


# Robust Optimization of Large-Scale Systems

John M. Mulvey, Robert J. Vanderbei, Stavros A. Zenios,

# ROBUST OPTIMIZATION OF LARGE-SCALE SYSTEMS

JOHN M. MULVEY and ROBERT J. VANDERBEI

*Princeton University, Princeton, New Jersey*

STAVROS A. ZENIOS

*University of Cyprus, Nicosia, Cyprus*

(Received June 1991; revision received July 1994; accepted August 1994)



Mathematical programming models with noisy, erroneous, or incomplete data are common in operations research applications. Difficulties with such data are typically dealt with *reactively*—through sensitivity analysis—or *proactively*—through stochastic programming formulations. In this paper, we characterize the desirable properties of a solution to models, when the problem data are described by a set of *scenarios* for their value, instead of using point estimates. A solution to an optimization model is defined as: *solution robust* if it remains "close" to optimal for all scenarios of the input data, and *model robust* if it remains "almost" feasible for all data scenarios. We then develop a general model formulation, called *robust optimization* (RO), that explicitly incorporates the conflicting objectives of solution and model robustness. Robust optimization is compared with the traditional approaches of sensitivity analysis and stochastic linear programming. The classical diet problem illustrates the issues. Robust optimization models are then developed for several real-world applications: power capacity expansion; matrix balancing and image reconstruction; air-force airline scheduling; scenario immunization for financial planning; and minimum weight structural design. We also comment on the suitability of parallel and distributed computer architectures for the solution of robust optimization models.

Whenever operations researchers attempt to build a model of a real-world system, they are faced with the problem of noisy, incomplete, or erroneous data. This is true irrespectively of the application domain. In business applications noisy data are prevalent. Returns of financial instruments, demand for a firm's products, the cost of fuel, consumption of power and other resources, are typical examples of model data that are usually known with some probabilistic distribution. In social sciences, data are often incomplete as, for example, in partial census surveys that are carried out periodically in lieu of a complete census of the population. Morgenstern's (1963) book is devoted to problems arising in economic modeling from incomplete data. In the physical sciences and engineering, data are usually subject to measurement errors. Such is the case, for example, in models of image restoration from remote sensing experiments and other inverse problems.

In contrast, the world of mathematical programming models is generally assumed to be deterministic. Models are typically formulated by "best guessing" uncertain values, or by solving "worst-case" problems. The solutions to such "worst-case" or "mean-value" problems are inadequate. Birge (1982) established the large error bounds that arise when one solves mean value problems. Worst-case formulations produce very conservative and, potentially, expensive solutions.

To reconcile the contradictions between the real-world data, and the realm of mathematical programming, management scientists employ *sensitivity analysis*. The goal of these post-optimality studies is to discover the impact of data perturbations on the model's recommendations. Such post-optimality studies are *reactive*; they only discover the impact of data uncertainties on the model's recommendations.

We believe that a *proactive* approach is needed. That is, we need model formulations that, by design, yield solutions that are less sensitive to the model data, than classical mathematical programming formulations. An approach that introduces probabilistic information about the problem data is that of *stochastic linear programming* which dates back to Beale (1955) and Dantzig (1955); see also Wets (1966, 1974, 1983).

In this paper we suggest an alternative approach, which we call *robust optimization* (RO). This approach integrates *goal programming* formulations with a *scenario-based* description of problem data. It generates a series of solutions that are progressively less sensitive to realizations of the model data from a scenario set. Robust optimization, while not without limitations, has some advantages over stochastic linear programming and is more generally applicable.

The need for robustness has been recognized in a number of application areas. Paraskevopoulos, Karakitsos and Rustem (1991) propose a capacity planning model for the plastics industry. They show it to be effective in controlling the model's recommendations to the uncertain data of the application. (Their formulation, developed independently from our work, can be cast as a special

*Subject classifications:* Finance: portfolio optimization. Programming: stochastic programming. Simulation: large-scale optimization.
*Area of review:* COMPUTING.



0030-364X/95/4302-0264 $01.25

case of RO). Sengupta (1991) discusses the notion of robustness for stochastic programming models. Escudero et al. (1993) presents an RO formulation for the problem of outsourcing in manufacturing, and Gutierrez and Kouvelis (1995) develop RO models for multinational production scheduling. The last two references are direct applications of the general framework developed here.

The rest of the paper is organized as follows: Section 1 defines the framework of RO and presents an illustrative example. Subsection 1.4, in particular, compares the RO framework with existing approaches for dealing with data uncertainty, i.e., sensitivity analysis and stochastic linear programming. RO is not a panacea for mathematical programming in the face of noisy data. Instead, we show in Section 2 how to introduce robustness in several real-world applications. Implementations of the models serve to illustrate the advantages of the RO formulations. We illustrate the generality of RO by showing that some well known mathematical programming formulations can be obtained as special cases.

The RO formulations are more complex, and computationally more expensive, than their linear programming counterparts. In Section 3 we discuss briefly solution options for the RO models that can be of extremely large size; parallel and distributed computers provide the required computational environment for solving RO. Concluding remarks are given in Section 4, where we also discuss open issues.

## 1. GENERAL MODELING FRAMEWORK OF ROBUST OPTIMIZATION

We are dealing with optimization models that have two distinct components: a *structural* component that is fixed and free of any noise in its input data, and a *control* component that is subjected to noisy input data. To define the appropriate model we introduce two sets of variables:

$x \in \mathbf{R}^{n_1}$, denotes the vector of decision variables whose optimal value is not conditioned on the realization of the uncertain parameters. These are the *design* variables. Variables in this set cannot be adjusted once a specific realization of the data is observed.

$y \in \mathbf{R}^{n_2}$, denotes the vector of *control* decision variables that are subjected to adjustment once the uncertain parameters are observed. Their optimal value depends both on the realization of uncertain parameters, and on the optimal value of the design variables.

The terminology of *design* and *control* variables is borrowed from the flexibility analysis of production and distribution processes (Seider et al. 1991). Design variables determine the structure of the process and the size of production modules. Control variables are used to adjust the mode and level of production in response to disruptions in the process, changes in demand or production yield, and so on.

The optimization model has the following structure.

### LP

$$\text{Minimize} \quad c^{\top}x + d^{\top}y \qquad x \in \mathbf{R}^{n_1},\ y \in \mathbf{R}^{n_2} \tag{1}$$

$$\text{subject to} \quad Ax = b, \tag{2}$$

$$Bx + Cy = e, \tag{3}$$

$$x, y \geq 0. \tag{4}$$

Equation 2 denotes the structural constraints whose coefficients are fixed and free of noise. Equation 3 denotes the control constraints. The coefficients of this constraint set are subject to noise.

To define the robust optimization problem, we now introduce a set of scenarios $\Omega = \{1, 2, 3, \ldots, S\}$. With each scenario $s \in \Omega$ we associate the set $\{d_s, B_s, C_s, e_s\}$ of realizations for the coefficients of the control constraints, and the probability of the scenario $p_s$, $(\sum_{s=1}^{S} p_s = 1)$. The optimal solution of the mathematical program (1)–(4) will be robust with respect to optimality if it remains "close" to optimal for any realization of the scenario $s \in \Omega$. It is then termed *solution robust*. The solution is also robust with respect to feasibility if it remains "almost" feasible for any realization of $s$. It is then termed *model robust*. The notions of "close" and "almost" are made precise through the choice of norms, later in this section.

It is unlikely that any solution to program (1)–(4) will remain both feasible and optimal for all scenario indices $s \in \Omega$. If the system that is being modeled has substantial redundancies built in, then it might be possible to find solutions that remain both feasible and optimal. Otherwise, a model is needed that will allow us to measure the tradeoff between solution and model robustness. The robust optimization model proposed next formalizes a way to measure this tradeoff.

We first introduce a set $\{y_1, y_2, \ldots, y_S\}$ of control variables for each scenario $s \in \Omega$. We also introduce a set $\{z_1, z_2, \ldots, z_S\}$ of error vectors that will measure the infeasibility allowed in the control constraints under scenario $s$. Consider now the following formulation of the robust optimization model.

### Model ROBUST

$$\text{Minimize} \quad \sigma(x, y_1, \ldots, y_S) + \omega\rho(z_1, \ldots, z_S) \tag{5}$$

$$\text{subject to:} \quad Ax = b, \tag{6}$$

$$B_s x + C_s y_s + z_s = e_s, \qquad \text{for all } s \in \Omega, \tag{7}$$

$$x \geq 0,\ y_s \geq 0, \quad \text{for all } s \in \Omega. \tag{8}$$

With multiple scenarios, the objective function $\xi = c^{\top}x + d^{\top}y$ becomes a random variable taking the value $\xi_s = c^{\top}x + d_s^{\top}y_s$, with probability $p_s$. Hence, there is no longer a single choice for an aggregate objective. We could use the mean value

$$\sigma(\cdot) = \sum_{s \in \Omega} p_s \xi_s, \tag{9}$$

which is the function used in stochastic linear programming formulations. In worst-case analysis the model minimizes the maximum value, and the objective function is defined by

$$\sigma(\cdot) = \max_{s \in \Omega} \xi_s. \tag{10}$$

Both of these choice are special cases of RO, but they are nevertheless standard in the literature. One novelty of the RO formulation is that it allows the introduction of higher moments of the distribution of $\xi_s$ in the optimization model. For example, we could introduce a utility function that embodies a tradeoff between mean value and variability in this mean value. Indeed, the introduction of higher moments is one of the distinguishing features of RO from stochastic linear programming. There will be more on this in subsections 1.2 and 1.4. For now, we summarize all possible choices by calling the aggregate function $\sigma(\cdot)$.

The second term in the objective function $\rho(z_1, \ldots, z_S)$ is a feasibility penalty function. It is used to penalize violations of the control constraints under some of the scenarios. The model proposed above takes a multicriteria objective form. The first term measures optimality robustness, whereas the penalty term is a measure of model robustness. The goal programming weight $\omega$ is used to derive a spectrum of answers that tradeoff solution for model robustness.

The introduction of the penalty function distinguishes the robust optimization model from existing approaches for dealing with noisy data. In particular, the model recognizes that it may not always be possible to get a feasible solution to a problem under all scenarios. Infeasibilities will inevitably arise; they will be dealt with *outside* the optimization model. For example, Prékopa (1980) suggested possible ways for the treatment of infeasibilities in network planning. The RO model will generate solutions that present the modeler with the least amount of infeasibilities to be dealt with outside the model. This use of a penalty function is distinct from the use of penalty methods for the solution of constrained optimization problems; see, e.g., Bertsekas (1982). The specific choice of penalty function is problem dependent, and it also has implications for the accompanying solution algorithm. We consider two alternative penalty functions:

$\boldsymbol{\rho}(z_1, \ldots, z_S) = \Sigma_{s \in \Omega} p_s z_s^{\top} z_s$. This quadratic penalty function is applicable to equality constrained problems where both positive and negative violations of the control constraints are equally undesirable. The resulting RO model is a quadratic programming problem.

$\boldsymbol{\rho}(z_1, \ldots, z_S) = \Sigma_{s \in \Omega} p_s \max\{0, z_s\}$. This exact penalty function applies to inequality control constraints when only positive violations are of interest. (Negative values of $z$ indicate slack in the inequality constraints.) With the addition of a slack variable this penalty function can be expressed using linear inequality constraints. Doing so, however, increases the size of the problem and destroys the underlying structure of the constraint matrix. Another approach is to work directly with the nondifferentiable penalty function. An $\epsilon$-smoothing of the exact penalty function results in a differentiable problem which is easier to solve, and it preserves the structure of the constraint matrix of the RO formulation. The smoothed problem will produce a solution that is within $\epsilon$ of the solution of the nondifferentiable problem; see Pinar and Zenios (1992) and Zenios, Pinar and Dembo (1994).

### 1.1. Example: The Diet Problem

We illustrate the RO concepts on the *diet problem*, first studied by Stigler (1945) and employed by Dantzig (1963) as the first test case for the simplex algorithm. This example shows the importance of *model robustness*, which is particularly novel in the context of optimization formulations: Feasibility is usually overemphasized in optimization models. The importance of *solution robustness* is more easily accepted. It also has been addressed, in other ways, by stochastic linear programming, and will be illustrated in the examples of Section 2.

The problem consists of finding a diet of minimum cost that will satisfy certain nutritional requirements. Stigler was faced with a problem of robust optimization as he had recognized in his paper, because the nutritional content of some food products may not be certain. Dantzig (1990) was still intrigued by this ambiguity in his article in *Interfaces*. He wrote:

> When is an apple an apple and what do you mean by its cost and nutritional content? For example, when you say *apple* do you mean a Jonathan, or McIntosh, or Northern Spy, or Ontario, or Winesap, or Winter Banana? You see, it can make a difference, for the amount of ascorbic acid (vitamin C) can vary from 2.0 to 20.8 units per 100 grams depending upon the type of apple.

The standard linear programming formulation will assume some average content for ascorbic acid and produce a diet. However, as our consumers buy apples of different ascorbic acid contents, they will soon build a deficit or surplus of vitamin C. This situation may be irrelevant for a healthy individual over long periods of time, or it may require remedial action in the form of vitamin supplements.

We use the diet model from the GAMS library (Brooke, Kendrick and Meeraus 1992) to illustrate how the diet problem can be cast in the framework of robust optimization. (This GAMS model does not include apples as part of the food selection, so we analyze the ambiguity on the calcium content of navybeans.) Let $x_f$ denote the dollar value of food-type $f$ in the diet, let $a_{fn}$ denote the contents of food $f$ in nutrient $n$ per dollar spent, and let $b_n$ be the required daily allowance of nutrient $n$. We also use $c$ to denote calcium from the set of nutrients and $N$ to denote navybeans, which is the food product with uncertain nutritional content. A point estimate for its calcium content is $a_{Nc} = 11.4$ per dollar spent. In our example, we assume that this coefficient takes values from the set of scenarios $\Omega = \{S_1, \ldots, S_8, \ldots, S_{13}\} \doteq \{9.5, 9.75, 10, 10.25, 10.5, 11, 11.25, 11.4, 11.5, 11.75, 12, 12.25, 12.5\}$. All values are equally likely, and $S_8$ denotes the scenario corresponding to the point estimate.

The robust optimization formulation of the diet problem can be stated as follows.

$$\text{Minimize}_{x_f} \sum_f x_f + \omega \frac{1}{|S|} \sum_{s \in \Omega} \left[ b_c - \sum_{f \neq N} a_{fc} x_f - a^s_{Nc} x_{Nc} \right]^2 \tag{11}$$

subject to

$$\sum_f a_{fn} x_f \geqslant b_n, \qquad \text{for all } n. \tag{12}$$

The weight $\omega$ is used to tradeoff feasibility robustness with cost. For $\omega = 0$ we are solving the classical linear programming formulation of the diet problem. In Figure 1 we plot the deficit/surplus of calcium of this optimal diet as navybeans of different quality are purchased. We then solve the robust optimization model. We plot the deficit/surplus of calcium of the optimal diet obtained with increasing values of $\omega$. For larger values of $\omega$ we obtain diets whose nutritional content varies very little as the quality of navybeans changes. This is also shown in Figure 1. Figure 2 shows the tradeoff in the cost of the diet as it becomes more robust with respect to nutritional content.

This simple example clarifies the meaning of a robust solution, and shows that robust solutions are possible, but at some cost. It is interesting to observe that a solution that is less sensitive to uncertainty than the linear programming solution is possible at very little cost (see the error curve and the cost function value corresponding to $\omega = 1.0$).

A reasonably good diet is the one obtained with $\omega = 5.0$, because it is quite insensitive to uncertainty, and not much more expensive than the linear programming solution. For example, if an error of $\pm 0.02$ units in total calcium intake is acceptable, no remedial action will be needed for this RO diet. On the other hand, the linear programming diet will need remedial treatment for 10 out of the 13 scenarios. The RO diet is only 4% more expensive than the diet produced by the linear program.[1]

### 1.2. The Choice of Norms: High Risk Decisions

In this section, we investigate possible choices for the model robustness term, $\sigma(\cdot)$. In low risk situations $\sigma(\cdot)$ can be taken to be the expected value given by (9), which is the objective of stochastic linear programs. This choice is inappropriate for moderate and high-risk decisions under uncertainty. Most decision makers are risk averse for important decisions. The expected value

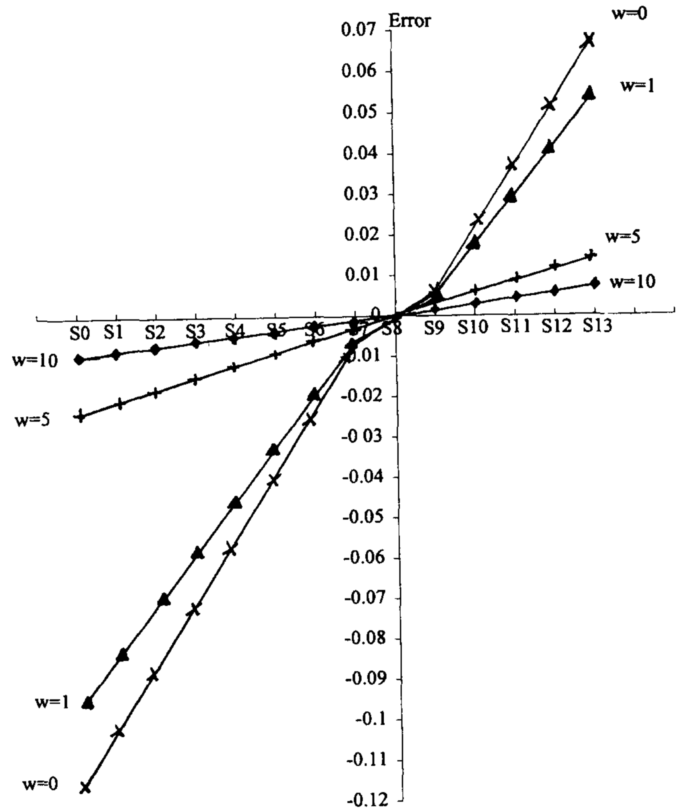



**Figure 1.** Error (negative for deficit, positive for surplus) of the dieter's intake of calcium as navybeans of different quality are added in the diet. The horizontal axis corresponds to scenarios for calcium contents of navybeans: $S1$ corresponds to navybeans with low calcium content of 9.5; $S13$ corresponds to high calcium content of 12.50. The vertical axis corresponds to (absolute) error in the target calcium level of the dieter. The diet obtained with the linear programming formulation (i.e., $w = 0$) is very sensitive to the calcium content of navybeans, whereas the diets obtained with the robust optimization model (i.e., $w = 1$–$10$) are much less sensitive.

objective ignores both the risk attribute of the decision maker, and the distribution of the objective values $\xi_s$.

Two popular approaches for handling risk are: mean/variance models (Markowitz 1991), and von Neumann-Morgenstern (1953) expected utility models. For the former, risk is equated with the variance of the outcomes. A high variance for $\xi_s = c^T x + d_s^T y_s$ means that the outcome is much in doubt. Given outcome variance as a surrogate for risk, we are naturally led to the minimization of the expected outcome for a given level of risk. An appropriate choice for $\sigma(\cdot)$ would be the mean plus a constant ($\lambda$) times the variance

$$\sigma(x, y_1, y_2, \ldots, y_S) = \sum_{s \in S} p_s \xi_s + \lambda \sum_{s \in S} p_s \left( \xi_s - \sum_{s' \in S} p'_s \xi_{s'} \right)^2. \tag{13}$$

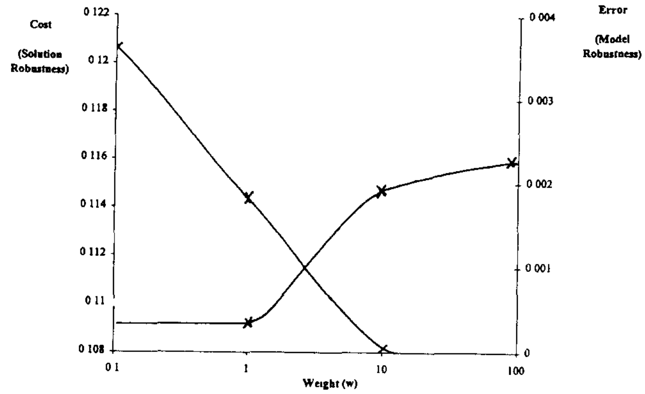



**Figure 2.** Tradeoff between cost (i.e., solution robustness) and expected error in the calcium contents (i.e., model robustness) for diets obtained using increasing weight $w$ in the robust optimization model.

An efficient frontier can be constructed readily by parametrizing the tradeoff between risk and expected outcome, as shown in Figure 3. This approach requires that the distribution of the random variable $\xi_s$ be symmetric around its mean. Third and higher moments are ignored. Still, the mean/variance approach is popular in financial planning and other areas. In finance, a robust portfolio is equated with a well diversified and hedged investment mix. The portfolio should do well under a variety of economic circumstances. Identifying the investor's risk, however, requires the integration of assets and liabilities and it is easy to render incorrect decisions (Berger and Mulvey 1994). The objective function should reflect the investor's net wealth:

Wealth = market value of assets
 − net present value of liabilities.

Risk is then associated with the variance of "wealth," sometimes called surplus. The model requires the

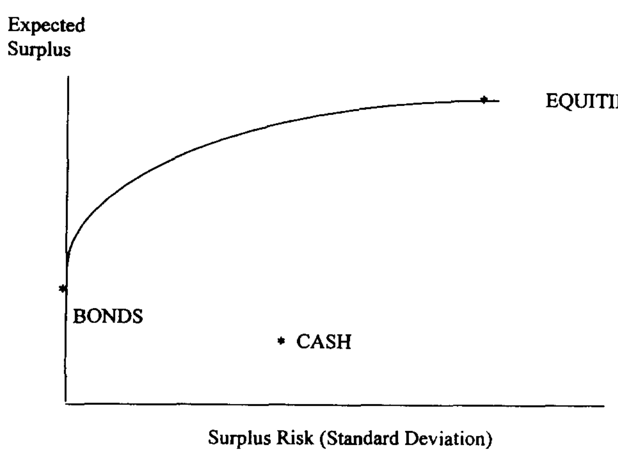



**Figure 3.** Efficient frontier for robust optimization (application from surplus optimization in asset allocation for financial planning).

computation of the covariances between the asset categories and the net present value of the liabilities. Figure 3 illustrates an example in which the cash asset poses a risk to the investor's wealth. In other applications, the outcomes for defining risk depend upon problem circumstances. For some recent applications of robust optimization in financial planning see Berger and Mulvey (1994), Carino et al. (1994), and Golub et al. (1994).

The derivation of the efficient frontier gives the user an opportunity to achieve a *robust* recommendation, which is not possible by means of traditional sensitivity analysis. The risk-return curve as shown in Figure 3 depicts the range of possible levels of solution robustness. As $\lambda$ increases we are lead to solutions that are less sensitive to changes in the data as defined by the scenarios.

An alternative and more general approach to handling risk is based on von Neumann-Morgenstern utility curves (von Neumann and Morgenstern 1953, Keeney and Raiffa, 1976) via the concept of certainty equivalence. The result is a concave utility function (for risk averse decision makers − $U(\cdot)$. A decision maker displays consistent behavior by maximizing expected utility. In this situation we define:

$$\sigma(\cdot) = -\sum_{s \in \Omega} p_s U(\xi_s).$$

The primary advantage of the expected utility model over the mean-variance approach is that asymmetries in the distribution of outcomes $\xi_s$ are captured. A consistent and repeatable decision process can also be implemented, given a time-invariant utility function. There is an additional information burden placed on the user who has to decide upon an appropriate level of risk tolerance.

### 1.3. Stochastic Linear Programming

This section defines the multistage stochastic linear programming model. Stochastic linear programs avoid the use of the penalty terms and generally minimize expected costs or maximize expected profits. Furthermore, the notion of stabilizing the solution over a period of time does not arise in stochastic linear programs. In the RO approach, the variance term, the risk aversion parameter, or the min-max strategy can be employed to reduced variability. This aspect is critical in many applications. Also, robust optimization allows for infeasibilities in the control constraints by means of the penalties.

To define a stochastic linear program, we assume that there is some underlying probability space $(\Omega, \mathcal{B}, P)$, a measurable objective function $f: \Omega \times \mathbf{R}^n \to \mathbf{R}$, a measurable multivalued mapping $X: \Omega \to 2^{\mathbf{R}^n}$ representing (event-dependent) constraints, a space $\mathcal{X}$ of measurable decision rules $x: \Omega \to \mathbf{R}^n$ and a subspace $\mathcal{M} \subset \mathcal{X}$ of implementable decision rules. For each elementary event $\omega \in \Omega$ we denote by $X_\omega$, $x_\omega$ and $f_\omega(x_\omega)$ the corresponding constraint set, decision and objective values. The problem is formulated as follows:

Find a decision rule $x:\Omega \to \mathbf{R}^n$ that minimizes

$$\int f_\omega(x_\omega)P\,(d\omega)$$

subject to $x_\omega \in X_\omega$ with probability 1, and $x \in \mathcal{M}$.

A particularly interesting and important case arises when the decision problem has a dynamic structure with time stages $t = 1, \ldots, T$ and

$$x_\omega = (x_\omega(1), x_\omega(2), \ldots, x_\omega(T)).$$

Typically, we interpret the elementary events $\omega \in \Omega$ as scenarios and use $X_\omega$ to represent the conditions that have to be satisfied by the decision sequence $x_\omega$ for each scenario. Condition $\mathcal{M}$ usually represents nonanticipativity constraints: For each $t$ decisions $x_\omega(t)$ must be equal for all scenarios $\omega$ that have a common past and present. Formally, this can be stated as a condition of measurability of $x_\omega(t)$ with respect to some $\sigma$-subfield $\mathcal{B}(t) \subseteq \mathcal{B}$, where $\mathcal{B}(t)$, $t = 1, \ldots, T$ is an increasing sequence of $\sigma$-subfields.

To be more specific, consider stochastic linear programs in conjunction with a finite probability space $\Omega$. (Continuous distributions cause severe modeling problems when correlations exist for the random variables and these have been avoided.) Let $D_\omega(t)$ and $H_\omega(t)$, $t = 1, \ldots, T$ be sequences of random $m_b \times m_x$ matrices and $b_\omega(t)$ and $c_\omega(t)$, $t = 1, \ldots, T$, be sequences of random vectors in $R^{m_b} \times R^{m_x}$, respectively. We will call each sequence

$$s_\omega(t) = (D_\omega(t), H_\omega(t), b_\omega)(t), c_\omega(t)), \quad t = 1, \ldots, T,$$

corresponding to some event $\omega \in \Omega$, a scenario. The problem is to find a collection $x_\omega(t)$, $t = 1, \ldots, T$, $\omega \in \Omega$ of random vectors in $R^{m_x}$ (a policy), which minimizes the linear form

$$\sum_{\omega \in \Omega} p_\omega \sum_{t=1}^{T} \langle c_\omega(t), x_\omega(t) \rangle \tag{14}$$

subject to the constraints

$$d_\omega(t)x_\omega(t-1) + H_\omega(t)x_\omega(t) = b_\omega(t), \quad t = 1, \ldots, T, \tag{15}$$

and $x_\omega(t) \geq 0$, $t = 1, \ldots, T$, $\omega \in \Omega$, with $x(0) = x_0$ fixed. The nonanticipativity constraint can be formulated as follows: For all $\omega$, $\zeta \in \Omega$ and any $t \in \{1, \ldots, T\}$

$$x_\omega(t) = x_\zeta(t) \text{ if } s_\omega(\tau) = s_\zeta(\tau) \quad \text{for } \tau = 1, \ldots, t. \tag{16}$$

In words, decisions corresponding to scenarios that are indistinguishable up to time $t$ should be identical.

Stochastic linear programs have been studied for four decades, starting with the early work by Dantzig (1955) and Beale (1955), and later by Wets (1974) and others. Despite these efforts and until recently, there have been few genuine applications of stochastic linear programs due to several interrelated factors. First, the models rapidly enlarge as a function of the number of time stages and scenarios. This computational issue has become less critical since the advent of more powerful computers and highly efficient solution algorithms. Second, stochastic linear programs do not handle risk aversion in a direct fashion. This restriction has excluded many important domains of application. But, again, the computational constraints have diminished as algorithms, such as nonlinear interior point methods, have become efficient for solving larger problems.

Next, it is often assumed that the second and subsequent decision stages display complete recourse. Thus, there is no need to worry about feasibility. This simplification can be overcome but with added modeling complexity, such as the penalty approach discussed in Section 1.

### 1.4. Comparisons With Sensitivity Analysis and Stochastic Linear Programming

We compare here RO with alternative approaches for dealing with uncertainty. We will see that RO enjoys several advantages, while it is not without its shortcomings.

Sensitivity analysis (SA) is a reactive approach to controlling uncertainty. It just measures the sensitivity of a solution to changes in the input data. It provides no mechanism by which this sensitivity can be controlled. For example, applying SA to the linear programming diet (subsection 1.1) we estimate a 6% change in the calcium intake of the dieter, per unit change in the calcium contents of the food products. By comparison, the SA of the RO diet (for $\omega = 5$) indicates a sensitivity of 1%. Using larger values of $\omega$ we can reduce the sensitivity even further.

Stochastic linear programming (SLP) is, similarly to robust optimization, a constructive approach. They are both superior to SA. With stochastic linear programming models the decision maker is afforded the flexibility of recourse variables. These are identical to the control variables of RO and provide the mechanism with which the model recommendations can be adjusted to account for the data realizations.

The SLP model, however, optimizes only the first moment of the distribution of the objective value $\xi_s$. It ignores higher moments of the distribution, and the decision maker's preferences toward risk. These aspects are particularly important for asymmetric distributions, and for risk averse decision makers. Furthermore, aiming at expected value optimization implicitly assumes an active management style whereby the control (i.e., recourse) variables are easily adjusted as scenarios unfold. Large changes in $\xi_s$ may be observed among the different scenarios, but their expected value will be optimal. The RO model minimizes higher moments as well, e.g., the variance of the distribution of $\xi_s$. Hence, it assumes a more passive management style. Since the value of $\xi_s$ will not differ substantially among different scenarios, little or no adjustment of the control variables will be needed. In this respect RO can be viewed as an SLP, whereby the recourse decisions are implicitly restricted.

This distinction between RO and SLP is important, and defines their domain of applicability. Applied to personnel planning, for example, an SLP solution will design a workforce that can be adjusted (by hiring or layoffs) to meet demand at the least expected cost. The important consideration of maintaining stability of employment cannot be captured. The RO model, on the other hand, will design a workforce that will need few adjustments to cope with demand for all scenarios. However, this cost will be higher than the cost of the SLP solution. The importance of controlling variability of the solution (as opposed to just optimizing its first moment) is well recognized in portfolio management applications, due to the work of Markowitz. It has been ignored in most other applications of mathematical programming. The RO framework addresses this issue directly.

Another important distinction of RO from SLP is the handling of the constraints. Stochastic linear programming models aim at finding the design variable $x$ such that for each realized scenario a control variable setting $y_s$ is possible that satisfies the constraints. For systems with some redundancy such a solution might always be possible. The SLP literature even allows for the notion of complete recourse, whereby a feasible solution $y_s$ exists for all scenarios, and for any value of $x$ that satisfies the control constraints. What happens in cases where no feasible pair $(x, y_s)$ is possible for every scenario? The SLP model is declared infeasible. RO explicitly allows for this possibility. In engineering applications (e.g., image restoration) such situations inevitably arise due to measurement errors. Multiple measurements of the same quantity may be inconsistent with each other. Hence, even if the underlying physical system has a solution (in this case, an image does exist!) it will not satisfy all the measurements. The RO model, through the use of error terms $\{z_s\}$ and the penalty function $\rho(\cdot)$, will find a solution that violates the constraints by the least amount. Such an approach is standard in medical imaging, see, e.g., the model of Levitan and Herman (1987), or the models of Elfving (1989) and Herman et al. (1990), but has received little attention in the OR literature.

Other properties of RO *vis-à-vis* SLP deserve investigation. Of particular interest is the stability of the respective solutions, see, e.g., Dupacova (1987, 1990), and the accuracy of the solutions when a limited number of scenarios is used (Ermoliev and Wets 1988).

While RO has some distinct advantages over SA and SLP, it is not without limitations. First, RO models are parametric programs and we have no a priori mechanism for specifying a "correct" choice of the parameter $\omega$. This problem is prevalent in multicriteria programming optimization (Keeney and Raiffa). Second, the scenarios in $\Omega$ are just one possible set of realizations of the problem data. RO does not provide a means by which the scenarios can be specified. This problem is prevalent in SLP models as well. Substantial progress has been made in recent years in integrating variance reduction methods, such as importance sampling, into stochastic linear programming, see Glynn and Iglehart (1989), Dantzig and Infanger (1991), and Infanger (1992). These techniques apply to RO.

Despite these potential shortcomings, we emphasize that working only with expected values (as in the linear programming formulations) is fundamentally limited for problems with noisy data. Even going a step further, that is, working with expected values and hedging against small changes in these values, is also inappropriate. This has been argued at length in the context of fixed-income portfolio management by Hiller and Schaack (1990) and was demonstrated in the application of Zenios and Kang (1993). In this respect RO provides a significantly improved modeling framework.

In summary, robust optimization integrates the methods of multiobjective programming with stochastic programming. It also extends SLP with the introduction of higher moments of the objective value, and with the notion of model robustness.

## 2. ROBUST OPTIMIZATION APPLICATIONS

This section describes the application of RO to several real-world problem domains. Most of these models were developed by the authors for diverse applications. These examples illustrate how robustness considerations can be incorporated in several important problems. They also show that RO models are solvable, even if they are more complex than the standard linear programming formulations, and the generated solutions can be robust with changes in the model data.

### 2.1. The Power Capacity Expansion Problem

The power system capacity expansion problem can be described as follows:

Select the capacities of a set of power plants that minimize the capital and operating cost of the system, meet customer demand, and satisfy physical constraints.

Demand for electric power is not constant over time: It changes during periods of the day or with the season of the year, and it exhibits long-term, yearly trends. Events like equipment and distribution line failures add to the complexities of managing such a system. Several authors have proposed stochastic programming formulations; see, e.g., Murphy, Sen and Soyster (1982), Sherali et al. (1984), Sanghvi and Shavel (1986), and Dantzig et al. (1989). An RO formulation for this problem was developed by Malcolm and Zenios (1994). It has some desirable properties: First, introducing a variance minimization term produces cost structures that are less volatile over time, and, hence, are easier to defend in front of administrative and legislative boards. Second, temporary shortages from a given plant configuration are usually met by outsourcing to other utility companies. Hence, introducing a penalty term that will minimize the levels of shortage across different scenarios will ease the arrangements between the collaborating utility companies, and also reduce the interperiod variability.

A single-period, deterministic, linear programming model for power system planning is given by:

$$\text{minimize} \sum_{i\in I} c_i x_i + \sum_{j\in J} \theta_j \sum_{i\in I} f_i y_{ij} \tag{17}$$

$$\text{subject to} \quad x_i - \sum_{j\in J} y_{ij} \geq 0, \quad \text{for all } i \in I, \tag{18}$$

$$\theta_j \sum_{i\in I} y_{ij} = d_j, \quad \text{for all } j \in J, \tag{19}$$

$$x_i \geq 0,\ y_{ij} \geq 0, \quad \text{for all } i \in I,\ j \in J. \tag{20}$$

Here $P_j$ and $\theta_j$ denote demand and duration for operating mode $j$; see Figure 4. Let $I$ denote the set of plant types (e.g., hydro, coal, etc.), $J$ is the set of operation modes (e.g., base, peak), and $c_i$ and $f_i$ are the annualized fixed cost (\$/MW) and operating cost (\$/MWh), respectively, for plant $i \in I$. The level of demand in (19) is obtained from the load duration curve by:

$$d_j = (P_j - P_{j-1})\theta_j.$$

Decision variables $x_i$ denote the total capacity installed at plant $i \in I$. These are the design variables. Variable $y_{ij}$ denotes the allocation of capacity from plant $i$ to supply operating mode $j$. For example, it determines what fraction of the capacity of a coal plant is used to supply peak load. The $y$'s are the control variables. The allocation of plant capacity to different modes of operation is determined after the plant capacities $(x_i)$ have been determined, and the demand levels for different modes $(d_j)$ have been observed.

The RO formulation of the power system planning model introduces a set of scenarios $s \in \Omega$ for the uncertain energy demands $\{d_j\}$. (Scenarios can be introduced for other forms of uncertainty, such as the fraction of plant capacity that will be available under each scenario due to equipment outages, etc.). The control variables are then scenario dependent and are denoted by $y_{ij}^s$. The linear programming model given above is reformulated in the following RO model:

$$\text{Min} \sum_{s\in\Omega} p_s \xi_s + \lambda \sum_{s\in\Omega} p_s\Big(\xi_s - \sum_{s'\in\Omega} p_{s'}\xi_{s'}\Big)^2$$

$$+ \omega \sum_{s\in\Omega} p_s\Big(\sum_{i\in I} (z_{1i}^s)^2 + \sum_{j\in J} (z_{2j}^s)^2\Big) \tag{21}$$

subject to:

$$x_i - \sum_{j\in J} y_{ij}^s = z_{1i}^s, \quad \text{for all } i \in I,\ s \in \Omega, \tag{22}$$

$$\theta_j \sum_{i\in I} y_{ij}^s + z_{2j}^s = d_j^s, \quad \text{for all } j \in J,\ s \in \Omega, \tag{23}$$

$$x_i \geq 0,\ y_{ij}^s \geq 0, \quad \text{for all } i \in I,\ j \in J,\ s \in \Omega. \tag{24}$$

The function $\xi_s$ is defined by

$$\xi_s = \sum_{i\in I} c_i x_i + \sum_{j\in J} \theta_j \sum_{i\in I} f_i y_{ij}^s.$$

The objective function of this RO formulation has three terms. The first term is the expected cost of the operation (in the traditional formulation of stochastic linear programs). The second term is the variance of the cost, weighted by the goal programming parameter $\lambda$. The third term penalizes a norm of the infeasibilities, weighted by parameter $\omega$. Table I summarizes comparative statistics between the solution of a stochastic programming formulation of this model, with the RO solution obtained for a particular setting of the parameters. (The parameters were set as $\lambda = 0.01$ and $\omega = 128$. Those values were determined as appropriate given the various tradeoffs between solution and model robustness analyzed in Malcolm and Zenios. Other parameters may be more appropriate depending on the goals of the decision makers.) Figure 5 illustrates the tradeoffs between the mean and variance of the solution for different values

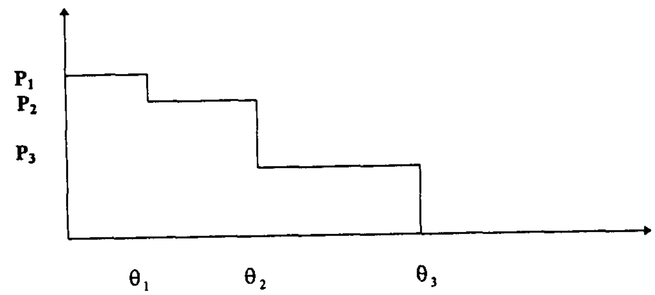
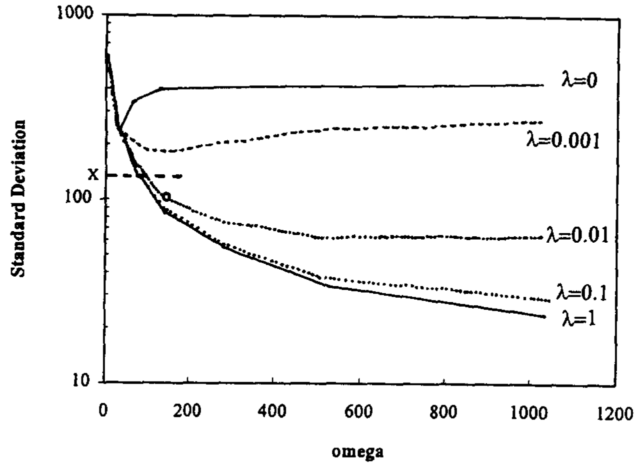



**Figure 4.** Piecewise linear load duration curve. The horizontal axis denotes duration for different levels of demand; these levels are indicated on the vertical axis.

**Table I**
Comparison of RO and Stochastic Linear Programming Solutions for the Power System Capacity Expansion Problem

| Scenario | Cost 1 | Cost 2 | Cost 3 | Cost 4 | Expected Cost | Variance of Cost | Excess Capacity |
|---|---|---|---|---|---|---|---|
| RO Model | 7,824 | 7,464 | 7,579 | 7,446 | 7,578 | 100 | 5.6 |
| Stochastic Programming | 7,560 | 7,320 | 7,620 | 7,380 | 7,470 | 124 | 7.3 |



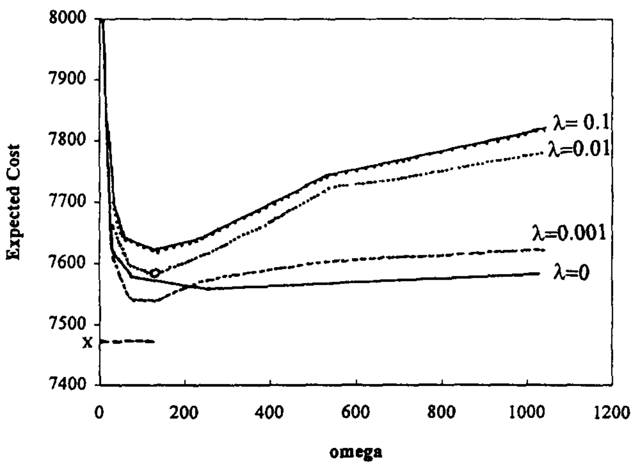




**Figure 5.** Tradeoff between the mean and variance of the solution in the RO formulation of the power capacity expansion model. The $x$ on the vertical axis indicates the solution of a stochastic linear programming formulation. The $o$ denotes the RO solution obtained for $\lambda = 0.01$ and $w = 128$. The RO solution has higher expected cost than the stochastic linear programming solution, but it has substantially lower standard deviation.

of $\lambda$. The mean and variance of the stochastic programming formulation of the same model are shown with an x on the same diagram.

### 2.2. Reconciliation of Data: The Matrix Balancing Problem

The problem of adjusting the entries of a large matrix of observed data to satisfy prior consistency requirements occurs in many diverse areas of application (see Schneider and Zenios 1990). A generic statement of this problem is as follows.

Given an $m \times n$ nonnegative matrix $A$ and positive vectors $u$ and $v$ of dimensions $m$ and $n$, respectively, determine a "nearby" nonnegative matrix $X = (x_{ij})$ (of the same dimensions) such that its entries satisfy a set of linear restrictions that are consistent with the observation vectors $u$ and $v$. Such restrictions take, for example, the form:

$$\sum_{j=1}^{n} x_{ij} = u_i, \tag{25}$$

$$\sum_{i=1}^{m} x_{ij} = v_j. \tag{26}$$

The following specific models fall under this framework: Estimation of migration patterns in regional planning, estimation of social accounting matrices for development planning, estimation of origin/destination matrices for transportation or telecommunications traffic, updating input/output tables for econometric modeling, reconciling census observations, and several others. Typically, the problem is one of adjusting the entries of the matrix such that row totals (i.e., the total income of a sector) are equal to observed values. Column totals (i.e., total expenditure of a sector) are also equal to observed values.

Under the requirement of biproportional adjustments[2] a suitable formulation of the matrix balancing problem is:

$$\underset{x \geq 0}{\text{minimize}} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \log \frac{x_{ij}}{a_{ij}} \tag{27}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij} = u_i, \quad \text{for } i = 1, 2, \ldots, m, \tag{28}$$

$$\sum_{i=1}^{m} x_{ij} = v_j, \quad \text{for } j = 1, 2, \ldots, n. \tag{29}$$

The observation vectors $u$ and $v$ are often subject to error. It is then possible that the problem of balancing the matrix $A$ has no solution. (Clearly, if $\sum_{i=1}^{m} u_i \neq \sum_{j=1}^{n} v_j$, then the optimization problem has no feasible solution.) Several suggestions appeared in the literature to overcome this problem. Folklore suggests that the vectors $u$ and $v$ are first scaled so that feasibility is ensured. Zenios, Drud and Mulvey (1989) suggested the updating of the vectors $u$ and $v$ based on a least-squares or entropy approximation. Jornsten and Wallace (1990) also suggested a least-squares update of the observation vectors. Censor and Zenios (1991) suggested that a range of permissible values be specified for the vectors $u$ and $v$, and the problem be solved as an interval-constrained entropy optimization model.

The RO formulation of matrix estimation problems can be written as:

$$\underset{x \geq 0}{\text{minimize}} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \ln\left(\frac{x_{ij}}{a_{ij}}\right) + \frac{\mu}{2}\left\{\sum_{i=1}^{m} y_i^2 + \sum_{j=1}^{n} z_j^2\right\} \tag{30}$$

subject to:

$$\sum_{j=1}^{n} x_{ij} - y_i = u_i, \quad \text{for } i = 1, 2, \dots, m, \tag{31}$$

$$\sum_{i=1}^{m} x_{ij} - z_j = v_j, \quad \text{for } j = 1, 2, \dots, n. \tag{32}$$

This formulation, derived here as a direct application of the RO framework, can be derived from statistical arguments (Elfving 1989, Zenios and Zenios 1992). The entropy term estimates the matrix which is the least biased (or maximally uncommitted) with respect to missing information, conditioned on the observations $\{a_{ij}\}$. An axiomatic characterization of entropy maximization solutions for matrix balancing problems is due to Balinski and Demange (1989). The quadratic terms are the logarithms of the probability distribution function of the error (i.e., noise) term, conditioned on the matrix $\{x_{ij}\}$. Here it is assumed that the errors are normally distributed with mean zero and standard deviations that are identical for all observations. The model maximizes (the logarithm) of the probability of the matrix $\{x_{ij}\}$, conditional on the noisy observations $\{u_i, v_j\}$, assuming a uniform prior distribution of $x_{ij}$, and normally distributed observation vectors. This is a Bayesian estimate of the matrix. The arguments that lead to this formulation can be found in Elfving for the problem of medical imaging, and Zenios and Zenios for the matrix balancing problem.

This RO model is a mixed entropy/quadratic optimization problem subject to network flow (transportation) constraints. For small sized problems it can be solved with off-the-shelf nonlinear optimizers, like MINOS. For medium to large sized problems it can be solved using state-of-the-art nonlinear network optimization codes, such as GENOS (Mulvey and Zenios 1987). As matrix balancing applications tend to be of very large size (1,000 × 1,000 matrices with several hundred thousand entries to be estimated are common), there is a tendency in the literature to develop special purpose algorithms for these problems. The references mentioned above (Zenios, Drud and Mulvey 1989, Elfving 1989, Schneider and Zenios 1990, Censor and Zenios 1991) develop specialized algorithms for different formulations of the problem.

A primal/dual row-action algorithm for the RO formulation of the matrix estimation problem is given in Zenios and Zenios. The algorithm was used to solve problems of size 1,000 × 1,000 with up to 800,000 coefficients within 3–5 minutes on an Alliant FX/8. The algorithm is also well suited for vector and parallel implementations. When these architectural features of the Alliant FX/8 were properly exploited the test problems were solved within 10–15 seconds.

#### 2.2.1. Image Reconstruction

A problem closely related to matrix balancing is that of image reconstruction. The reconstruction of images from partial and, perhaps, incomplete observations appears in numerous areas of application. In medical imaging problems, for example, one has to reconstruct an image of a cross-section of the human body by observing the absorption of x-rays along different views (i.e., directions) of the cross-section. In emission computerized tomography, images are reconstructed by observing the emissions of positrons from material that has been injected in the blood stream.

Similar techniques are applied in nondestructive material testing: Identify imperfections in a structure by observing the deflection of some radiation pattern, as opposed to subjecting the structure to stress or strain tests. In seismic data analysis researchers try to understand the subsurface earth structure by observing the deflection of seismic waves. These, and several other applications, are discussed in the book by Herman (1980). Matrix balancing can be viewed as a problem of image reconstruction from two orthogonal projections.

A typical approach for image reconstruction is to discretize the cross-section of the image, and assume that its density is uniform within each picture element (i.e., pixel) and given by a vector $x$. The geometry of the cross section and the directions along which the image was observed, together with the numerical values of the observations, specifies a set of linear restrictions of the form $Ax = b$. Here $x_j$ denotes the density of the $j$th pixel, $a_{ij}$ is the length of intersection of the $j$th pixel by the $i$th x-ray, and $b_i$ denotes the observed intensity of the $i$th ray. Statistical or physical considerations, depending on the application, lead to an entropy optimization formulation of the form:

$$\underset{x \geq 0}{\text{minimize}} \; -\text{ent}(x) = \sum_j x_j \log \frac{x_j}{m_j} \tag{33}$$

$$\text{subject to} \quad Ax = b. \tag{34}$$

Here, $m_j$ denotes an a priori estimate for the density of the $j$th pixel.

While the matrix $A$ is usually well specified from the geometry of the problem, the vector of observations $b$ is noisy. For example, in emission computerized tomography, $b$ is a Poisson random variable with an unknown mean $\beta$. A popular approximation used to model noisy $b$'s is to assume that the deviation from the mean is normally distributed with mean zero and some known standard deviation; i.e., $b := \beta + r$, where $r$ is a mean zero noise vector. Detailed justification for this model of medical image reconstruction from noisy data is given in Elfving; see also Herman (1980), Minerbo (1981), and Smith and Grandy (1985).

This problem can be formulated as a special case of robust optimization: It consists only of design variables $x$ and the matrices $\{B_1, B_2, \dots, B_S\}$ are identical. Only the right-hand side vectors $\{e_1, e_2, \dots, e_S\}$ are subject to noise. The model is written as:

$$\underset{x \geq 0}{\text{minimize}} \; -\text{ent}(x) + \frac{\mu}{2} z^{\top} z \tag{35}$$

subject to $z = Ax - b$. (36)

The parameter $\mu > 0$ reflects the relative importance between the data term (i.e., minimum entropy solution) and the noise-smoothing term. In our terminology, the entropy term measures solution robustness, while the quadratic term measures model robustness. This formulation for image reconstruction problems was first proposed by Elfving. He also proposed the use of Wahba's (1977), cross-validation technique for picking $\mu$. This formulation has been used in several image reconstruction problems. See O'Sullivan and Wahba (1985) for applications in remote sensing experiments, Elfving for x-ray medical imaging, and Herman et al. for emission computerized tomography.

### 2.3. Airline Allocation for the Air Force: The STORM Model

This example involves the scheduling of aircraft to routes for U.S. military operations worldwide. The responsibility for these decisions lies with the Air Mobility Command (AMC), previously the Military Airlift Command or MAC. A large deterministic linear program is used to make these assignments to minimize total costs to supply the aircraft. This system played an important role in operations such as the 1991 Iraqi conflict and the 1993 incidents in Somalia.

Each month the various services (Army, Navy, Air Force, and Marines) send to AMC their projections for the amount of cargo to be sent across various routes worldwide, for instance, from California to Saudi Arabia. These estimates then become point forecasts for the number of aircraft to be supplied over the military transportation network. Unfortunately, at least for the Air Force planners, the estimates must be modified as conditions change during the month as the services adjust their needs. As an additional condition, the load of an aircraft may be event-dependent provided that it does not exceed its maximum capacity. The military sorties, however, must be determined before the actual demands will be known. At times, demand can be handled by leasing aircraft from commercial airlines.

There are several alternative model formulations for STORM, depending upon the needs of the Air Force planners. The most compact and commonly used involves decision variables representing the overall number of flights by each aircraft type over the transportation network. This leads to the following category of decision variables:

$x$ - the number of sorties on each route in the network (by aircraft type);

$v$ - the amount of undelivered cargo using military aircraft.

The deterministic version of the aircraft allocation problem is defined as follows.

**Problem STORM**

$$\text{minimize } f(x, v) \tag{37}$$

$$\text{subject to} \quad Ax = b, \tag{38}$$

$$Bx + Cv = e, \tag{39}$$

$$x, v \geq 0. \tag{40}$$

Simply speaking, the model minimizes total costs $f(\cdot)$, while satisfying a variety of flow balance (structural), and other constraints (structural and control). For instance, there must be a balance of landings and takeoffs, the cargo must fit within the capacity of the air type, and the total aircraft flying hours must stay within the design specifications. Cargo can be moved directly from location to location or via transshipment links in the network.

The model was designed to meet the cargo and other transportation needs of the services by employing military planes, or by leasing space or equipment from commercial carriers. The later strategy is generally more expensive and avoided if possible. The basic model is a deterministic linear program of the form shown above. All model coefficients represent the best estimates of the planners.

In 1991, AMC became interested in robust optimization for several reasons. First, the deep reductions in overall force size led to a large reduction in the need for **STORM** capacity. In the past, the Air Force provided a level of service that would meet or exceed most demands—a worst-case plan. The cost of providing this service, however, grew as the cost of excess capacity (overage) became recognized. Both overage and underage costs are included in the RO model. Next, the planners were searching for a systematic method for reducing the variability of the solution from month to month. Each scheduling change requires a cascading set of operational modifications. RO provides the approach for smoothing out the changes over time by using a conservative, risk averse utility function.

To accomplish the transformation of the **STORM** model, we developed a forecasting system for projecting traffic based on scenarios. As before, a scenario provides a single plausible and coherent set of parameters for **STORM**. Letting $\Omega$ represent the set of scenarios we expand the basic model as follows.

**Problem RO-STORM**

$$\text{Minimize } \sum_{s=1}^{S} p_s U[f(x, v) + \rho(c_0(v)\delta^+(v) + c_u(v)\delta^-(v))]$$

$$\text{subject to} \quad Ax = b,$$

$$B_s x + Cv_s = e_s + \delta^+(v) - \delta^-(v), \quad \text{for all } s \in \Omega,$$

$$x, \delta^+(v), \delta^-(v), v_s \geq 0, \quad \text{for all } s \in \Omega,$$

where $c_0(\nu)$ and $c_u(\nu)$ depict the costs for overage and underage, $U[\cdot]$ is the von Neumann-Morgenstern utility function, and $\delta^+(\nu)$ and $\delta^-(\nu)$ are the deviations for violations of the control constraints. The overage and underage costs are determined by the military planners based on their priorities. The utility function is chosen in a way to reduce the variations in the schedule over time. A highly risk averse function displays much less variation, especially with regard to the military planes. The actual degree of robustness accepted by the planners is a decision that is based on the efficient frontier of expected costs versus the anticipated number of likely scheduling changes. An interactive procedure is under development for assistance with these tradeoff decisions.

The ideal solution from the standpoint of model robustness is to identify groups of aircraft flights that are constant across time. This group of fights is called the *core*. Variables outside of the core are allowed to change as demand for cargo services increases or decreases. The robust models are used in conjunction with a routine which selects the core variables. Additional constraints are imposed on total transportation costs during these runs. Our tests show that the variability of costs can be reduced by over 40% on a monthly basis with only a 5% increase in expected cost.

The size of a typical transcontinental version of the deterministic formulation of **STORM** (e.g., U.S. and Europe) is approximately 585 rows, 1,380 variables, and 3,799 nonzero coefficients. Thus, by adding multiple scenarios, we generate large optimization problems, as shown in Table II. In our computational tests, the quadratic interior point code LOQO (Vanderbei 1992) has been able to solve up to 20 scenarios of **STORM** on an SGI Indigo workstation. Results are shown in the same table. Much larger problems can be solved by employing distributed and parallel computers, as discussed in Section 3; see, e.g., Berger, Mulvey and Ruszczyński (1994), and Jessup, Yang and Zenios (1994).

### 2.4. Scenario Immunization

Dembo (1992) introduced the notion of *scenario immunization* for the management of fixed income portfolios under uncertain interest rate scenarios. A portfolio is termed *immunized* if the present value of the assets in the portfolio matches the present value of the liabilities that the portfolio is expected to fund. However, a difficulty is encountered when deciding what interest rates to use for discounting both assets and liabilities to their present value. If interest rates are fixed, and known a priori, then by matching present values of both sides of the balance sheet we are guaranteed that assets and liabilities will grow in the same way, and the liabilities will be fully funded. Hence, assuming some scenario $s$ for discount rates we can write the portfolio immunization problem as:

$$\underset{x_s \geq 0}{\text{Minimize}}\ c_s^{\top} x_s \tag{41}$$

$$\text{subject to} \quad \sum_{j \in J} V_{sj} x_{sj} = V_{sL}. \tag{42}$$

We use $J$ to denote the set of available instruments with market price vector $c_s$ and present value $V_s = (V_{sj})$. The present value of the liabilities under the assumed scenario is $V_{sL}$, and $x_s$ denotes the composition of the optimized portfolio. Let the optimal value of this problem be $\nu_s$. Dembo proposed to solve the portfolio immunization when the scenarios $s$ takes values from a set $\Omega$ by solving the following *tracking* problem:

$$\underset{x \geq 0}{\text{minimize}} \sum_{s \in \Omega} p_s \left[ (c_s^{\top} x - \nu_s)^2 + \left( \sum_{j \in J} (V_{sj} x_j - V_{sL})^2 \right) \right]. \tag{43}$$

This model is a special case of the robust optimization framework when only design variables and structural constraints are present.[3] The second term of the objective function corresponds to a quadratic penalty function on feasibility robustness. (Just define $z_s = \sum_{j \in J} V_{sj} x_j - V_{sL}$ in the scenario immunization model and the relation with the robust optimization model (5)–(8) follows.) The first term of the objective function is a quadratic penalty function for feasibility robustness of the constraints:

$$\sum_{j \in J} c_{sj} x_j = \nu_s \quad \text{for all } s \in \Omega.$$

These constraints are imposed to enforce optimality robustness. The optimal robust solution $x^*$ will have an objective that remains close to the optimal value $\nu_s$ for any one of the realized scenarios.

**Table II**
Size of the Robust Optimization Problem as a Function of the Number of **STORM** Scenarios and Solution Times Using LOQO

| Scenarios | Rows | Columns | Nonzeros | CPU(s) | Elapsed(s) |
|---|---|---|---|---|---|
| 1 | 585 | 1,380 | 3,799 | 3.2 | 4 |
| 3 | 1,755 | 4,140 | 11,397 | 27.4 | 31 |
| 5 | 3,530 | 6,900 | 20,205 | 84.1 | 91 |
| 8 | 4,680 | 11,040 | 30,392 | 270.6 | 282 |
| 10 | 5,850 | 13,800 | 37,900 | 491.4 | 519 |
| 12 | 7,020 | 16,560 | 45,588 | 909.6 | 929 |
| 20 | 11,700 | 27,600 | 75,980 | 3815.7 | 3935 |

### 2.5. Minimum Weight Structural Design

This subsection describes an example of robust optimization in which the selection of scenarios is carefully done to cover the largest number of events possible. This topic will become important as applications become more common.

The minimum weight structural design problem can be described as follows. Given a collection of admissible joints (some of which are designated as anchor joints) and a corresponding collection of admissible members (i.e., beams) connecting pairs of joints, find the cross-sectional area of each member so that the resulting structure has minimal weight and can accommodate a given load at each joint. The constraint that the design be capable of supporting a given load can be expressed as a set of linear equations that must be satisfied:

$$Af = -\lambda, \tag{44}$$

where each row of this system of equations represents the balancing of either an $x$ or a $y$ component of force at one of the nonanchored joints (assuming the problem is formulated in two dimensions). Each component of the vector $\lambda$ appearing on the right-hand side contains a corresponding $x$ or $y$ component of the given load and the vector $f$ represents the forces in each admissible member (positive forces represent tension and negative forces represent compression). The optimization criterion is to choose the design having minimal weight. The weight of a member is proportional to its length times its cross-sectional area and we assume that the cross-sectional area is proportional to the amount of force in the member. Hence, the total weight of the structure is proportional to

$$\sum_j l_j |f_j|, \tag{45}$$

where $l_j$ denotes the length of member $j$. Assuming that the proportionality constant is equal to one, the problem is to minimize (45) subject to (44).

To convert this problem to an equivalent linear programming problem each variable $f_j$ is replaced by the difference between its positive and its negative parts $f_j = f_j^+ - f_j^-$. Then the problem can be rewritten as

$$\text{minimize } l^T f^+ + l^T f^- \tag{46}$$

$$\text{subject to } Af^+ - Af^- = -\lambda, \tag{47}$$

$$f^+ \geq 0, f^- \geq 0. \tag{48}$$

Figure 6 shows an example of a structure designed in this manner. See Ho (1975) for a further discussion of the optimal design problem. In this example, the two nodes on the bottom level are the anchor nodes. Most of the other nodes have no applied loads. The only exceptions are the three nodes on the top row, the three inner nodes two rows down from the top, and the three inner nodes four rows down from the top. At these nine nodes there is a vertical load representing the weight due to gravity of an external load applied at these points. Hence, the structure is a tower that must support weight on three different levels.

There are drawbacks to the optimal structural design problem as formulated above. First, structures must always be designed to accommodate a variety of load scenarios. For example, the effects of wind must be considered. However, wind is variable. One could consider applying just a wind of a fixed intensity from a fixed direction, say the left. Figure 7 illustrates the optimal structure obtained from such a model with wind included. Suddenly the optimal design is entirely unreasonable because part of the structure is standing on a point. Given the model, this structure is correct since the wind is considered to be constant and so it exactly counterbalances the tendency for the leaning portion of the structure to tip over. (Even using only vertical loads, it is possible to obtain optimal structures that are similarly unreasonable.) To remedy this situation we must consider at least two scenarios, one with a fixed wind coming from the left and another with a fixed wind coming from the right. By choosing a reasonably large fixed wind velocity, it is clear that these two scenarios yield a structure that can withstand a wide range of wind directions and intensities.

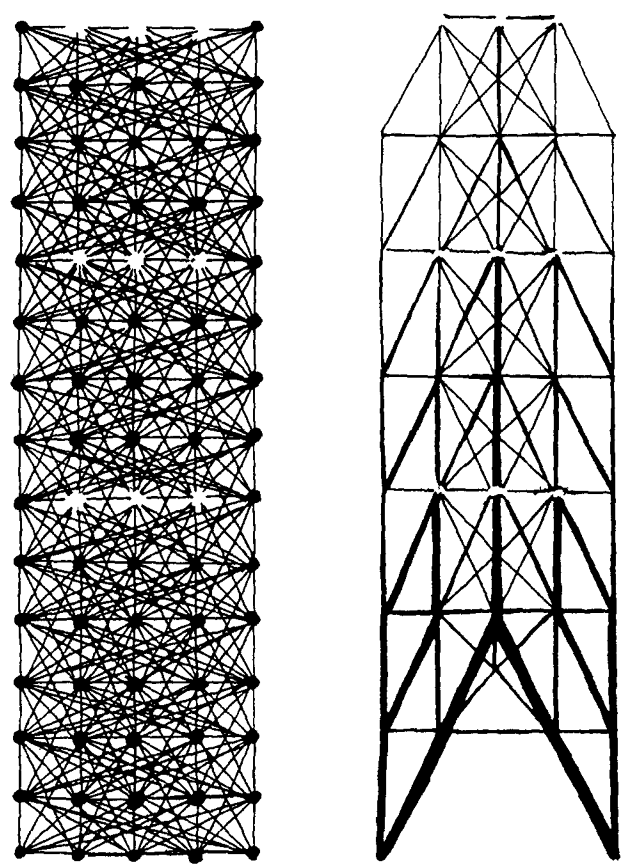


**Figure 6.** An optimal structural design based on a single scenario. (The figure on the left illustrates all admissible joints from which the joints of the right figure, and their cross sections, have been selected.)

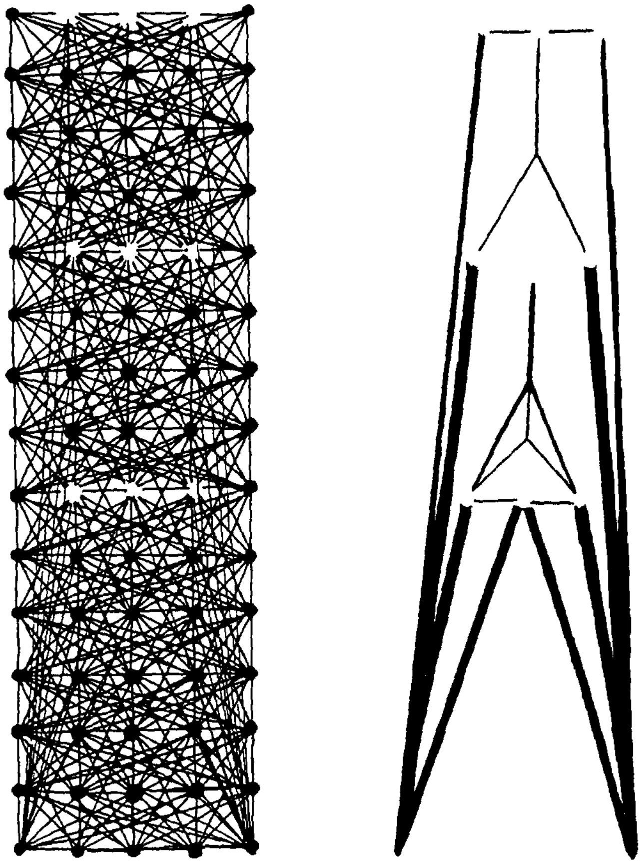


**Figure 7.** Difficulty of optimizing a structural design over a single scenario. (The figure on the left illustrates all admissible joints from which the joints of the right figure, and their cross sections, have been selected.)

Hence, for the optimal structural design problem, we are naturally led to consider a "robust" version. Indeed, suppose that there are two load scenarios $\lambda^1$ and $\lambda^2$. If we let $f^k$ denote the forces in the members under the $k$th scenario and let $\phi$ denote the vector of cross-sectional areas of the members, then the robust formulation can be written as:

minimize $l^{\top}\phi$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \end{bmatrix} \begin{bmatrix} f^1 \\ f^2 \\ \phi \end{bmatrix} = -\begin{bmatrix} \lambda^1 \\ \lambda^2 \end{bmatrix}$$

$$\begin{bmatrix} -I & 0 & I \\ I & 0 & I \\ 0 & -I & I \\ 0 & I & I \end{bmatrix} \begin{bmatrix} f^1 \\ f^2 \\ \phi \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\phi \geq 0.$$

The first two sets of nonnegativity constraints simply say that $|f^1| \leq \phi$, and the second two sets say that $|f^2| \leq \phi$.

One might argue that we have only considered two scenarios and that the varieties in real life are much broader. However, we observe that any scenario that is a subconvex combination of the given scenarios is automatically covered. To see this, suppose that a new load scenario $\sigma$ is given as a subconvex combination of $\lambda^1$ and $\lambda^2$:

$$\sigma = p\lambda^1 + q\lambda^2,$$

where $p$ and $q$ are nonnegative real numbers satisfying $p + q \leq 1$. Let $f = pf^1 + qf^2$. Then

$$Af = pAf^1 + qQf^2 = -\sigma,$$

and

$$|f| = |pf^1 + qf^2| \leq \max(|f^1|, |f^2|) \leq \phi.$$

Hence, $\phi$ is big enough to handle the load given by $\sigma$. Therefore, for problems in two dimensions, two scenarios can cover a pretty wide set of scenarios, whereas in three dimensions one would probably use four scenarios. Figure 8 illustrates the design obtained using an RO formulation of the problem.

### 2.6. Applications to NETLIB Models

The robust optimization concepts can be applied to a general linear program in which very small perturbations are made to the data coefficients. In effect, this approach corresponds to a formal method for minimizing the solution perturbations in the context of forward error analysis.

To demonstrate the ideas, we set up tests with selected NETLIB linear programs. A generator was designed to modify the test problems in a very small way, specifically the coefficients: ($A_i$, $b_i$, and $c_i$). Random perturbations in these coefficients were made using a uniform density function with range equal to $\pm\kappa$. (The structural elements—all coefficients with 0, +1, and −1—remained unchanged). Similar ranges were established for the $b_i$ and the $c_j$ coefficients.

The combined objective function ((expected value) − $1/\alpha$ (variance)) was solved for various values of $\alpha$, as depicted in Figure 9. This example is for the **AFIRO** problem using ten scenarios. The resulting efficient frontier is also shown in this figure. This experiment illustrates how the RO formulation can reduce the variance of the solution at the expense of its expected value.

The solution time of the RO model increases with the number of scenarios. However, a small number of scenarios could be solved with LOQO (Vanderbei) in solution times comparable to those required to solve the linear programming model. As the number of scenarios becomes large the solution times grow substantially. For example, **AFIRO** was solved in 0.3 seconds as a linear programming model, in 3.0 seconds as a robust optimization model with 10 scenarios, in 28.5 seconds with 40 scenarios, and in 16,200 seconds with 300 scenarios.

## 3. DATA PARALLEL COMPUTATION OF ROBUST SOLUTIONS

We turn now our attention to the solution of robust optimization problems. For a large number of scenarios we need to design special purpose algorithms. Such algorithms have been designed for general nonlinear programs (Mulvey and Ruszczyński 1992, Berger, Mulvey and Ruszczyński 1994, Jessup, Yang and Zenios 1994) and for network problems (Mulvey and Vladimirou 1991 and Nielsen and Zenios 1993a, b).

When applying algorithms such as those developed in the above references one can exploit the novel architectures of high-performance computers. One of the primary motivating factors for the development of the RO framework has been the recent developments with parallel and distributed computing.

Within the context of parallel computing the notion of *data-level parallelism* is particularly suitable for solving robust optimization problems. This model of programming parallel machines, introduced in Hillis (1987), postulates that parallelism is achieved by operating simultaneously, and with homogeneous operations, on multiple copies of the problem data. This form of parallelism has been embodied in the Connection Machine CM-2 and other SIMD architectures (MassPar and Active Memory Technologies DAP), and, more recently, in the Connection Machine CM-5, which is based

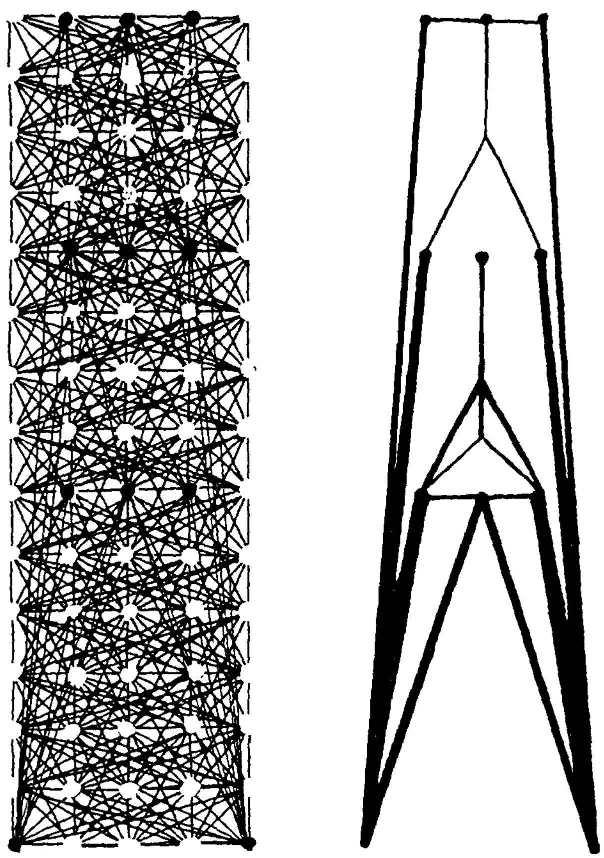


**Figure 8.** Improved structural design with multiple scenarios. (The figure on the left illustrates all admissible joints from which the joints of the right figure, and their cross sections, have been selected.)

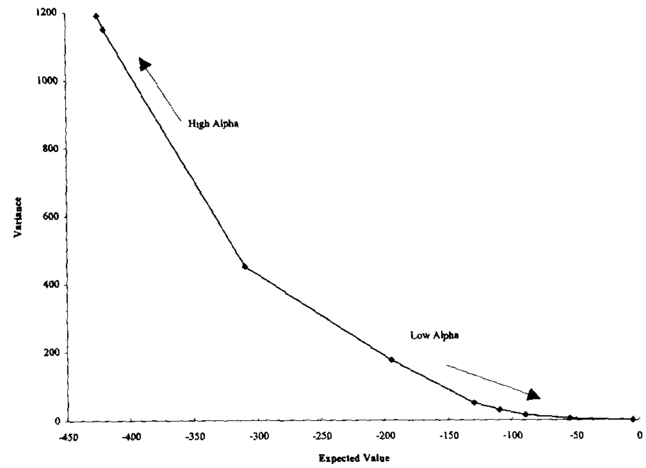



**Figure 9.** Robust efficient frontier for **AFIRO**.

on an MIMD architecture. The robust optimization problems have a natural mapping to data-level parallel architectures. The optimization problem remains structurally unchanged for different realizations of the scenarios. Only (some of) the problem data change. Hence, with the suitable use of decomposition algorithms we have to solve a series of problems that are structurally identical, while they differ in some of the coefficients. This is precisely the mode of operation of data-level parallel architectures. A survey of data-level parallel algorithms for large-scale optimization is given in Zenios (1994).

Another important development in computer architectures involves the notion of *distributed heterogeneous* computing, whereby a group of computers are linked via high-speed communications for the purpose of solving problems which are too large for any single machine. This concept has gotten attention due to the recent improvements in telecommunication capabilities. Berger, Mulvey and Ruszczyński (1994) describe a decomposition method which does not need a master controller for coordinating the informational flows, and is therefore well suited for distributed architectures.

## 4. CONCLUSIONS

This paper argues that robustness ought to be included in the development of mathematical programming models. It is extremely rare that real-world data is known within machine precision. Thus, noisy data is a fact of life and should be handled directly, rather than as part of *ex poste* sensitivity analysis. (Situations in which parametric analysis indicates that the results are relatively insensitive to data perturbations are automatically taken care of.)

We have developed a general framework for achieving robustness. We have discussed the relative merits of RO over sensitivity analysis and stochastic programming. We have also seen how RO models would indeed generate robust solutions for several applications. The RO framework also embodies as special cases several other approaches that have been proposed in the past for handling noisy and incomplete data.

To make robustness a part of an optimization model requires high performance computers. We must be able to solve hundreds or even thousands of optimization programs simultaneously with some degree of coordination. Sequential computers are unable to handle this task except for a small number of scenarios. We have pointed out some promising directions for future work in the area of parallel and distributed algorithms.

Several open issues deserve further investigation. One is the issue of designing and implementing suitable decomposition algorithms. Another is the development of modeling environments where problems with noisy data can be specified easily. Existing modeling languages, like GAMS or AMPL, lack such capabilities, but they could be extended. Finally, additional work is needed in specifying effective procedures for selecting scenarios and specifying the multiobjective programming weights. Interactive and visual-based systems should help in this regard. These issues arise in other applications of mathematical programming as well. We expect that the topic of robust optimization will receive increasing attention as its importance is realized.

## NOTES

1. This model, and its implications, are admittedly artificial. However, it was brought to our attention at the 1991 TIMS/ORSA National Meeting (Roush et al. 1992) that controlling the contents of the diet for animal feeding is a problem of practice significance: Corn delivered to cattle from different parts of a field will have different nutritional properties. Deviations of the animals' diet from preset targets may have adverse effects on the quality and quantity of produced milk.
2. Biproportional adjustments (Bacharach 1970) require that entries of the matrix A should be adjusted to obtain the balanced matrix X in such a way that the adjustments are proportional to the magnitude of the entries. That is, larger values of $a_{ij}$ are adjusted more than smaller values.
3. This is, indeed, the major limitation of the scenario immunization model: It ignores the possibility of rebalancing the portfolio which could be captured by the use of controls variables. Nevertheless, the model is easy to solve and has been proven successful in several applications.

## ACKNOWLEDGMENT

The research of S. A. Zenios was funded in part by NSF grants CCR-91-04042 and SES-91-00216 and AFOSR grant 91-0168, while he was with the University of Pennsylvania. Computing resources were made available through Thinking Machines Corporation and the Army High-Performance Computing Research Center (AH-PCRC) at the University of Minnesota. The research of J. M. Mulvey was funded in part by NSF grant CCR-9102660 and Air Force grant AFOSR-91-0359. We would like to acknowledge the assistance of Scott Malcolm with the power capacity planning model and of Stefanos Zenios with the matrix balancing model. The comments of the referees and the associate editor led to substantial improvements in the organization of the paper.

## REFERENCES

BACHARACH, M. 1970. *Bi-proportional Matrices and Input-Output Change*. Cambridge University Press, U.K.

BALINSKI, M. L., AND G. DEMANGE. 1989. An Axiomatic Approach to Proportionality Between Matrices. *Math. Opns. Res.* **14,** 700–719.

BEALE, E. M. L. 1955. On Minimizing a Convex Function Subject to Linear Inequalities. *J. Royal Stat. Soc.* **17,** 173–184.

BERGER, A. J., AND J. M. MULVEY. 1994. Errors is Asset Management. SOR Report 94-8, Statistics and Operations Research, Princeton University, Princeton, N.J.

BERGER, A. J., J. M. MULVEY AND A. RUSZCZYŃSKI. 1994. A Distributed Scenario Decomposition Algorithm for Large Scale Stochastic Programs. *SIAM J. Optim.* (to appear).

BERTSEKAS, D. P. 1982. *Constrained Optimization and Lagrange Multipliers Method*. Academic Press, New York.

BIRGE, J. R. 1982. The Value of the Stochastic Solution in Stochastic Linear Programs With Fixed Recourse. *Math. Prog.* **24,** 314–325.

BROOKE, A., D. KENDRICK AND A. MEERAUS. 1992. *GAMS: A User's Guide, Release 2.25*. The Scientific Press.

CARINO, D. R., T. KENT, D. H. MYERS, C. STACY, M. SYLVANUS, A. L. TURNER, K. WATANABE AND W. T. ZIEMBA. 1994. The Russell-Yasuda Kasai Model: An Asset/Liability Model for a Japanese Insurance Company Using Multistage Stochastic Programming. *Interfaces* **24,**(1), 29–49.

CENSOR, Y., AND S. A. ZENIOS. 1991. Interval-Constrained Matrix Balancing. *Lin. Alg. and Its Applic.* **150,** 393–421.

DANTZIG, G. B. 1955. Linear Programming Under Uncertainty. *Mgmt. Sci.* **1,** 197–206.

DANTZIG, G. B. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton, N.J.

DANTZIG, B. B. 1990. The Diet Problem. *Interfaces* **20,** 43–47.

DANTZIG, G. B., AND G. INFANGER. 1991. Large-Scale Stochastic Linear Programs: Importance Sampling and Benders Decomposition. Report 91-4, Department of Operations Research, Stanford University, Stanford, Calif.

DEMBO, R. S. 1992. Scenario Immunization. In *Financial Optimization*, S. A. Zenios (ed.). Cambridge University Press, Cambridge, U.K., 290–308.

DUPACOVA, J. 1987. Stochastic Programming With Incomplete Information: A Survey of Results on Postoptimization and Sensitivity Analysis. *Optim.* **18,** 507–532.

Dupacova, J. 1990. Stability and Sensitivity Analysis for Stochastic Programming. *Anns. Opns. Res.* **27,** 115–142.

Elfving, T. 1989. An Algorithm for Maximum Entropy Image Reconstruction From Noisy Data. *Math. and Comput. Model.* **12,** 729–745.

Ermoliev, Yu., and R. J.-B. Wets. 1988. Stochastic Programming, An Introduction. In *Numerical Techniques for Stochastic Optimization*, Yu. Ermoliev and R. J. B. Wets (eds.). Springer-Verlag, Berlin, 1–32.

Escudero, L. F., P. V. Kamesam, A. King and R. J.-B. Wets. 1993. Production Planning Via Scenario Modeling. *Anns. Opns. Res.* **43,** 311–335.

Dantzig, G. B., et al. 1989. Decomposition Techniques for Multi-Area Generation and Transmission Planning Under Uncertainty. Report El-6484, EPRI, Palo Alto, Calif.

Glynn, P. W., and D. L. Iglehart. 1989. Importance Sampling for Stochastic Simulations. *Mgmt. Sci.* **35,** 1367–1392.

Golub, B., M. Holmer, R. McKendall, L. Pohlman and S. A. Zenios. 1994. Stochastic Programming Models for Money Management. *Eur. J. Opnl. Res.* (to appear).

Gutierrez, G. J., and P. Kouvelis. 1995. A Robustness Approach to International Outsourcing. *Anns. Opns. Res.* (to appear).

Herman, G. T. 1980. *Image Reconstruction From Projections: The Fundamentals of Computerized Tomography*. Academic Press, New York.

Herman, G.T., D. Odhner, K. Toennies and S. A. Zenios. 1990. A Parallelized Algorithm for Image Reconstruction From Noisy Projections. In *Large Scale Numerical Optimization*, T. Coleman and Y. Li (eds.). SIAM, Philadelphia, 3–21.

Hiller, R. S., and C. Schaack. 1990. A Classification of Structured Bond Portfolio Modeling Techniques. *J. Portfolio Mgmt. Fall*, 37–48.

Hillis, W. D. 1975. The Connection Machine. *Sci. Am.*, June.

Ho, J. K. 1975. Optimal Design of Multistage Structures: A Nested Decomposition Approach. *Comput. and Struct.* **5,** 249–255.

Infanger, G. 1992. Monte Carlo (Importance) Sampling With a Benders Decomposition Algorithm for Stochastic Linear Programs. *Anns. Opns. Res.* **39,** 69–95.

Jessup, E. R., D. Yang and S. A. Zenios. 1994. Parallel Factorization of Structured Matrices Arising in Stochastic Programming. *SIAM J. Optim.* **4,** 833–846.

Jornsten, K., and S. Wallace. 1990. Overcoming the Problem of (Apparent) Inconsistency in Estimating O/D Matrices. Working Paper, Norwegian Institute of Technology, Trondheim, Norway.

Keeney, R. L., and H. Raiffa. 1976. *Decisions With Multiple Objectives*. John Wiley, New York.

Levitan, E., and G. T. Herman. 1987. A Maximum a Posteriori Probability Expectation Maximization Algorithm for Image Reconstruction in Emission Tomography. *IEEE Trans. Med. Imag.* **6,** 185–192.

Malcolm, S., and S. A. Zenios. 1994. Robust Optimization for Power Capacity Expansion Planning. *J. Opnl. Res. Soc,* **45,** 1040–1049.

Markowitz, H. 1959. *Portfolio Selection, Efficiency Diversification of Investments*. Cowles Foundation Monograph 16, Yale University Press, New Haven, Conn. (second edition, Basil Blackwell, Cambridge, 1991).

Minerbo, G. 1981. Ment: A Maximum Entropy Algorithm for Reconstructing a Source From Projection Data. *Comput. Graph. and Image Proc.* **10,** 48–68.

Morgenstern, O. 1963. *On the Accuracy of Economic Observations*. Princeton University Press, Princeton, New Jersey.

Mulvey, J. M., and A. Ruszczyński. 1992. A Diagonal Quadratic Approximation Method for Large Scale Linear Programs. *O.R. Letts.* **12,** 205–215.

Mulvey, J. M., and H. Valdimirou. 1991. Solving Multistage Stochastic Networks: An Application of Scenario Aggregation. *Networks* **21,** 619–643.

Mulvey, J. M., and S. A. Zenios. 1987. GENOS 1.0: A Generalized Network Optimization System. User's Guide. Report 87-12-03, Decision Sciences Department, University of Pennsylvania, Philadelphia, Penn.

Murphy, F. H., S. Sen and A. L. Soyster. 1982. Electric Utility Capacity Expansion Planning With Uncertain Load Forecasts. *IIE Trans.* **14**(1), 52–59.

Nielsen, S., and S. A. Zenios. 1993a. A Massively Parallel Algorithm for Nonlinear Stochastic Network Problems. *Opns. Res.* **41,** 319–337.

Nielsen, S. S., and S. A. Zenios. 1993b. Proximal Minimizations With $D$-Functions and the Massively Parallel Solution of Linear Stochastic Network Programs. *Int. J. Supercomput. Applic.* **7**(4), 349–364.

O'Sullivan, F., and G. Wahba. 1985. A Cross Validated Bayesean Retrieval Algorithm for Nonlinear Remote Sensing Experiments. *J. Comput. Phys.* **59,** 441–455.

Paraskevopoulos, D., E. Karakitsos and B. Rustem. 1991. Robust Capacity Planning Under Uncertainty. *Mgmt. Sci.* **37,** 787–800.

Pinar, M. C., and S. A. Zenios. 1992. Parallel Decomposition of Multicommodity Network Flows Using Linear-Quadratic Penalty Functions. *ORSA J. Comput.* **4,** 235–249.

Prékopa, A. 1980. Network Planning Using Two-Stage Programming Under Uncertainty. In *Recent Results in Stochastic Programming*, P. Kall and A. Prékopa (eds.). Number 179 in Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, 215–237.

Roush, W. B., R. H. Stock, T. L. Cravener and T. H. D'Alfonso. 1992. Stochastic Nonlinear Programming for Formulating Commercial Animal Feed. Technical Report, Poultry Science Department, Pennsylvania State University, University Park, Penn.

Sanghvi, A. P., and I. H. Shavel. 1986. Investment Planning for Hydrothermal Power System Expansion: Stochastic Programming Employing the Dantzig-Wolfe Decomposition. *IEEE Trans. Power Syst.* **1**(2), 115–121.

Schneider, M. H., and S. A. Zenios. 1990. A Comparative Study of Algorithms for Matrix Balancing. *Opns. Res.* **38,** 439–455.

Seider, W. D., D. D. Brengel and S. Widagdo. 1991. Nonlinear Analysis in Process Design. *AIChE J.* **37**(1), 1–38.

Sengupta, J. K. 1991. Robust Solutions in Stochastic Linear Programming. *JORSA* **42**(10), 857–870.

SHERALI, H. D., A. L. SOYSTER, F. H. MURPHY AND S. SEN. 1984. Intertemporal Allocation of Capital Cost in Electric Utility Capacity Expansion Planning Under Uncertainty. *Mgmt. Sci.* **30,** 1–19.

SMITH, C. R., AND W. T. GRANDY (EDS.). 1985. *Maximum-Entropy and Bayesian Methods in Inverse Problems*. Reidel, Boston.

STIGLER, G. J. 1945. The Cost of Subsistence. *J. Farm Econ.* **27,** 303–314.

VANDERBEI, R. J. 1992. LOQO User's Manual. Technical Report SOR 92-5, Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.

VON NEUMANN, J., AND O. MORGENSTERN. 1953. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, New Jersey.

WAHBA, G. 1977. Practical Approximate Solution to Linear Operator Equations When the Data Are Noisy. *SIAM J. Num. Anal.* **44,** 651–667.

WETS, R.-J. B. 1966. Programming Under Uncertainty: The Equivalent Convex Program. *SIAM/J. Appl. Math.* **14**(1), 89–105.

WETS, R.-J. B. 1974. Stochastic Programs With Fixed Resources: The Equivalent Deterministic Problem. *SIAM Rev.* **16,** 309–339.

WETS, R.-J. B. 1983. Solving Stochastic Programs With Simple Recourse. *Stoch.* **10,** 219–242.

ZENIOS, S. A. 1994. Data Parallel Computing for Network-Structured Optimization Problems. *Comput. Optim. and Applic.* **3,** 199–242.

ZENIOS, S. A., A. DRUD AND J. M. MULVEY. 1989. Balancing Large Social Accounting Matrices With Nonlinear Network Programming. *Networks* **17,** 569–585.

ZENIOS, S. A., AND P. KANG. 1993. Mean-Absolute Deviation Portfolio Optimization for Mortgage Backed Securities. *Anns. Opns. Res.* **45,** 433–450.

ZENIOS, S. A., M. C. PINAR AND R. S. DEMBO. 1994. A Smooth Penalty Function Algorithm for Network-Structured Problems. *Eur. J. Opnl. Res.* **78,** 1–17.

ZENIOS, S. A., AND S. A. ZENIOS. 1992. Robust Optimization for Matrix Balancing From Noisy Data. Report 92-01-02, Decision Sciences Department, University of Pennsylvania, Philadelphia.