Math. Program., Ser. A 99: 109–126 (2004)

Digital Object Identifier (DOI) 10.1007/s10107-003-0373-y

Samer Takriti · Shabbir Ahmed*

On robust optimization of two-stage systems

Received: April 29, 2002 / Accepted: January 22, 2003
Published online: March 21, 2003 – © Springer-Verlag 2003

Abstract. Robust-optimization models belong to a special class of stochastic programs, where the traditional expected cost minimization objective is replaced by one that explicitly addresses cost variability. This paper explores robust optimization in the context of two-stage planning systems. We show that, under arbitrary measures for variability, the robust optimization approach might lead to suboptimal solutions to the second-stage planning problem. As a result, the variability of the second-stage costs may be underestimated, thereby defeating the intended purpose of the model. We propose sufficient conditions on the variability measure to remedy this problem. Under the proposed conditions, a robust optimization model can be efficiently solved using a variant of the L-shaped decomposition algorithm for traditional stochastic linear programs. We apply the proposed framework to standard stochastic-programming test problems and to an application that arises in auctioning excess electric power.

Key words. stochastic programming – robust optimization – decomposition methods – risk modeling – utility theory

1. Introduction

Applications requiring decision-making under uncertainty are often modeled as two-stage stochastic programs. In these models, the decision variables are partitioned into two sets. The first is that of variables that are decided prior to the realization of the uncertain event. The second is the set of recourse variables which represent the response to the first-stage decision and realized uncertainty. The objective is to minimize the cost of the first-stage decision and the expected cost of optimal second-stage recourse decisions. The classical two-stage stochastic program [4, 9] with fixed recourse is

$$\min_x \left\{ cx + E_P\{Q(x,\tilde{\xi})\} \mid Ax = b,\ x \geq 0 \right\},$$

where $Q(x,\tilde{\xi}) = \min_y\{q(\tilde{\xi})y \mid Dy = h(\tilde{\xi}) - T(\tilde{\xi})x,\ y \geq 0\}$. The vector x is the first-stage decision variable with an associated cost vector c and a feasible set $\{x \mid Ax = b,\ x \geq 0\}$. The uncertain parameters are functions of the random vector $\tilde{\xi}$ belonging to the set Ξ with a probability distribution P. For a given first-stage decision and realization

S. Takriti: Corresponding Author. Mathematical Sciences Department, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598, USA, e-mail: takriti@us.ibm.com, Tel.: (914) 945-1378, Fax: (914) 945-3434

S. Ahmed: School of Industrial & Systems Engineering, Georgia Institute of Technology, 765 Ferst Drive, Atlanta, Georgia 30332, USA, e-mail: sahmed@isye.gatech.edu

Mathematics Subject Classification (1991): 90C15, 90C33, 90B50, 90A09, 90A43

* Supported in part by NSF Grants DMI-0099726 and DMI-0133943

of the uncertain parameters, the cost of second-stage decisions is given by the optimal value of the recourse problem $Q(x, \tilde{\xi})$. Note that both cost vectors, c and $q(\tilde{\xi})$, have a row representation.

If the distribution of the random variable is discrete with a finite support, one can write the two-stage stochastic program in the form

$$\min_x \left\{ cx + \sum_{k=1}^{K} p_k Q_k(x) \mid Ax = b,\ x \geq 0 \right\}, \tag{1}$$

where for each k

$$Q_k(x) = \min_y \left\{ q_k y \mid Dy = h_k - T_k x,\ y \geq 0 \right\}. \tag{2}$$

Formulation (1) assumes K possible realizations – scenarios – for the random variable $\tilde{\xi}$, each with a probability $p_k > 0, k = 1, \ldots, K$. Due to the separability of the recourse problems, $Q_k(x), k = 1, \ldots, K$, model (1) can be reformulated as

$$\begin{aligned} \min_{x, y_1, \ldots, y_K} \Bigg\{ cx + \sum_{k=1}^{K} p_k q_k y_k \mid Ax = b,\ x \geq 0, \\ T_k x + D y_k = h_k,\ y_k \geq 0,\ k = 1 \ldots, K \Bigg\}. \end{aligned} \tag{3}$$

Note that the KKT optimality conditions for (1) imply those for (3) and vice versa. Consequently, given an optimal solution $x^*, y_1^*, \ldots, y_K^*$, for problem (3), the vector y_k^* is optimal for $Q_k(x^*), k = 1, \ldots, K$. The linear program in (3) is called the deterministic equivalent [4, 9].

Formulation (1) generally assumes that the decision maker – modeler – is risk neutral; i.e., an optimal x is chosen solely on its present and expected future cost $cx + \sum_{k=1}^{K} p_k Q_k(x)$. To capture the notion of risk in a two-stage stochastic program, several studies suggest modifying the probability measure $p_k, k = 1, \ldots, K$ [5, 12]. Alternatively, Mulvey, Vanderbei, and Zenios [18] propose capturing risk explicitly using the following model:

$$\begin{aligned} \min_{x, y_1, \ldots, y_K} \Bigg\{ cx + \sum_{k=1}^{K} p_k q_k y_k + \lambda f(q_1 y_1, \ldots, q_K y_K) \mid Ax = b,\ x \geq 0, \\ T_k x + D y_k = h_k,\ y_k \geq 0,\ k = 1, \ldots, K \Bigg\}, \end{aligned} \tag{4}$$

where $f : \mathbb{R}^K \to \mathbb{R}$ is a variability measure, usually the variance, on the second-stage costs. The non-negative scalar λ represents the risk tolerance of the modeler. Depending on the value of λ, the optimization may favor solutions with a higher expected second-stage cost $\sum_{k=1}^{K} p_k q_k y_k$ in exchange for a lower variability in the second-stage costs as measured by $f(q_1 y_1, \ldots, q_K y_K)$. Mulvey et al. [18] refer to their model as robust optimization. This framework has found applications in power-system capacity planning [17], chemical-process planning [1], telecommunications-network design [2, 13], and financial planning [2, 18].

In this paper we show that, for an arbitrary variability measure f, the robust optimization model (4) might lead to second-stage solutions y_k, $k = 1, \dots, K$, that are not optimal to the recourse problem (2). As a result, formulation (4) may underestimate the actual variability of the second-stage costs which, we feel, violates the intent of the model and results in misleading decisions. We propose sufficient conditions on f to remedy this problem. Furthermore, we show that, under the proposed conditions, a robust optimization model can be efficiently solved using a variant of the L-shaped decomposition algorithm for stochastic linear programs.

The remainder of this paper is organized as follows. In the next section, we illustrate the potential difficulty with model (4) through an example, and discuss the optimality conditions of the model to explain this behavior. In Section 3, we propose conditions to ensure that the second-stage solutions of a robust model are optimal with respect to (2) and suggest a decomposition scheme for solving the resulting robust problem. We also discuss extensions of two-stage stochastic programming to a more general dis-utility minimization setting which permits capturing the modeler's risk attitude. Finally, in Section 4, we report on our computational experience with the proposed approach.

2. Issues with robust optimization

2.1. An illustrative example

Consider the following instance of the two-stage stochastic-programming model (1) with two equi-probable scenarios

$$\min_x \left\{ 2x + 0.5Q_1(x) + 0.5Q_2(x) \mid x \geq 0 \right\},$$

where $Q_1(x) = \min_y \{ y \mid y \geq 3 - x,\ y \geq 0 \}$, and $Q_2(x) = \min_y \{ y \mid y \geq 2 - x,\ y \geq 0 \}$. Clearly, the optimal second-stage solutions are $[3-x]_+$ and $[2-x]_+$ for scenarios 1 and 2, respectively, where $[\,\cdot\,]_+ = \max(0,\ \cdot)$. Let us now consider the robust optimization framework (4) with the aim of controlling the variability of the second-stage costs

$$\min_{x, y_1, y_2} \left\{ 2x + 0.5y_1 + 0.5y_2 + \lambda f(y_1, y_2) \mid x + y_1 \geq 3,\ x + y_2 \geq 2, \right.$$
$$\left. x \geq 0,\ y_1 \geq 0,\ y_2 \geq 0 \right\}, \tag{5}$$

where f is the variance of the second-stage costs; i.e., $f(y_1, y_2) = (y_1 - y_2)^2/4$. It is easily verified that for $0 \leq \lambda < 1$, the optimal solution of (5) is $x = 0$, $y_1 = 3$, and $y_2 = 2$. In this case, for a first-stage value of $x = 0$, the solutions y_1 and y_2 are optimal to the recourse problems corresponding to scenarios 1 and 2, respectively. However, as λ exceeds one, the optimal solution of (5) becomes $x = 0$, $y_1 = 3$, and $y_2 = 3 - 1/\lambda$. Note that y_2 is no longer optimal to $\min_y \{ y \mid y \geq 2 - x,\ y \geq 0 \}$, when $x = 0$. While the robust model (4) results in a variance of $1/4\lambda^2$, the actual variance at $x = 0$ is $1/4$.

By varying λ in formulation (5), the decision maker's intent is to control risk – second-stage variance. However, the change in the variance is not a result of changing x but rather a consequence of changing y_2. We believe that it is unlikely that a decision maker would be interested in such a model. As a matter of fact, it is often the case that decision makers do not control the second-stage decisions completely; and that the model is an attempt to approximate their costs. An example is the truss-design problem discussed in [18]. The first stage determines the structural design, while the second-stage decisions represent the forces in the truss elements. For a given load, these forces – second-stage decisions – are completely controlled by laws of physics and cannot be affected by the decision maker. The only way to control them is through the adoption of a suitable design x that minimizes the deviation in the second-stage response. As a result, we feel that an appropriate model is one that enforces robustness while guaranteeing that the second-stage solution is optimal for the recourse problem (2). Additional examples illustrating that robust models may result in second-stage solutions that are suboptimal for the recourse problem are presented in [11] and [23].

2.2. *Optimality conditions*

The above simple example illustrates that the second-stage solutions obtained from the robust optimization model (4) can be suboptimal to the second-stage recourse problem (2) and that the variability may be underestimated. In this section, we analyze the optimality conditions of (4) to understand this behavior.

Throughout the rest of this paper, we assume that the two-stage model (1), and equivalently (3), is feasible and bounded below, and that the robust model (4) is bounded below. We also assume, without loss of generality, that all probabilities, p_k, $k = 1, \ldots, K$, are strictly positive. Furthermore, we assume that the variability measure $f : \mathbb{R}^K \mapsto \mathbb{R}$ is subdifferentiable, and denote the subdifferential of f at $z = (z_1, \ldots, z_K)$ by $\partial f(z)$.

For a given x and λ, let us consider the value function

$$\phi_\lambda(x) = \min_{y_1,\ldots,y_K} \left\{ \sum_{k=1}^{K} p_k q_k y_k + \lambda f(q_1 y_1, \ldots, q_K y_K) \mid D y_k = h_k - T_k x \right.$$
$$\left. y_k \geq 0,\ k = 1, \ldots, K \right\}. \tag{6}$$

Proposition 1. *Let f be subdifferentiable. Given λ and x, let y_k, $k = 1, \ldots, K$, be a feasible solution for (6) and denote $q_k y_k$ by z_k, $k = 1, \ldots, K$. Then, y_k, $k = 1, \ldots, K$, satisfies the Kuhn-Tucker optimality conditions for (6) if and only if there exists a subgradient $g = (g_1, \ldots, g_K) \in \partial f(z_1, \ldots, z_K)$ such that the following two conditions are satisfied:*

(a) if $(p_k + \lambda g_k) > 0$, the vector y_k solves the kth recourse problem

$$Q_k(x) = \min_y \left\{ q_k y \mid D y = h_k - T_k x,\ y \geq 0 \right\};$$

and

(b) if $(p_k + \lambda g_k) < 0$, *the vector* y_k *solves*

$$R_k(x) = \max_y \left\{ q_k y \mid Dy = h_k - T_k x,\ \ y \geq 0 \right\}. \tag{7}$$

Proof. Note that a feasible solution y_k, $k = 1, \ldots, K$, for (6) is feasible for both (2) and (7). First, we assume that y_k, $k = 1, \ldots, K$, satisfies the Kuhn-Tucker optimality conditions for (6) and show that conditions (a) and (b) must hold. Since y_k is optimal, then there exists a subgradient $g \in \partial f(z_1, \ldots, z_K)$ and a row vector π_k, $k = 1, \ldots, K$, such that $\pi_k D \leq (p_k + \lambda g_k) q_k$ and $(p_k q_k + \lambda g_k q_k - \pi_k D) y_k = 0, k = 1, \ldots, K$, where g has row representation. For claim (a), the row vector $\mu_k = \pi_k / (p_k + \lambda g_k)$ is dual feasible and satisfies complementarity slackness $(q_k - \mu_k D) y_k = 0$ for (2). Therefore, y_k is optimal for (2), which proves claim (a). Claim (b) can be shown in a similar fashion through duality. Now, we show the sufficiency of the condition; i.e., the presence of a suitable g that satisfies both (a) and (b) guarantees that y_k, $k = 1, \ldots, K$, is optimal for (6). Let μ_k be the dual multiplier associated with y_k in an optimal solution for (2) if $p_k + \lambda g_k > 0$, or for (7) if $p_k + \lambda g_k < 0$. Now, define π_k to be $(p_k + \lambda g_k)\mu_k$ if $p_k + \lambda g_k \neq 0$ and to be zero otherwise. Then, the pair (y, π) satisfies the Kuhn-Tucker optimality conditions for (6). □

Corollary 1. *Let* f *be subdifferentiable. Given* λ *and a solution* x^*, *let* $y_1^*, \ldots, y_K^*$, *be optimal for (6); let* $z_k^* = q_k y_k^*$, $k = 1, \ldots, K$; *and let* $g^* \in \partial f(z_1^*, \ldots, z_K^*)$. *Let* $\Gamma_Q = \{k \mid p_k + \lambda g_k^* > 0,\ k = 1, \ldots, K\}$, *and* $\Gamma_R = \{k \mid p_k + \lambda g_k^* < 0,\ k = 1, \ldots, K\}$. *Then the row vector* $-\sum_{k=1}^K (p_k + \lambda g_k^*) \mu_k^* T_k$ *is a subgradient of* ϕ_λ *at* x^*, *where* μ_k^* *is an optimal dual solution for (2) if* $k \in \Gamma_Q$, *an optimal dual solution for (7) if* $k \in \Gamma_R$, *and the zero vector otherwise.*

Proposition 1 explains why the robust model's second-stage solution may be suboptimal for the recourse problem. Depending on the nature of f and the value of λ, the robust model may yield a second-stage solution y_k that is not optimal for the kth recourse problem (2). Referring back to the example in Section 2.1, for $0 \leq \lambda < 1$, the optimal solution is $x = 0$, $y_1 = 3$, and $y_2 = 2$, with $p_1 + \lambda \nabla f_1 = (1 + \lambda)/2 > 0$ and $p_2 + \lambda \nabla f_2 = (-1 + \lambda)/2 > 0$. Accordingly, each of y_1 and y_2 solves its corresponding recourse problem (2). However, when λ exceeds one, the optimal solution is $x = 0$, $y_1 = 3$, and $y_2 = 3 - 1/\lambda$, with $p_1 + \lambda \nabla f_1 = 1$ and $p_2 + \lambda \nabla f_2 = 0$. Hence, the solution y_2 is no longer guaranteed to be optimal for (2).

Note that for a given x, the actual variability of the optimal second-stage costs is $f(Q_1(x), \ldots, Q_K(x))$. The following proposition shows that the robust model (4) underestimates the actual second-stage variability.

Proposition 2. *Given* $\lambda \geq 0$, *let* x^*, $y_1^*, \ldots, y_K^*$, *be an optimal solution for (4). Then,*

$$f(Q_1(x^*), \ldots, Q_K(x^*)) \geq f(q_1 y_1^*, \ldots, q_K y_K^*),$$

where $Q_k(x^*)$ *is as defined in (2).*

Proof. Let y_k^Q be optimal for the recourse problems; i.e., $Q_k(x^*) = q_k y_k^Q$, $k = 1, \ldots, K$. As $x^*, y_1^Q, \ldots, y_K^Q$, is a feasible solution for (4), we have

$$\begin{aligned} cx^* &+ \sum_{k=1}^{K} p_k q_k y_k^* + \lambda f(q_1 y_1^*, \ldots, q_K y_K^*) \\ &\leq cx^* + \sum_{k=1}^{K} p_k q_k y_k^Q + \lambda f(q_1 y_1^Q, \ldots, q_K y_K^Q). \end{aligned}$$

The result follows from noting that $\lambda \geq 0$ and that y_k^* is feasible for the recourse problem, hence $q_k y_k^* \geq q_k y_k^Q$, $k = 1, \ldots, K$. □

3. Robust optimization under an appropriate variability measure

The previous section highlights a crucial deficiency of the robust optimization framework under arbitrary variability measures. In this section, we suggest sufficient conditions on the variability measure f to ensure that the robust model (4) produces second-stage solutions that are optimal for (2). Under the proposed conditions, a modified version of the L-shaped decomposition algorithm solves (4) efficiently. Finally, we discuss the issue of modeling risk using a dis-utility minimization setting.

3.1. Conditions guaranteeing second-stage optimality

Consider the following robust model

$$\min_x \left\{ cx + \sum_{k=1}^{K} p_k Q_k(x) + \lambda f\,(Q_1(x), \ldots, Q_K(x)) \mid Ax = b,\ x \geq 0 \right\}. \tag{8}$$

The above model dispenses with the issue of second-stage suboptimality, since the optimal second-stage costs are explicitly considered. The discussion in Section 2.2 suggests that under an arbitrary variability measure f, models (4) and (8) are not equivalent.

We now provide a sufficient condition on f to guarantee the equivalence of models (4) and (8). Given two vectors z^1 and z^2 in $\mathbb{R}^K$, the notation $z^1 < z^2$ means $z_k^1 \leq z_k^2$ for $k = 1, \ldots, K$, and $z_k^1 < z_k^2$ for some k. A function $f : \mathbb{R}^K \mapsto \mathbb{R}$ is called *non-decreasing* if $z^1 < z^2$ implies $f(z^1) \leq f(z^2)$; and is called *increasing* if $z^1 < z^2$ implies $f(z^1) < f(z^2)$.

Proposition 3. *Let f be a non-decreasing function and $\lambda \geq 0$. Then, models (4) and (8) are equivalent.*

Proof. Let x^Q be an optimal solution for (8) with the vectors $y_1^Q, \ldots, y_K^Q$, being optimal for the recourse problems; i.e., $Q_k(x^Q) = q_k y_k^Q$, $k = 1, \ldots, K$. Note that $x^Q, y_1^Q, \ldots, y_K^Q$, is feasible for (4); i.e., the optimal value of (4) is a lower bound on that of (8). Now, let $x^*, y_1^*, \ldots, y_K^*$, be an optimal solution for (4). Furthermore, assume that there exists a k such that y_k^* is not optimal for the kth recourse problem; i.e., $q_k y_k^* > q_k \overline{y}_k$ where $\overline{y}_k$ solves $\min_y\{q_k y \mid Dy = h_k - T_k x^*\}$. Then, replacing y_k^* with $\overline{y}_k$ maintains feasibility. In addition, and due to the assumptions that f is non-decreasing, $\lambda \geq 0$, and $p_k > 0$, the objective function value of (4) improves, which contradicts the assumption that the solution $x^*, y_1^*, \ldots, y_K^*$, is optimal. Therefore, the vector y_k^*, $k = 1, \ldots, K$, solves the kth recourse problem, resulting in the value of (4) being an upper bound on the optimal value of (8). Hence, program (4) and program (8) are equivalent. □

The above result suggests that using a non-decreasing variability measure resolves the issue of second-stage suboptimality in (4). Under subdifferentiability of f, this condition is also evident from Proposition 1, since a non-decreasing subdifferentiable function has a non-negative subgradient. The use of non-decreasing variability measures is not new to the modeling community. Such functions have been widely used in the finance literature. We refer the reader to [3], [6], [10], and [20] for examples of such functions.

Next, we consider an alternative robust optimization model where second-stage cost variability is minimized subject to an upper bound on the expected cost.

Proposition 4. *Let f be an increasing function and $\rho \in \mathbb{R}$. Then,*

$$\min_x \left\{ f\left(Q_1(x), \ldots, Q_K(x)\right) \mid Ax = b,\ x \geq 0,\ cx + \sum_{k=1}^{K} p_k Q_k(x) \leq \rho \right\}, \tag{9}$$

is equivalent to

$$\min_{x, y_1, \ldots, y_K} \left\{ f(q_1 y_1, \ldots, q_K y_K) \mid Ax = b,\ x \geq 0,\ cx + \sum_{k=1}^{K} p_k q_k y_k \leq \rho, \right.$$
$$\left. T_k x + D y_k = h_k,\ y_k \geq 0,\ k = 1, \ldots, K \right\}. \tag{10}$$

Proof. Observe that either both (9) and (10) are feasible or both are infeasible. We assume that both are feasible under the given ρ. As in the proof of Proposition 3, one can show that the optimal objective value of (10) is a lower bound on that of (9). Now, let x^*, $y_1^*, \ldots, y_K^*$, be an optimal solution for (10). Furthermore, assume that there exists a k such that y_k^* is not optimal for the kth second-stage problem; i.e., $q_k y_k^* > q_k \overline{y}_k$ where $\overline{y}_k$ solves $\min_y\{q_k y \mid Dy = h_k - T_k x^*\}$. Then, replacing y_k^* with $\overline{y}_k$ maintains feasibility. Furthermore, and due to the assumption that f is increasing, the objective function value improves, which contradicts the assumption that the solution $x^*, y_1^*, \ldots, y_K^*$, is optimal for (10). Therefore, the vector y_k^*, $k = 1, \ldots, K$, solves the kth second-stage problem, resulting in the value of (9) being a lower bound on the optimal value of (10). Hence, program (9) and program (10) are equivalent. □

Note that a non-decreasing f in (4) and an increasing f in (10) are sufficient to guarantee that the resulting second-stage solutions are optimal for the recourse problem (2). However, for computational tractability, one needs to assume that f is convex.

The convex and non-decreasing nature of f, in conjunction with the convexity of Q_k guarantee that the function $f(Q_1(\cdot), \ldots, Q_K(\cdot))$ is convex in x (see, for example, Theorem 5.1 of [22]). Furthermore, also from convexity, problem (10) can be reformulated in the form of (4) by relaxing the constraint $cx + \sum_{k=1}^{K} p_k q_k y_k \leq \rho$ with an appropriate Lagrange multiplier. Hence, formulations (4) and (10) are equivalent.

Corollary 2. *Let f be convex and increasing. Then, formulations (4), (8), (9), and (10) are equivalent.*

3.2. A modified L-shaped decomposition algorithm

In this section, we suggest a decomposition algorithm for (4) for a convex and non-decreasing variability measure f.

In the space of the first-stage variables, model (4) can be stated as

$$\min_x \left\{ cx + \phi_\lambda(x) \mid Ax = b,\ x \geq 0 \right\}, \tag{11}$$

where $\phi_\lambda(x)$ is defined as in (6). Problem (11) is a convex non-linear program, and can be solved using a cutting-plane algorithm, such as generalized Benders' decomposition [7]. Recall that, in each iteration, this class of algorithms solves a linear relaxation of the problem, wherein the convex function $\phi_\lambda(x)$ is under-approximated by a collection of supporting hyperplanes using subgradient information. As the iterations progress, the approximation of $\phi_\lambda(x)$ improves and the solutions to the relaxation converge to a solution of the true problem.

In general, evaluating and computing a subgradient of $\phi_\lambda(x)$ would require the solution of the large-scale non-linear program (6). However, under the non-decreasing condition on f suggested by Proposition 3, the function $\phi_\lambda(x)$ can be evaluated by (independently) solving the linear recourse programs $Q_k(x)$, $k = 1, \ldots, K$, and assigning

$$\phi_\lambda(x) = \sum_{k=1}^{K} p_k Q_k(x) + \lambda f(Q_1(x), \ldots, Q_K(x)).$$

Furthermore, from Corollary 1, a subgradient of $\phi_\lambda(x)$ can be calculated using a subgradient of f and the optimal dual multipliers of the recourse problems $Q_k(x), k = 1, \ldots, K$. Thus, given a first-stage solution x, a support of $\phi_\lambda(x)$ can be computed in a decomposed fashion by independently solving the linear recourse problems $Q_k(x)$, $k = 1, \ldots, K$. A cutting-plane algorithm for (11) can then be stated as follows.

Initialization. Let $\lambda \geq 0$ be given. Set the iteration counter i to 1. Set $UB \leftarrow \infty$ and $LB \leftarrow -\infty$. Let $\epsilon > 0$ be the error tolerance in an optimal solution. Let $\mathcal{I}_o \leftarrow \emptyset$ and $\mathcal{I}_f \leftarrow \emptyset$ be the sets of indices corresponding to the optimality and feasibility cuts, respectively.

General Step.

1. Set the value of the LB to

$$\min_{x,w}\{cx + w \mid Ax = b,\ x \geq 0,\ w \geq \phi_\lambda(x^\iota) - \sum_{k=1}^{K} \pi_k^\iota T_k(x - x^\iota),\ \iota \in \mathcal{I}_o,$$
$$\eta_k^\iota(h_k - T_k x) \leq 0,\ k \in \mathcal{K}_\iota,\ \iota \in \mathcal{I}_f\},$$

and let x^i be its optimal solution. If the problem is unbounded, choose any feasible x as a solution and set $LB \leftarrow -\infty$.

2. Solve the second-stage recourse problems $\min_y\{q_k y \mid Dy = h_k - T_k x^i,\ y \geq 0\}$, $k = 1, \ldots, K$. If all recourse problems are feasible,

   Let z_k be the optimal objective value and μ_k be optimal dual multipliers for the kth recourse problem. Calculate $UB = cx + \sum_{k=1}^{K} p_k z_k + \lambda f(z_1, \ldots, z_K)$. If $UB - LB < \epsilon$, terminate. The current solution x^i, $y_1^i, \ldots, y_K^i$ is ϵ-optimal for (4). Otherwise, let $\pi_k^i = (p_k + \lambda g_k)\mu_k^i$, where $(g_1, \ldots, g_K) \in \partial f(z_1, \ldots, z_K)$. Let $\mathcal{I}_o \leftarrow \mathcal{I}_o \cup \{i\}$. Set $i \leftarrow i + 1$. Go to Step 1.

3. If any of the second-stage recourse problems is infeasible with respect to the current x^i,

   Let $\mathcal{K}_i$ be the set of infeasible constraints and η_k^i, $k \in \mathcal{K}_i$, be the extreme dual rays associated with these constraints. Set $i \leftarrow i + 1$. Go to Step 1.

Apart from the scenario-wise decomposition in the evaluation of $\phi_\lambda(x)$ and its subgradient, the essential elements of the above algorithm are identical to standard cutting-plane methods for convex programs, and convergence follows from existing results (see, for example, [7]). In particular, the above method is a modification of the L-shaped decomposition algorithm [25] for stochastic linear programs, wherein the optimality-cut coefficients corresponding to the dual solutions of the kth recourse problems are scaled by the factor $(p_k + \lambda g_k)$, with g_k being the kth component of a subgradient of f. We refer the reader to [4] for a detailed discussion on the L-shaped decomposition algorithm for stochastic linear programs.

3.3. Extensions to a dis-utility minimization setting

In this section, we discuss risk modeling in two-stage systems using a more general dis-utility minimization framework. In this setting, risk may be incorporated into the two-stage stochastic program using a dis-utility function $U : \mathbb{R} \to \mathbb{R}$ that captures the modeler's risk tolerance. For a given U, two random cost variables can be ranked using the expected value of the dis-utility associated with each of them [8]. In (1), the cost associated with a decision x is a random variable with K realizations, $cx + Q_k(x)$, $k = 1, \ldots, K$. Therefore, a feasible solution x^* is preferred over another solution x if $\sum_{k=1}^{K} p_k U(cx^* + Q_k(x^*)) < \sum_{k=1}^{K} p_k U(cx + Q_k(x))$. Then, an optimal decision can be found by minimizing the expected dis-utility over the feasible domain of the first-stage variable x

$$\min_x \left\{ \sum_{k=1}^{K} p_k U(cx + Q_k(x)) \mid Ax = b,\ x \geq 0 \right\}, \tag{12}$$

where $U(.)$ is non-decreasing; i.e., lower costs are preferred over higher costs, and $Q_k(x)$, $k = 1, \dots, K$, is as defined in (2). The choice of U determines the modeler's attitude towards risk. For example, if U is a convex function, then the model is risk averse, while if it is linear, the model is risk neutral.

Let $\theta(x) = \sum_{k=1}^{K} p_k U(cx + Q_k(x))$. For a convex and non-decreasing dis-utility function U, the objective function $\theta(x)$ of problem (12) is a convex function of x. Consequently, standard cutting-plane algorithms for convex programs are applicable.

Proposition 5. *Let $U : \mathbb{R} \to \mathbb{R}$ be convex and increasing. Given a solution x^*, let μ_k^* be an optimal dual for the kth recourse problem $Q_k(x^*)$, $k = 1, \dots, K$. Furthermore, let $w_k^* = cx^* + Q_k(x^*)$, $k = 1, \dots, K$, and $v_k^* \in \partial U(w_k^*)$, $k = 1, \dots, K$. Then the row vector $\sum_{k=1}^{K} p_k v_k^*(c - \mu_k^* T_k)$ is a subgradient of θ at x^*.*

Proof. Due to the convexity of $Q_k(x)$ and to the fact that $-\mu_k^* T_k$ is a subgradient of Q_k at x^*, we can write $cx + Q_k(x) \geq cx^* + Q_k(x^*) + (c - \mu_k^* T_k)(x - x^*)$, $k = 1, \dots, K$. Now, one can apply the increasing function U to both sides of the previous inequality, resulting in

$$\begin{aligned} U(cx + Q_k(x)) &\geq U(cx^* + Q_k(x^*) + (c - \mu_k^* T_k)(x - x^*)) \\ &\geq U(cx^* + Q_k(x^*)) + v_k^*(c - \mu_k^* T_k)(x - x^*). \end{aligned}$$

The last inequality follows from the convexity of U. The statement of the proposition follows from multiplying both sides by $p_k > 0$ and summing over all scenarios $k = 1, \dots, K$. □

Note that for a given x, evaluating $\theta(x)$ can be accomplished by independently solving the recourse problems and calculating $\sum_{k=1}^{K} p_k U(cx + Q_k(x))$. A subgradient can be evaluated, as suggested by Proposition 5, using the dual multipliers of the recourse problems. Consequently, the modified L-shaped decomposition algorithm of Section 3.2 is directly applicable to the more general dis-utility minimization problem (12).

4. Computational results

In this section, we report on our numerical experience with the proposed robust optimization framework on standard test problems and on an application arising in the electric-power industry. The purpose of our numerical experiments is two-fold. The first is to illustrate the applicability of robust optimization to control variability in stochastic programming applications. The second is to demonstrate the computational advantage of the proposed linear-programming-based decomposition algorithm over direct non-linear programming approaches for solving robust optimization models.

4.1. Application to standard test problems

Our first set of experiments concerns with two standard two-stage stochastic-linear programs from the literature, namely STORM and 20TERM. Briefly, STORM is a problem of routing cargo-carrying flights over a set of routes in a network while meeting the

Table 1. The number of constraints and variables in the first and second stage of the two test problems

STORM				20TERM			
1st Stage		2nd Stage		1st Stage		2nd Stage	
Const.	Vars.	Const.	Vars.	Const.	Vars.	Const.	Vars.
185	121	528	1259	3	63	124	764

uncertain demand [19]; and 20TERM is a fleet routing problem in which the second-stage demand is uncertain [16, 21]. Data for these problems were obtained from the web site of [14]. Table 1 provides the number of rows and columns in a deterministic instance for each of the two problems. The stochastic parameters in these test problems have independent discrete distributions resulting in a total of 6×10^{81} scenarios for STORM and 1.1×10^{12} scenarios for 20TERM. We generated test instances with up to 100 scenarios using Monte Carlo sampling. A 100-scenario instance of STORM has 52,985 constraints and 126,021 variables and that of 20TERM has 12,403 constraints and 76,463 variables.

To test the robust optimization framework, we extended STORM and 20TERM by appending a measure of second-stage-cost variability in the objective. We use the following variability measure for the second-stage costs in our computations

$$f(Q_1(x), \ldots, Q_K(x)) = \sum_{k=1}^{K} p_k [Q_k(x) - R^*]_+^2, \tag{13}$$

where R^* is the target second-stage cost. Note that the variability measure is convex and non-decreasing. Various properties of this measure have been studied in the context of financial applications [6]. We compared three approaches for solving the robust extension of STORM and 20TERM: direct solution of model (4) as a large-scale quadratic program; a generalized Benders' decomposition approach for (11) where $\phi_\lambda(x)$ is computed directly by solving the quadratic program (6); and the linear-programming-based decomposition algorithm of Section 3.2. We refer to these approaches by (Q), (QD), and (LD), respectively. We use the CPLEX 7.5 barrier solver for approach (Q). Approaches (QD) and (LD) were implemented using C++ with CPLEX's dual simplex solver for the linear subproblems, and CPLEX's barrier solver for the quadratic subproblems. All calculations were performed on a Sun Sparc Workstation 450 MHz running Solaris 5.7.

We begin by comparing the effect of the problem size in terms of the number of scenarios K on each of the solution approaches. Figure 1 compares the execution times of the three solution approaches for STORM when the number of scenarios K is varied from 5 to 100 for three λ-R^* combinations. These combinations are chosen such that the penalty on cost variability is increasing from chart (a) to chart (c) in Figure 1. The significant CPU advantage of approach (LD) over the other two approaches as the number of scenarios increases is evident. It can be observed that increasing the penalty on variability increases the required computational effort for all three approaches. We observed similar behavior with 20TERM.

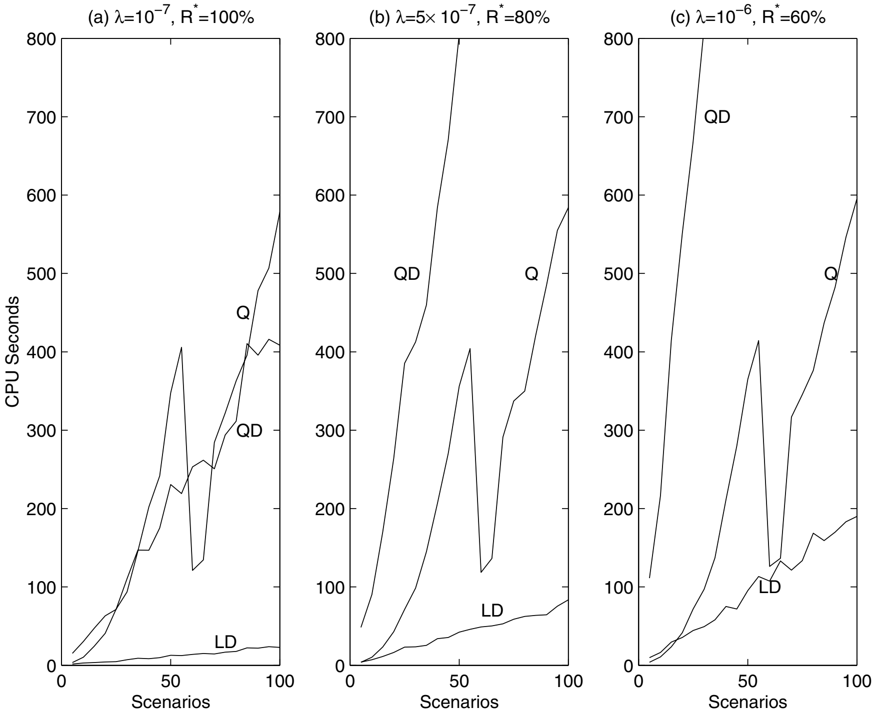

Fig. 1. Comparison of execution times for STORM when number of scenarios K is changed between 5 and 100

Next, we study the applicability of the proposed framework for controlling the variability of the second-stage costs with respect to a target cost R^*. We use the proposed linear-programming-based decomposition approach (LD) for solving the robust model for various λ-R^* combinations. Table 2 and Table 3 present the results for STORM and 20TERM when $K = 100$. The first column of each table provides the target cost R^* as a percentage of the optimal second-stage cost $\sum_{k=1}^{K} p_k Q_k(x^*)$ when $\lambda = 0$ (the standard stochastic program). For example, the expected second-stage cost for STORM without any penalty on variability is 9679627. When Table 2 provides the value of 80% in column "R^*," the target value is $9679627 \times 80\% = 7743701$. The second column provides the value of λ in multiples of 10^{-7} in case of Table 2, and in multiples of 10^{-5} in case of Table 3. Column "Cuts" provides the number of cuts added in the (LD) algorithm. Note that the number of cuts depends on the accuracy required in an optimal solution. Our code stops when the relative difference between the upper and lower bounds is below 10^{-6}. Columns "cx^*," "$EQ(x^*)$," "Cost," and "$f(\cdot)$" provide the values of the first-stage cost, the expected value of the second-stage cost, the sum of first and expected second-stage costs, and the variability as computed using (13), respectively. Finally, the last column provides the CPU seconds required by (LD). For 20TERM, both "Cost" and "$f(.)$" are depicted graphically in Figure 2.

As expected, for a given value of R^*, increasing λ reduces the variability at the expense of increasing the total cost. For a given λ, decreasing the target second-stage cost R^* forces the second-stage costs to be smaller at the expense of higher first-stage

Table 2. Results for STORM with 100 scenarios. The values of λ are in multiples of 10^{-7} and those of $f(.)$ are in multiples of 10^7

R^*	λ	Cuts	cx^*	$EQ(x^*)$	Cost	$f(.)$	CPU
–	0	50	5818723	9679627	15498350	–	26.70
100%	1	44	5853398	9645328	15498726	4990	23.40
	5	57	5899601	9601937	15501537	4104	29.46
	10	52	5941350	9564835	15506186	3454	25.21
90%	1	47	5942561	9563800	15506361	82936	24.39
	5	121	6487955	9159346	15647301	29997	59.99
	10	120	6880558	8890016	15770574	11749	59.69
80%	1	89	6123922	9417984	15541905	290568	44.54
	5	162	7823787	8251453	16075240	35771	83.61
	10	150	8257502	7967127	16224629	13913	77.42
70%	1	115	6626946	9063971	15690917	533799	59.24
	5	146	9249246	7319257	16568503	39551	77.91
	10	191	9715286	7020317	16735603	15029	109.45
60%	1	134	7775395	8283304	16058699	622996	67.56
	5	219	10599947	6472183	17072130	53542	137.30
	10	291	11140819	6155373	17296192	20731	188.72

Table 3. Results for 20TERM with 100 scenarios. The values of λ are in multiples of 10^{-5} and those of $f(.)$ is in multiples of 10^5

R^*	λ	Cuts	cx^*	$EQ(x^*)$	Cost	$f(.)$	CPU
–	0	314	53602	200981	254583	–	131.19
100%	1	302	53807	200800	254607	597	123.73
	5	421	54900	200068	254968	406	184.90
	10	484	55513	199865	255378	356	217.65
90%	1	360	54501	200293	254793	4683	164.62
	5	530	55500	199844	255344	4399	233.49
	10	815	56499	199708	256208	4284	364.23
80%	1	374	55050	199986	255036	16209	165.90
	5	485	55800	199787	255587	15991	224.85
	10	694	57300	199607	256907	15794	299.30
70%	1	442	55200	199914	255113	35912	202.16
	5	1110	56696	199672	256368	35536	516.78
	10	1276	57500	199588	257088	35417	643.18
60%	1	366	55200	199913	255113	63758	163.30
	5	981	57188	199615	256803	63182	421.51
	10	1052	58000	199561	257561	63081	469.92

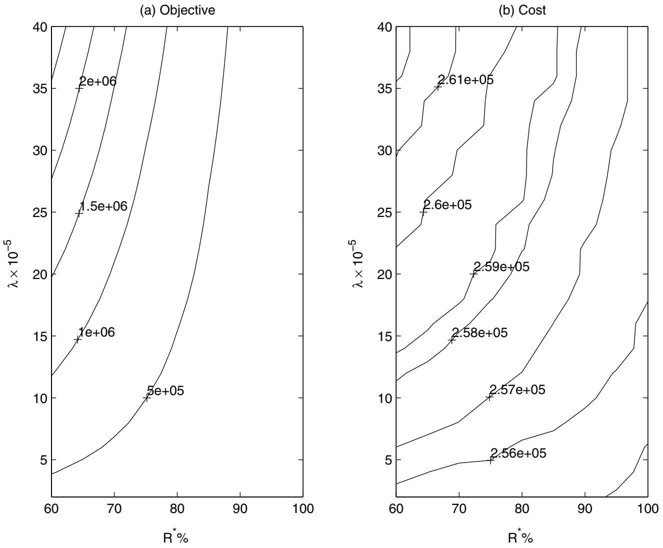

Fig. 2. Contour graphs for 20TERM demonstrating the objective value $cx^* + \sum_{k=1}^{K} p_k Q_k(x^*) + \sum_{K=1}^{K} p_k[Q_k(x^*) - R^*]_+^2$ and the cost $cx^* + \sum_{k=1}^{K} p_k Q_k(x^*)$ as functions of λ and R^*

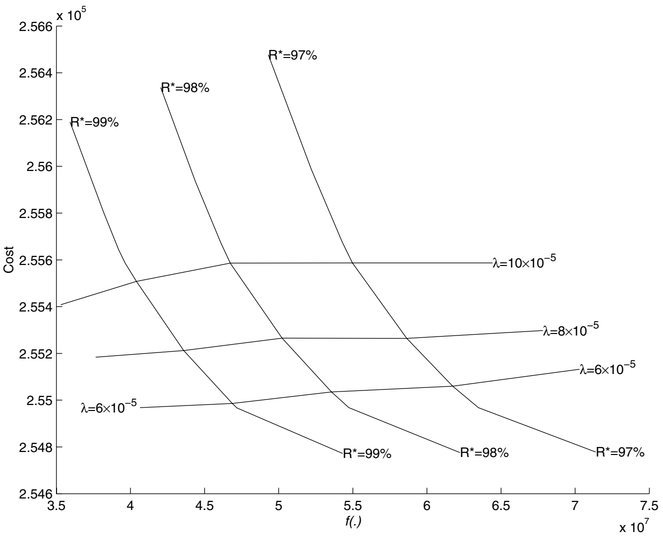

Fig. 3. Variability-cost tradeoff for different values of λ and R^* for 20TERM

costs. As a result, the overall cost increases. This behavior is clearer in Figure 3, where cost-variability trade-off curves for several different values of λ and R^* for 20TERM are presented.

Finally, Table 2 and Table 3 indicate that the computational effort in terms of the number of cuts and CPU time typically increases with increase in the penalty on variability, i.e., increasing λ and decreasing R^*.

4.2. Auctioning short-term electricity

Our second set of experiments is based on a problem that is frequently faced by electric-power producers, namely that of auctioning excess generation capacity. A power producer often finds itself in the position of having excess power at its disposal as a result of over-committing its generators. Excess power is sold in the short-term market, usually 24 hours, by holding an auction. A buyer submits its required capacity, which reflects the maximum amount of power that the buyer has the right, but not the obligation, to consume at time period t. When a bid i is submitted, the buyer provides the acceptable fixed charge, α_i, and the desired variable charge, β_i.

Once all bids are received, the producer creates a set of scenarios, $d^i_{tk}, t = 1, \dots, T$, $k = 1, \dots, K$, for the potential consumption of each contract – bid i – at each time period t. Given the probability $p_k, k = 1, \dots, K$, of each scenario, the expected future revenue of bid i is $-c_i = \alpha_i + \beta_i \sum_{k=1}^K p_k \sum_{t=1}^T d_{tk}$, $i = 1, \dots, I$. We denote the producer's excess capacity by C_{tk} and the cost of generating the additional power by $q_{tk}(\cdot)$, where q_{tk} is a piecewise-linear convex function. Note that both C_{tk} and q_{tk} vary with time t and scenario k as they are the result of solving a stochastic optimization problem [24]. Then, the producer's problem can be posed as

$$\min_{x_i, y_{tk}} \Bigg\{ \sum_{i=1}^{I} c_i x_i + \sum_{k=1}^{K} p_k \sum_{t=1}^{T} q_{tk}(y_{tk}) + \lambda \sum_{k=1}^{K} p_k \left[\sum_{t=1}^{T} q_{tk}(y_{tk}) - R^* \right]_+^2 \mid x_i \in \{0, 1\},$$
$$i = 1, \dots, n, \; \sum_{i=1}^{n} d^i_{tk} x_i \le y_{tk} \le C_{tk}, \; t = 1, \dots, T, \; k = 1, \dots, K \Bigg\}, \qquad (14)$$

where x_i denotes whether a bid is accepted or rejected, and y_{tk} represents the amount of power to be produced. As before, we use the variability measure $\sum_{k=1}^K p_k$ $[\sum_{t=1}^T q_{tk}(y_{tk}) - R^*]_+^2$ to control the second-stage cost. Note that the first-stage variable x is binary and that the second-stage program is a multi-period model that is separable in its periods. Furthermore, without the variability component, the problem has a piece-wise linear objective function and can be reformulated into a mixed-integer linear program.

We use a decomposition-based branch-and-cut strategy for model (14). The integrality of the first-stage binary variables is enforced by a branch-and-bound scheme. In each node of the branch-and-bound tree, we solve the continuous relaxation of (14) using the linear-programming-based decomposition algorithm of Section 3.2. We use the mixed-integer solver of CPLEX 7.5 to manage the branch-and-bound tree in conjunction with call-back routines to our C++ implementation of (LD) for solving the continuous relaxations.

Our experimental data for model (14) is based on the generation system of a power company based in the Midwestern US. The number of periods is $T = 24$ hours. We consider two sets of problems. The first has $n = 100$ bids and $K = 200$ scenarios, while the second has $n = 200$ bids and $K = 200$ scenarios. The computational results for these data sets are summarized in Table 4 and Table 5. The first column lists the target value R^* as a percentage of the optimal expected second-stage cost for the standard stochastic programming formulation. The second column labeled "λ" provides the value of the penalty parameter in multiples of 10^{-6}. The statistics related to solving the root node of the branch-and-bound tree is listed in the third and fourth columns. The third column provides the number of cuts needed to solve the continuous relaxation in the root node to a relative accuracy of 10^{-6} using algorithm (LD), and the fourth column provides the corresponding execution time in seconds. Columns 5, 6, and 7 provide statistics related to the overall branch-and-bound search. Column "Nodes" is the total number of nodes in the branch-and-bound tree; "Inc." is the number of incumbent solutions found; and "CPU" is the total execution time in seconds. The last two columns display the overall expected cost and variability of the solutions.

The results in Table 4 and Table 5 demonstrate the usefulness of the robust optimization framework in trading-off cost variability with expected costs. It appears that significant reduction in cost variability is possible with marginal increase in the expected costs. It is also interesting to note that the execution time tends to decrease as the penalty increases. This may be attributed to the fact that higher penalties reduce the set of solutions that need to be searched by the branch-and-bound algorithm. Incidentally,

Table 4. Results for an auction with 100 bids and 200 scenarios. The values of λ are in multiples of 10^{-6} and those of $f(.)$ are in multiples of 10^6

		Root		Branch & Bound			Objective	
R^*	λ	Cuts	CPU	Nodes	Inc.	CPU	Cost	$f(.)$
–	0	368	22.51	14899	179	107.84	–663305	–
100%	1	361	25.99	17249	212	128.87	–663305	630
	5	368	27.78	27435	342	218.2	–661306	6
	10	372	27.53	23272	307	182.06	–661306	6
90%	1	341	24.85	17300	310	136.82	–658629	2630
	5	382	30.07	12262	217	113.92	–652026	724
	10	452	37.78	10415	164	116.85	–651215	639
80%	1	359	26.07	5204	201	69.38	–652026	24091
	5	168	9.54	63	2	10.49	–613865	8501
	10	113	6.06	3980	97	18.64	–563850	2129
70%	1	255	16.02	3258	106	34.54	–641675	71495
	5	33	1.72	20	4	2.12	–511100	18130
	10	34	1.8	57	7	2.32	–419576	4020
60%	1	177	10.11	682	28	13.89	–628611	145902
	5	27	1.41	240	17	2.63	–344328	21001
	10	27	1.43	755	14	2.89	–246979	6347

Table 5. Results for an auction with 200 bids and 200 scenarios. The values of λ are in multiples of 10^{-6} and those of $f(.)$ are in multiples of 10^6

		Root		Branch & Bound			Objective	
R^*	λ	Cuts	CPU	Nodes	Inc.	CPU	Cost	$f(.)$
–	0	219	23.86	8561	106	72.44	–683928	–
100%	1	227	26.26	6616	113	71.89	–683928	210
	5	295	26.33	20046	121	148.69	–683928	210
	10	327	41.98	68885	139	395.29	–683928	210
90%	1	329	41.76	4601	83	81.88	–680733	10450
	5	163	18.03	2518	35	29.42	–661300	3165
	10	162	17.48	3711	31	30.83	–647234	1421
80%	1	311	38.33	756	29	48.62	–672797	42551
	5	57	5.9	537	15	8.67	–599236	13643
	10	37	3.86	672	17	7	–529397	3221
70%	1	185	20.53	610	21	26.03	–661300	99763
	5	27	2.91	27	4	3.58	–456955	16602
	10	30	3.11	51	4	3.95	–382324	5348
60%	1	82	8.63	6169	59	25.87	–647234	182488
	5	33	3.47	26	4	4.24	–305622	22427
	10	32	3.36	248	5	4.65	–207044	7335

CPLEX failed to find a feasible solution to the deterministic equivalent of either problem when $\lambda = 0$ with a CPU-time limit of two hours.

Acknowledgements. The authors thank A. R. Conn and G. Paleologo of IBM Research, and A. Shapiro of Georgia Institute of Technology for helpful discussions. We also thank two anonymous referees and the Associate Editor for constructive comments.

References

1. Ahmed, S., Sahinidis, N.V.: Robust process planning under uncertainty. Industrial & Engineering Chemistry Research **37**(5), 1883–1892 (1998)
2. Bai, D., Carpenter, T., Mulvey, J.M.: Making a case for robust models. Management Sci. **43**, 895–907 (1997)
3. Bawa, V.: Optimal rules for ordering uncertain prospects. J. Financial Economics **2**(1), 95–121 (1975)
4. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming, Springer Verlag, 1997
5. Birge, J.R.: Option methods for incorporating risk into linear capacity planning models. Manufacturing & Service Operations Management **2**(1), 19–31 (2000)
6. Fishburn, P.C.: Mean-risk analysis with risk associated with below-target returns. American Economic Rev. **67**(2), 116–125 (1977)
7. Geoffrion, A. M.: Generalized Benders decomposition. J. Optim. Theory & Appl. **10**, 237–260 (1972)

8. Huang, C.-F., Litzenberger, R.H.: Foundations for Financial Economics. Prentice Hall, 1988
9. Kall, P., Wallace, S.W.: Stochastic Programming. John Wiley & Sons, 1994
10. King, A.J.: Asymmetric risk measures and tracking models for portfolio optimization under uncertainty. Ann. Operations Res. **45**, 165–177 (1993)
11. King, A.J., Takriti, S., Ahmed, S.: Issues in risk modeling for multi-stage systems. IBM Research Report, RC-20993, 1997
12. King, A.J.: Duality and martingales: a stochastic programming perspective on contingent claims. Math. Programming **91**(3), 543–562 (2002)
13. Laguna, M.: Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty. Management Sci. **44**, S101–S110 (1998)
14. Linderoth, J., Shapiro, A., Wright, S.: The empirical behavior of sampling methods for stochastic programming. Optimization Technical Report 02-01, Computer Sciences Department, University of Wisconsin, January 2002, `http://www.cs.wisc.edu/~swright/`
15. Luenberger, D.G.: Investment Science. Oxford University Press, 1998
16. Mak, W.-K., Morton, D.P., Wood, R.K.: Monte Carlo bounding techniques for determining solution quality in stochastic programs. Operations Res. Lett. **24**, 47–56 (1999)
17. Malcolm, S.A., Zenios, S.A.: Robust optimization of power systems capacity expansion under uncertainty. J. operational res. soc. **45**, 1040–1049 (1994)
18. Mulvey, J.M., Vanderbei, R.J., Zenios, S.A.: Robust optimization of large-scale systems. Operations Res. **43**, 264–281 (1995)
19. Mulvey, J.M., Ruszczyński, A.: A new scenario decomposition method for large scale stochastic optimization. Operations Res. **43**, 477–490 (1995)
20. Nawrocki, D.: A brief history of downside risk measures. J. Investing 9–26, Fall 1999
21. Porter, R.B.: Semivariance and stochastic dominance: A comparison, the American economic review, **64**(1), 200–204 (1974)
22. Rockafellar, R.T.: Convex Analysis, Princeton University Press, 1970
23. Sen, S., Higle, J.L.: An introductory tutorial on stochastic linear programming. Interfaces **29**, 33–61 (1999)
24. Takriti, S., Krasenbrink, B., Wu, L.S.-Y.: Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. Operations Research **48**(2), 268–280 (2000)
25. van Slyke, R., Wets, R.J.-B.: L-shaped linear programs with applications to linear control and stochastic programming. SIAM J. Appl. Math. **17**(4), 638–663 (1969)