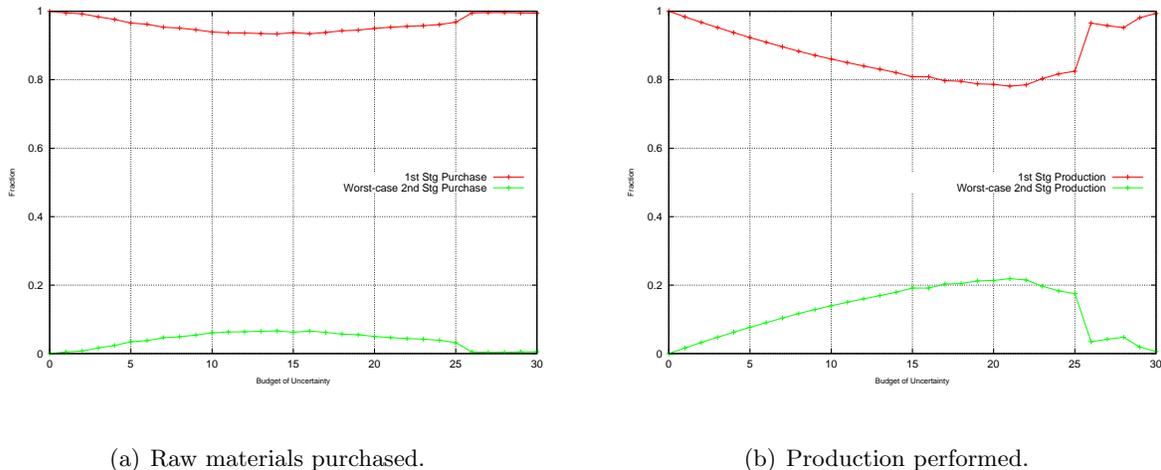
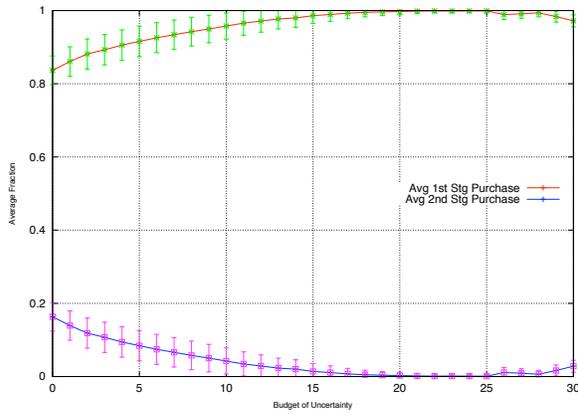


Figure 6: First- and second-stage purchasing and production, as fractions of the total purchasing and production under worst-case demand outcome.

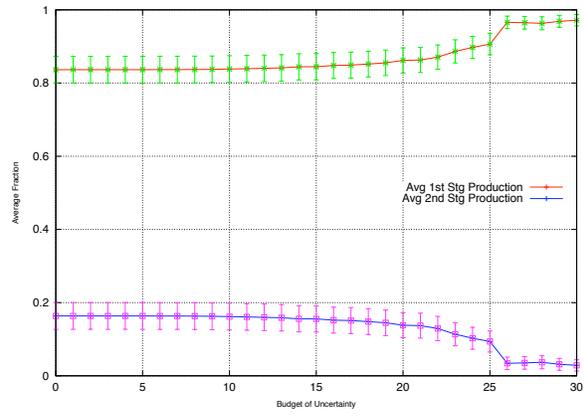
information for total production, and Figures 7(c) and 7(d) display these fractions for uniformly distributed demands.) The errors bars show the standard deviation of these average fractions. With uniformly distributed demands, first-stage decisions obtained using high values of Γ in the description of the uncertainty set actually satisfy the realized demand in most cases, as the average second-stage purchasing is zero, and second-stage production is nearly zero, as 7(c) and 7(d) show (recall also that for these high values of Γ , the average cost of these solutions is almost equal to their worst-case costs). With normally distributed demands, average fraction of material purchases done in the second stage is nearly zero for $\Gamma \in [20, 25]$, but increases for solutions obtained with higher values of Γ ; average fraction of second-stage production decreases with Γ , but never reaches zero. Again, this behavior can partially be explained by higher variability of demands under Normal distribution.

Figures 8(a) and 8(b) summarize our computational experience by illustrating the effect of budget of uncertainty on the number of iterations and the running time of Algorithm 3.2 on this problem. The number of iterations required is not particularly sensitive to the value of Γ (except for very small and very large values of Γ), and therefore the determining factor in the running time are the computational demands of solving the integer program (37) (the adversarial problem), which generally increase with Γ . Therefore, the overall running time of the algorithm generally increases with Γ up to $\Gamma = 23$ and then drops off sharply. Thus, the algorithm has fairly low computational demands for budgets of uncertainty of $\Gamma = 15$ and lower, which were most appropriate for determining robust solutions with low average costs in this and other experiments.

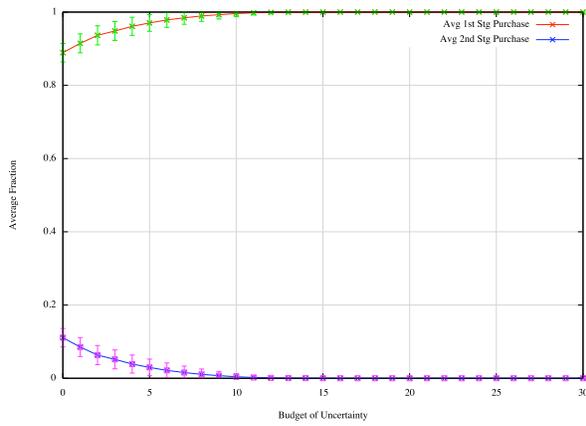
It should be pointed out that, in addition to the value of Γ , the computational demands of the adversarial problem were greatly influenced by the density of the productivity matrix \mathbf{A} . Indeed, if the productivity matrix is dense (as it is in the example presented here), all the raw materials would contribute roughly equally towards the production of most or all of the products, which makes it harder to determine which products (and implicitly which raw materials) are more sensitive to demand fluctuations. In examples where a sparse productivity matrix with pronounced block structure allowed the decisions to be implicitly decomposed by material, instances of the MIP (37)



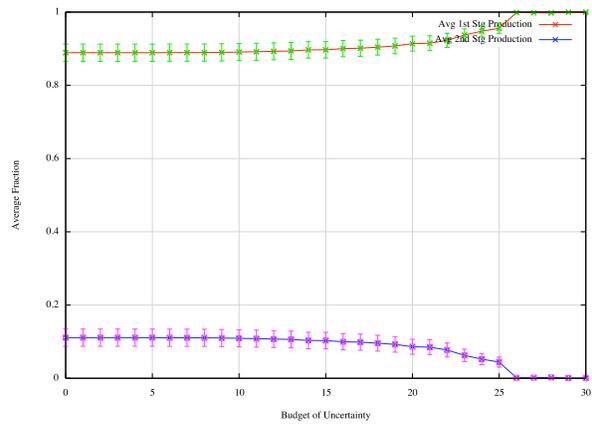
(a) Raw material purchased, normally distributed demands.



(b) Production performed, normally distributed demands.

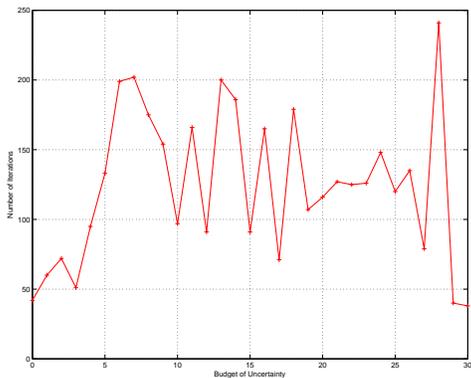


(c) Raw material purchased, uniformly distributed demands.

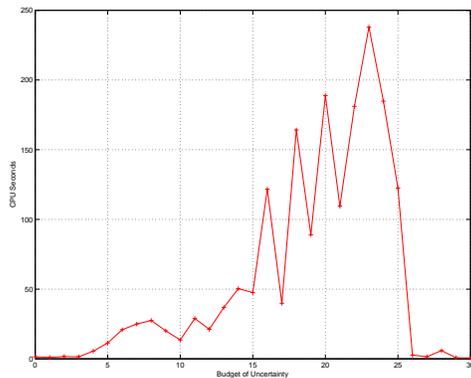


(d) Production performed, uniformly distributed demands.

Figure 7: Sample averages of first- and second-stage purchasing and production, as fractions of the total purchasing and production, under Normal and Uniform demand distributions.



(a) Iteration count vs. Γ .



(b) Run time (CPU seconds) vs. Γ .

Figure 8: The impact of the budget of uncertainty Γ on the number of iterations and run time (CPU seconds) for production planning problem.

were easier and required shorter solution times.

Finally, we would like to remark that, depending on relative magnitudes of problem parameters (e.g., the relative magnitudes of inventory and first- and second-stage production costs), one can devise heuristics that would generate an approximate solution to the adversarial problem (34) (i.e., a “bad,” if not worst, demand realization) quite easily — almost as easily as solving the adversarial problem to optimality in the case of simple recourse. If the demand realization found is worse, in the adversarial sense, than realizations already considered, it can be used to produce a weak cut that separates the current iterate from the epigraph of $\mathcal{Q}(\mathbf{x}, \mathbf{u})$, but is typically not a supporting hyperplane (thus, increasing the number of iterations, but possibly leading to faster overall solution times). Alternatively, an algorithm solving (37) can be terminated prematurely once a high-quality incumbent solution has been found. In fact, in our experiments we observed that the optimal solution of (37) was discovered relatively quickly, and, as is often the case, the bulk of the solution time was spent improving dual bounds and proving optimality of the incumbent solution.

8 Conclusions

We have proposed an approach to linear optimization with recourse that is robust with respect to the underlying probabilities. Specifically, instead of relying on the actual distribution, which would be difficult to estimate accurately, or a family of distributions, which would significantly increase the complexity of the problem at hand, we have modeled random variables as uncertain parameters in a polyhedral uncertainty set and analyzed the problem for the worst-case instance within that set. We have shown that this robust formulation can be solved using a cutting-plane algorithm and standard linear optimization software. We tested our approach on a multi-item newsvendor problem and a production planning problem with demand uncertainties, with encouraging computational results. Analysis of obtained solutions provides insight into appropriate levels of conservatism in planning (as captured by the budget of uncertainty) to obtain lower average costs.

References

- [1] Alper Atamtürk and Muhong Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.
- [2] A. Ben-Tal, A. Goryashko, A. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Math. Program.*, 99(2):351–376, 2004.
- [3] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.
- [4] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, August 1999.
- [5] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Math. Program.*, Ser. A 88:411–424, 2000.
- [6] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, 4:238–252, 1962.
- [7] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Math. Program.*, 98(1-3, Ser. B):49–71, 2003. Integer programming (Pittsburgh, PA, 2002).
- [8] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Oper. Res.*, 52(1):35–53, 2004.
- [9] Dimitris Bertsimas and Aurélie Thiele. A robust optimization approach to inventory theory. *Oper. Res.*, 54(1):150–168, 2006.
- [10] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [11] D Bienstock. Histogram models for robust portfolio optimization. 2007.
- [12] D Bienstock and N Özbay. Computing robust basestock levels. *Discrete Optimization*, Jan 2008.
- [13] John R. Birge and François Louveaux. *Introduction to stochastic programming*. Springer Series in Operations Research. Springer-Verlag, New York, 1997.
- [14] John R. Birge and Roger J.-B. Wets. Computing bounds for stochastic programming problems by means of a generalized moment problem. *Math. Oper. Res.*, 12(1):149–162, 1987.
- [15] Michèle Breton and Saeb El Hachem. Algorithms for the solution of stochastic dynamic minimax problems. *Comput. Optim. Appl.*, 4(4):317–345, 1995.
- [16] Wenqing Chen and Melvyn Sim. Goal-driven optimization. *Oper. Res.*, 57(2):342–357, 2009.
- [17] George B. Dantzig. Linear programming under uncertainty. *Management Sci.*, 1:197–206, 1955.
- [18] J. Dupačová. On minimax decision rule in stochastic linear programming. In *Studies on mathematical programming (Papers, Third Conf. Math. Programming, Mátrafüred, 1975)*, volume 1 of *Math. Methods Oper. Res.*, pages 47–60. Akad. Kiadó, Budapest, 1980.

- [19] J Dupačová. The minimax approach to stochastic programming and an illustrative application. *Stochastics*, 20(1):73–88, 1987.
- [20] J. Dupačová. Stochastic programming: minimax approach. In Ch. A Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, volume V, pages 327–330. Kluwer Academic Publishers, 2001.
- [21] Laurent El Ghaoui and Hervé Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.*, 18(4):1035–1064, 1997.
- [22] Laurent El Ghaoui, Francois Oustry, and Hervé Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. on Optimization*, 9(1):33–52, 1998.
- [23] Yu. Ermoliev, A. Gaivoronski, and C. Nedeva. Stochastic optimization problems with incomplete information on distribution functions. *SIAM J. Control Optim.*, 23(5):697–716, 1985.
- [24] R. Jagannathan. Minimax procedure for a class of linear programs under uncertainty. *Operations Res.*, 25(1):173–177, 1977.
- [25] Peter Kall and Stein W. Wallace. *Stochastic programming*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons Ltd., Chichester, 1994.
- [26] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.
- [27] A Mutapcic and S Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. 2008.
- [28] Fernando Ordóñez and Jiamin Zhao. Robust capacity expansion of network flows. *Networks*, 50(2):136–145, 2007.
- [29] Morten Riis and Kim Allan Andersen. Applying the minimax criterion in stochastic recourse programs. *European J. Oper. Res.*, 165(3):569–584, 2005.
- [30] Alexander Shapiro. Monte Carlo sampling methods. In *Stochastic programming*, volume 10 of *Handbooks Oper. Res. Management Sci.*, pages 353–425. Elsevier, Amsterdam, 2003.
- [31] Alexander Shapiro and Shabbir Ahmed. On a class of minimax stochastic programs. *SIAM J. Optim.*, 14(4):1237–1249 (electronic), 2004.
- [32] Alexander Shapiro and Anton Kleywegt. Minimax analysis of stochastic problems. *Optim. Methods Softw.*, 17(3):523–542, 2002. Stochastic programming.
- [33] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.*, 21(5):1154–1157, 1973.
- [34] A Takeda, S Taguchi, and R Tütüncü. Adjustable robust optimization models for a nonlinear two-period system. *Journal of Optimization Theory and Applications*, Jan 2008.
- [35] Samer Takriti and Shabbir Ahmed. Managing short-term electricity contracts under uncertainty: a minimax approach. Technical report, Georgia Institute of Technology, 2002.
- [36] Samer Takriti and Shabbir Ahmed. On robust optimization of two-stage systems. *Math. Program.*, 99(1, Ser. A):109–126, 2004.

- [37] Jitka Žáčková. On minimax solutions of stochastic linear programming problems. *Časopis Pěst. Mat.*, 91:423–430, 1966.