Current Trends in Stochastic Programming Computation and Applications

John R. Birge[1]
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, Michigan 48109-2117
USA
8/14/95

Abstract: While decisions frequently have uncertain consequences, optimal decision models often replace those uncertainties with averages or best estimates. Limited computational capability may have motivated this practice in the past. Recent computational advances have, however, greatly expanded the range of stochastic programs, optimal decision models with explicit consideration of uncertainties. This paper describes basic methodology in stochastic programming, recent developments in computation, and some practical application examples.

[1]This work is supported by the National Science Foundation under Award Number DDM-9215921.

1 Introduction

Decision making inherently involves consideration of uncertain outcomes. While computational advances in mathematical programming tools have aided decision making in many areas, their greatest impact may lie in enhancing decision making under uncertainty through *stochastic programming*, the optimization branch explicitly concerned with models with random parameters.

In this article, we explore recent advances in computational capabilities for stochastic programs and the structure of problems that enables these procedures. We also provide some insight into the range of possible applications of the methods through a set of examples from actual practice.

The need for including uncertainty in complex decision models arose early in the history of mathematical programming. The first model forms, involving an action followed by observation and reaction (or *recourse*), appear in Beale [4] and Dantzig [36]. An alternative model (called *chance* or *probabilistically constrained programming*), was also developed quite early, principally by Charnes and Coooper [32]. Both forms have their origin in statistical decision theory (see, e.g., Wald [107]) but, in contrast to decision theory, stochastic programming has emphasized methods of solution and analytical solution properties over procedures for constructing objectives and updating probabilities.

Since the size of a stochastic program can grow rapidly as additional uncertainties are modelled, many of the algorithmic developments in the area have concentrated on methods to exploit stochastic program structure. For example, Dantzig and Madansky [39] introduced Dantzig-Wolfe decomposition as a possible method, while Van Slyke and Wets [104] developed a form of Benders [5] decomposition for stochastic programs. In general, these methods concentrate on linear models as we shall here.

Beyond algorithms, much of stochastic programming research has considered various theoretical properties. Important convexity, continuity, and stability properties appear, for example, in Wets [111] for recourse models. For chance-constrained models, Prékopa [87] derived key convexity conditions. Wets [114] provides a general overview of these results. In our consideration here, we assume conditions for convexity where needed while emphasizing actual methods of solution.

The range of stochastic programming applications is as broad as any in optimization. Early applications include an airline fleet assignment model by Ferguson and Dantzig [45]. In that model, planes are assigned to routes to maximize expected revenue given a distribution on passenger demand on each route. This application is particularly noteworthy because the authors exploit the generalized network structure of the their model with perhaps the first generalized network method. Their solution also yielded a net gain of $142,000 or 9% over a solution that ignores uncertainty.

We consider a sample of other applications in Section 4 of this paper. Before that, we give a brief overview of stochastic programs in Section 2. We then describe various solution techniques and their computational implementations in Section 3. Our emphasis here is on methods with results in practically-sized, large-scale problems. We mainly discuss linear models because of their more consistently developed state at this time.

2 Formulating Stochastic Programs

The simplest form of a stochastic program may be the news vendor (also know as the "newsboy") problem. We use this familiar example to illustrate some of the fundamental characteristics of a stochastic program and to movitate broader uses.

In the news vendor model, the vendor must determine how many papers (x) to buy now at cost c without knowing the demand represented by a random variable, ξ, with distribution function, F, and given a selling price, p. We assume no salvage value, so that any papers bought in excess of demand are simply discarded with no return.

Assuming that the vendor observes the same distribution on demand every day, a reasonable objective is to maximize expected daily profit:

$$\max_{x \geq 0} \phi(x) = -cx + \int_0^x p\xi dF(\xi) + \int_x^\infty pxdF(\xi), \tag{1}$$

which has the basic form of the stochastic linear program with recourse: action x, followed by observation (ξ), and recourse (sales at price p).

Program 1 has the added advantage of an analytical solution. In this case, assuming F is continuous with density, f, we can differentiate to find x^* optimal in (1) such that

$$-cx + p\xi f(\xi) + \int_x^\infty pxdF(\xi) - p\xi f(\xi) = 0,$$

or, the well-known result,

$$F(x^*) = \frac{p-c}{p}.$$

The formula above illustrates the problem in simply substituting expected values for random quantities, as is often done in practice and widely thought to be of little consequence. If this is done in (1), the obvious solution is $x^{det} = \bar{\xi} = \int \xi f(\xi) d\xi$, the mean demand. This value only matches x^* if $F(x^{det}) = \frac{p-c}{p}$. For symmetric distributions with mean equal median, that result only occurs if $p = 2c$.

In the more likely case of low margins, the results are quite different. Suppose, for example, that $p = \$0.25$ while $c = \$0.20$. In this case, $F(x^*) = 0.2$. Suppose a uniform demand on $[50, 150]$ with mean of 100. In this case, $x^* = 70$ while $x^{det} = 100$. The result is that $\phi(x^*) = -20(70) + 25(12) + 25(70)(4/5) = \$3.00$ while $\phi(x^{det}) = -20(100) + 25(75/2) + 25(100)(1/2) = \$1.875$. Thus, the stochastic model yields an increase of \$1.125 or 60% in profits over the deterministic formulation. This difference is called the *value of the stochastic solution* (VSS) (see Birge [10]).

The VSS is similar to the *expected value of perfect information* (EVPI) (see, e.g., Raiffa and Schlaifer [88]), but the VSS measures a different aspect of uncertainty. For example, EVPI compares the stochastic program solution value with the value of a situation in which all uncertainty is revealed before a decision is made (i.e., with perfect information). In the news vendor case, we have a perfect information solution, $x^{PI} = \xi$, with expected value, $\int_\Xi (p-c)\xi f(\xi) d\xi = (p-c)\bar{\xi}$. For the example given, we would have an expected value with perfect information (EVWPI) of \$5.00 and $EVPI = EVWPI - \phi(z^*) = \$2.00$.

In this case, $EVPI > VSS$ but this is not always true. In fact, either value can be zero while the other is positive (see Birge [10]). The difference is that the value of resolving uncertainty can be quite different from the cost of ignoring uncertainy in making a decision. Our emphasis is on this latter form where uncertainty cannot be resolved further (in some economic way) but computational short cuts, such as substituting mean values for random parameters, might still be used.

This simple example illustrates another difficulty for modelers attempting to use mean values alone. If the deterministic model is followed, then one might expect a profit of $(p-c)\bar{\xi}$ or \$5.00 in this case. The vendor would, of course, be disappointed to find only \$1.875 , or \$3.125 (62.5%) less than the value using means alone. In fact, this loss compared to the mean value problem *always* occurs when mean values for random constraint parameters are substituted in convex stochastic programs (see, e.g., Madansky [69] or Birge and Wets [26]). This result may explain some losses of confidence in optimization modelers when results do not turn out as well as the model predicts.

To generalize the news vendor problem to other models, we preserve the characteristic of action, observation, and reaction. The new vendor problem is an example of a *two-stage stochastic linear program with (fixed) recourse* (2S-SLPR), which is written:

$$\begin{array}{cc} \min_{x \in \Re^{n_1}} & c^T x + \mathcal{Q}(x) \\ \text{s. t.} & Ax = b, \\ & x \geq 0, \end{array} \qquad (2S-SLPR)$$

where $\mathcal{Q}(x) = \int_\Xi Q(x,\xi) P(d\xi)$, with $Q(x,\xi) =$

$$\begin{array}{cc} \min_{y \in \Re^{n_2}} & q^T y \\ \text{s. t.} & Wy = h - Tx, \\ & y \geq 0, \end{array} \qquad (LRP)$$

and $b \in \Re^{m_1}$, $W \in \Re^{m_2 \times m_2}$, and $\xi = (q, h, T_{\cdot 1}, \ldots, T_{\cdot N_2})$ is an underlying random vector defined on a probability space, $(\Xi, \mathcal{A}, P)$. The resource is called fixed because W is deterministic. While W could also be random, it is usually assumed fixed to obtain clear definitions of feasible solutions (see, for example, Kall [55]).

To see how the news vendor problem fits the (2S-SLPR) framework, note that in (1), we have

$$Q^{news}(x,\xi) = \{\min -py | 0 \leq y \leq x, y \leq \xi\}. \qquad (2)$$

The news vendor model can also be written as a special form of (2S-SLPR) called *simple recourse* in which $W = [I | -I]$, forcing the y values by the sign of the components of $h - Tx$. While several methods are quite efficient for this problem (see, for example, Wets [112] and Nazareth and Wets [76]), the simple nature of the recourse makes $\mathcal{Q}(x)$ straightforward to evaluate so that the simple recourse model can be solved as a convex (with finite random variables, also piecewise linear) program with linear constraints. (Note that $Q(x,\xi)$ is convex in x, yielding $\mathcal{Q}(x)$ convex.)

An additional common characteristic of stochastic programs is to have a constraint that holds with a certain probability. For example, the news vendor may want to have papers for at least a fraction $\alpha < 1$ of the potential customers to maintain good public relations. The addition of these *probabilistic constraints* or *chance constraints*, generally of the form,

$$P\{T'x \geq h'\} \geq \alpha, \tag{3}$$

creates a *chance* or *probabilistically constrained* stochastic program. These constraints are often simply replaced by a deterministic equivalent form, $\bar{T}'x \geq \bar{h}'$, and added to the deterministic constraints. Since this action does not generally change problem structure, our development here only considers recourse problems with all probabilistic constraints implicitly represented in the deterministic constraints.

The choice of a recourse model, a chance-constrained model or some combination of these depends on the decision makers' attitute toward risk. While (2S-SLPR) appears to imply risk neutrality with its linear objective, we can model many kinds of risk attitudes through appropriate discount factors and constraint modification without creating nonlinearties (see Birge [13]).

In some cases, the decision makers may have multiple objectives or other utility forms. The (2S-SLPR) model can be easily updated to replace $c^T x$ by $c(x)$ and $q^T y$ by, for example, $U(q,y)$. Many problems involve not just a single observation and reaction but a sequence of these events. In this case, with linear objectives, we have the *multi-stage stochastic linear program with recourse* (MS-SLPR):

$$\begin{array}{llllll}
\min z & = c_1 x_1 & +E_\Omega[c_2(\omega^1)x_2(\omega^1)+ & \cdots + c_H(\omega^{H-1})x_H(\omega^{H-1})] & & \\
\text{s. t.} & W_1 x_1 & & & & = h_1, \\
 & T_1(\omega^1)x_1 & +W_2 x_2(\omega^1) & & & = h_2(\omega^1), \\
 & & \cdots & \cdots & & \vdots, \\
 & & T_{H-1}(\omega^{H-1})x_{H-1}(\omega^{H-1}) & +W_H x_H(\omega^{H-1}) & & = h_H(\omega^{H-1}), \\
 & x^1 \geq 0; & x_t(\omega^{t-1}) \geq 0, a.s.; & t = 2, \ldots, H; & &
\end{array} \tag{$MS-SLPR$}$$

where c_1 is a known vector in $\Re^{n_1}$, h_1 is a known vector in $\Re^{m_1}$, each W_t is a known $m_t \times n_t$ matrix (for fixed recourse), and we represent the random observations at time $t+1$ through a random element, ω_t, and a history process, $\omega^t = \{\omega_1, \ldots, \omega_t\}$, defined on a probability space, $(\Omega_1 \times \cdot \times \Omega_t, \mathcal{A}^t, P^t)$. We then have the subscript notation for a single period and the superscript to denote all previous periods (and an iteration counter in the algorithm discussion below). The complete data process, $\{\omega = (\omega_1, \ldots, \omega_{H-1})\}$, is defined on $(\Omega, \mathcal{A}, P)$. The random observations at $t+1$ are N_t-vectors, $\xi_{t+1}(\omega^t) = (c_{t+1}(\omega^t), h_{t+1}(\omega^t), (T_t)_{1\cdot}(\omega^t), \ldots, (T_t)_{m_{t+1}\cdot}(\omega^t))$.

This type of formulation becomes somewhat cumbersome since we have implicitly restricted the decisions to depend only on past data. The more general formulation includes explicit constraints to maintain this property.

More generally, we may have nonlinear functions in the objectives of constraints as well as integer variable restrictions. In this case, we can formulate a general multistage stochastic program with recourse:

$$\begin{array}{ll} \inf_{\mathbf{x}\in\mathcal{N}} & \mathrm{E}[\sum_{t=0}^{H} f_{t+1}(\omega, x^t(\omega), x_{t+1}(\omega))], \\ & \text{s. t. } x_t(\omega) \in X_t(\omega) \text{ a. s. }, \end{array} \tag{4}$$

where "E" is expectation with respect to the overall sigma field of events, $\mathcal{A}$, and $\mathcal{N}$ denotes the closed linear subspace of *nonanticipative processes*. Nonanticipativity of the decision process at time t implies that decisions must only depend on the data up to time t, i.e. $x^t(\omega)$ must be $\mathcal{A}^t$–measurable, where $\mathcal{A}^t$ is the sigma-field of events up to time t. An alternative characterization of this *nonanticipative* property is that $x^t(\omega) = \mathrm{E}\{x^t(\omega)|\mathcal{A}^t\}$ a.s., $t = 0, \ldots$, where $\mathrm{E}\{\cdot|\mathcal{A}^t\}$ is conditional expectation with respect to the σ–field $\mathcal{A}^t$. Using the projection operator $\Pi^t : z \to \Pi^t z := \mathrm{E}\{z|\mathcal{A}^t\}, t = 0, \ldots$, this is equivalent to

$$(I - \Pi^t)x^t = 0, \; t = 0, \ldots \; . \tag{5}$$

In our previous examples, the contraints in (2S-SLPR) were implicit by replacing $x(\omega)$ with a single value, x, wherever $x(\omega)$ might appear.

The overwhelming difficulty in stochastic programming stems from the evaluation of random functions and their expectations, such as $\mathcal{Q}(x)$ in (2S-SLPR) or an analogously defined value function in (MS-SLPR), $\mathcal{Q}_t(x_t)$, to reflect the expectation of all future objective terms given decision x_t at time t. Except in special cases, such as the simple recourse case mentioned above, practical solution requires some form of approximation. In some (perhaps most practical) situations, a distribution may not be known. In these cases, an approximating distribution may be chosen to yield results which can be far superior to a single best estimate or mean value substitution for all random parameters.

The majority of approximation methods involve some discretization of an underlying problability measure. The discretization may be chosen to provide upper or lower bounds on values with a known probability measure or unknown distribtuion with certain known characteristics (such as mean and variance ranges, see, e.g. Birge and Wets [26] and Kall, Ruszczyński and Frauendorfer [57]). Similar continuous bounding approximations replace the original high-dimensional integral with a combination of low-diminesional integrals (see Birge and Wallace [25], Birge and Wets [28], Wallace [108], Birge and Qi [22]). Each of these methods relies heavily on the convexity of the recourse problem.

Another approach is to attempt to find improvement relative to a range of possible distributions or objective functions (see Marti [72]). The other common technique is to use an empirical distribution based on past observations or some random generation scheme. For random methods, they can either be applied to a single large sample with useful asymptotic characteristics (see, for example, Dupačová and Wets [42], King and Rockafellar [61], Römisch and Schultz [93], Shapiro [98]) or implemented in conjunction with an algorithm (see, e.g., Ermoliev [44], Gaivoronski [47], Higle and Sen [50], and Pflug [83]).

While of approximation choices are critical for effective implementation, we concentrate on solutions given an approximation to focus on the range of current computation. The next section presents the basic methods that exploit stochastic program structure.

3 Optimization Methods using Stochastic Programming Structure

Our goal in this section is to describe the fundamental methods and current experience with their computational implementation. The key factor we stress is how the various methods take advantage of problem structure. We do not consider broad general techniques except for comparisons.

3.1 Direct Methods

In this section, we consider solution methods that use standard optimization procedures without large-scale strategies such as decomposition and relaxation. We still, however, wish to concentrate on methods that use specific stochastic program structure. The computational results in this area are generally for linear models so we also focus on the basic extreme point (simplex) method and interior point methods.

3.1.1 Extreme Point Methods

For extreme methods, we first consider the two-stage model, (2S-SLPR) with a discrete distribution supported on, $\Xi = \{\xi_1, \ldots, \xi_N\}$, with associated probabilities, $p_1, \ldots, p_N$. With these assumptions, (2S-SLPR) becomes:

$$
\begin{array}{lccccc}
\min_{x,y} & c^T x & +p_1 q_1 y_1 + & \cdots & +p_N q_N y_N & \\
\text{s. t.} & Ax & & & & = b, \\
 & T_1 x & +W y_1 & & & = h_1, \\
 & \vdots & & \ddots & & \vdots \\
 & T_N x & & & W y_N & = h_N, \\
 & x, & y_1, & \ldots & y_N & \geq 0.
\end{array}
\tag{6}
$$

The structured direct methods take advantage of the nonzero pattern of (6), called *dual block angular* . Extreme point methods consider a basis from the matrix in (6) and attempt a specific factorization (see Kall [56] and Strazicky [100]). This full factorization employs separate submatrix factorizations corresponding to diagonal blocks (with W components) and a single (at most $n_1 \times n_1$) additional working basis.

While some advantages over standard simplex methods were reported in early implementations (see Strazicky [100]), few implementations specifically exist for this structure due to the existing sparse factorization schemes in general purpose linear programming codes and the relatively limited improvement potential for extreme point methods over commercial codes. It has also been shown (see Birge [12]) that a variant of the factorization method can trace the same iterates as decomposition methods, and, hence, that decomposition may afford a broader range of possibilities.

Extensions of basis factorization techniques are also possible in multiple stages (Birge [9]), but here the additional specific structure appears to outweigh any computational savings. The results may, however, be quite different for interior point methods. The key may be the degree to which each method can be performed in parallel.

3.1.2 Interior Point Methods

Factorization schemes also offer substantial promise for interior point methods, where there is much speculation that the solution effort grows linearly in the size of the problem. Discussion of computational results appears, for example, in Lustig, Marsten, and Shanno [67, 68]. In interior point methods, problems of form (6) are again solved. If we denote the constraint matrix in (6) by $\bar{A}$, then the effort in most interior point methods is dominated by computations with a matrix, $M = \bar{A}D^2\bar{A}^T$, where D is a diagonal matrix, so that, letting $D_0 = diag(x^\nu)$, $D_N = diag(y_N^\nu), k = 1, \ldots, N$, we would have

$$M = \begin{pmatrix} AD_0^2A^T & AD_0^2T_1^T & \cdots & AD_0^2T_N^T \\ T_1D_0^2A^T & T_1D_0^2T_1^T + WD_1^2W^T & \cdots & T_1D_0^2T_N^T \\ \vdots & \vdots & \ddots & \vdots \\ T_ND_0^2A^T & T_1D_0^2T_N^T & \cdots & T_ND_0^2T_N^T + WD_N^2W^T \end{pmatrix}, \quad (7)$$

which is much denser than the original constraint matrix in (6). In this case, a straightforward implementation of an interior point method that solves systems with M is quite inefficient.

Note, however, that M in (7) has a great deal of structure that can be exploited in any solution scheme. This is the object of the factorization scheme given by Birge and Qi [21], based on the following proposition.

Theorem 1 *Let $S_0 = I_2 \in \Re^{m_1 \times m_1}$, $S_l = W_l D_l^2 W_l^T$, $l = 1, \ldots, N$, $S = \mathrm{diag}\{S_0, \ldots, S_N\}$. Then $S^{-1} = \mathrm{diag}\{S_0, S_1^{-1}, \ldots, S_N^{-1}\}$. Let I_1 and I_2 be identity matrices of dimensions n_1 and m_1 respectively. Let*

$$G_1 = (D_0)^{-2} + A^T S_0^{-1} A + \sum_{l=1}^{N} T_l^T S_l^{-1} T_l, G_2 = -AG_1^{-1}A^T, \quad (8)$$

$$U = \begin{pmatrix} A & I_2 \\ T_1 & 0 \\ \vdots & \vdots \\ T_N & 0 \end{pmatrix}, V = \begin{pmatrix} A & -I_2 \\ T_1 & 0 \\ \vdots & \vdots \\ T_N & 0 \end{pmatrix}.$$

If A and W have full row rank, then G_2 and M are invertible and

$$M^{-1} = S^{-1} - S^{-1}U \begin{pmatrix} I_1 & G_1^{-1}A^T \\ 0 & I_2 \end{pmatrix} \begin{pmatrix} I_1 & 0 \\ 0 & G_2^{-1} \end{pmatrix} \begin{pmatrix} I_1 & 0 \\ A & I_2 \end{pmatrix} \begin{pmatrix} G_1^{-1} & 0 \\ 0 & -I_2 \end{pmatrix} V^T S^{-1}. \quad (9)$$

Proof: Follows Birge and Qi [21]. See also Birge and Holmes [18]. □

Following the assumptions and using the complexity result in Karmarkar [58], the number of arithmetic operations using this factorization can be reduced from $O((n_1 + Kn_2)^4)$ as in the general projective scaling method. Birge and Qi show that the effort is, in fact, dominated by $O(N(n_2^3 + n_2^2 n_1 + n_2 n_1^2))$. It is also possible to reduce this bound further as Karmarkar does with a partial rank-one updating scheme. In this case, for $n = n_1 + Nn_2$, the complexity using the factorization in (9) becomes $O((n^{0.5}n_2^2 + n\max\{n_1, n_2\} + n_1^3)nL)$ for the entire algorithm, where L represents the size of the data, or, if $N \sim n_1 \sim n_2$, the full arithmetic complexity is $O(n^{2.5}L)$, compared to Karmarkar's general result of $O(n^{3.5}L)$. Thus, the factorization in (9) provides an order of magnitude improvement over a general solution scheme if the number of realizations N approaches the number of variables in the first and second stage. The same improvement applies to lower order primal-dual interior point methods as well.

In practice, we would not compute M^{-1} explicitly. The work in (9) is dominated by the effort to solve systems of the form

$$Mv = u, \tag{10}$$

using

$$v = p - r, \tag{11}$$

where

$$\begin{array}{ll} Sp & = u, \\ Gq & = V^T p, \\ Sr & = Uq. \end{array} \tag{12}$$

The systems in (12) require solving systems with S_l, computation of G_1 and G_2 and solving systems with G_1 and G_2. In practice, we find a Cholesky factorization of each S_l, use them to find G_1 and G_2 and find Cholesky factorizations of G_1 and G_2.

Before we describe results using the factorization in (9), we consider some other options for interior point methods. These possibilities are:

- Schur complement updates;
- Column splitting;
- Solution of the dual;
- Direct solution of the symmetric indefinite augmented system.

The Schur complement approach is used in many interior point method implementations (see, for example, Choi, et al. [33]). The basic idea is to write M as the sum of a matrix with sparse columns, $A_s D_s^2 A_s^T$, and a matrix with dense columns, $A_d D_d^2 A_d^T$. Using a Cholesky factorization of the sparse matrix, $LL^T = A_s D_s^2 A_s^T$, the method involves solving $Mu = v$ by:

$$\begin{pmatrix} LL^T & -A_d D_d \\ D_d A_d^T & I \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} u \\ 0 \end{pmatrix}, \tag{13}$$

which involves solving $[I + D_d A_d^T (LL^T)^{-1} A_d D_d]w = -D_d A_d^T (LL^T)^{-1} b$ and $LL^T v = b + A_d D_d w$, where $I + D_d A_d^T (LL^T)^{-1} A_d D_d$ is a Schur complement. The Schur complement is thus quite similar to the factorization method given above. If every column of x is considered a dense column, then the remaining matrix is quite sparse but rank deficient. The factorization in (9) is a method for maintaining an invertible matrix when $A_s D_s^2 A_s^T$ is singular. It can thus be viewed as an extension of the Schur complement to the stochastic linear program. Because of the possible rank deficiency and the size of the Schur complement, the straightforward Schur complement approach in (13) is quick but leads to numerical instabilities as Carpenter et al. [30] report.

Carpenter, et al., also propose the column splitting technique. The basic idea is to rewrite problem (6) with explicit constraints on nonanticipativity. The formulation then becomes:

$$\begin{array}{llr}
\min & \sum_{k=1}^{N} p_k (c^T x_k + q_k^T y_k) & (14) \\
\text{s.t.} & A x_k = b, & (15) \\
 & T_k x_k + W y_k = h_k, \quad k = 1, \ldots, N; & (16) \\
 & x_k - x_{k+1} = 0, k = 1, \ldots, N-1; & (17) \\
 & x_k \geq 0, \quad y_k \geq 0, \quad k = 1, \ldots, N. & (18)
\end{array}$$

The difference now is that the constraints in (15) and (16) separate into separate subproblems k and constraints (17) link the problems together. Alternating constraints, (15), (16) and (17) for each k in sequence, the full constraint matrix has the form:

$$\bar{A} = \begin{pmatrix}
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T_1 & W & 0 & 0 & 0 & 0 & 0 & 0 \\
I & 0 & -I & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & A & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & T_2 & W & 0 & 0 & 0 & 0 \\
0 & 0 & I & 0 & -I & 0 & 0 & 0 \\
\vdots & \vdots & 0 & \ddots & \vdots & \ddots & 0 & \vdots \\
0 & 0 & 0 & 0 & I & 0 & -I & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & A & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & T_k & W
\end{pmatrix}. \tag{19}$$

If we form $\bar{A}\bar{A}^T$, then we obtain $\bar{A}\bar{A}^T =$

$$\begin{pmatrix} AA^T & AT_1^T & A & 0 & 0 & 0 & 0 & 0 \\ T_1A^T & T_1T_1^T + WW^T & T_1 & 0 & 0 & 0 & 0 & 0 \\ A^T & T_1^T & 2I & -A^T & 0 & 0 & 0 & 0 \\ 0 & 0 & -A & AA^T & AT_2^T & A & 0 & 0 \\ 0 & 0 & T_2A^T & \begin{matrix} T_2T_2^T \\ +WW^T \end{matrix} & T_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & T_2^T & 2I & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & 0 & \vdots \\ 0 & 0 & 0 & A^T & T_{N-1}^T & 2I & -A^T & 0 \\ 0 & 0 & 0 & 0 & 0 & -A & AA^T & AT_N^T \\ 0 & 0 & 0 & 0 & 0 & 0 & T_NA^T & \begin{matrix} T_NT_N^T \\ +WW^T \end{matrix} \end{pmatrix}, \tag{20}$$

which is clearly much sparser than the original matrix in (7). It is, however, larger than M in (7) leading to a trade-off of increased size for reduced density.

The third additional approach is to form the dual of (6) and to solve that problem using a basic interior point method. The dual problem is:

$$\begin{array}{lcr} \max & b^T\rho + \sum_{k=1}^N p_k \pi_k^T h_k & (21) \\ \text{s.t.} & A^T\rho + \sum_{k=1}^N p_k T_k^T \pi_k \le c, k = 1, \ldots, N, & (22) \\ & W^T \pi_k \le q, k = 1, \ldots, N, & (23) \end{array}$$

where the variables are not restricted in sign. For this problem, we can achieve a standard nonnegative variable, equality constraint form by splitting the variables π_k and ρ into differences of nonnegative variables and by adding slack variables to constraints (22) and (23)[2]. In this way the constraint matrix for (22) and (23) becomes:

$$A' = \begin{pmatrix} A^T & -A^T & T_1^T & -T_1^T & 0 & T_2^T & -T_2^T & \cdots & T_N^T & -T_N^T & 0 & I \\ 0 & 0 & W^T & -W^T & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W^T & -W^T & I & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & \ddots & 0 & 0 & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & W^T & -W^T & I & 0 \end{pmatrix}. \tag{24}$$

The matrix in (24) may again be much larger than the original, but the gain comes in considering $A'A'^T$ which is now:

$$\begin{pmatrix} 2(A^TA + \sum_{k=1}^N T_k^T T_k) + I & 2T_1^TW & 2T_2^TW & \cdots & 2T_k^TW \\ 2W^TT_1 & 2W^TW + I & 0 & 0 & 0 \\ 2W^TT_2 & 0 & 2W^TW + I & 0 & 0 \\ \vdots & 0 & 0 & \ddots & 0 \\ 2W^TT_k & 0 & 0 & 0 & 2W^TW \end{pmatrix}, \tag{25}$$

[2]The dual problem no longer may have a bounded set of optima causing some theoretical difficulties for convergence results. In practice, bounds are placed on the variables to guarantee convergence.

with inherent sparsity. In fact, it is not necessary to take the dual in order to use this alternative factorization form. As shown in Birge, Freund and Vanderbei [17], the Sherman-Morrison-Woodbury formula can be applied to the original problem in (6) so that computations with the structure of $\bar{A}\bar{A}^T$ are replaced by computations with $\bar{A}^T\bar{A}$.

In Carpenter, et al. [30], increasing numbers of scenarios were included into a network formulation of a financial decision problem. They used the variable splitting option with an additional observation that many of the T_k columns were zero and that the corresponding variables need not be split. By splitting only those variables with nonzero T_k entries, they developed a *partial splitting* model that proved most effective. Their results with this partial splitting model show an approximately linear increase in speed-up for using split variables compared to the original primal form with the interior point code OB1 (Lustig, Marsten, Shanno [67]). For 18 scenarios, the speed-up was 1.48 while for 72 scenarios of the same problem, the partial splitting speed-up was 11.82.

Carpenter, et al. also used the Schur complement approach and achieved additional speed-ups over partial splitting, although the additional speed-up was about the same (essentially a factor of 2) for all numbers of scenarios. They did, however, report an order of magnitude increase in a measure of primal infeasibility due to numerical instability with the basic Schur complement approach.

In Birge and Holmes [18], the dual approach and the primal factorization of Birge and Qi are compared. We repeat the results of some of these problems in Table 1. The problems are a test set of stochastic programs based on the multistage problems compiled by Ho and Loute [51]. They represent problems in economic development, agriculture, energy modeling, structural design, traffic assignment, and production scheduling. The problems are solved as multistage stochastic programs in Birge [11] with a small number of scenarios. For the interior point tests, reported in Table 1, they were formed as two-stage problems with increasing numbers of second stage realizations or scenarios.

Table 1 shows the number of scenarios in each problem and the speed-up or the ratio of solution times for solving each problem in the original primal and the dual form to the time for solutions using the factorization in Proposition 2. An affine scaling form (without a centering term) of the interior point method given above was implemented in FORTRAN on an IBM 3090 at the University of Michigan. The program used SPARSPAK ([34]) to solve the systems of equations.

Table 1 shows that the dual form appears to be the most efficient. However, in some cases, this matrix became too large for the operating system. This was the case for larger versions of SCSD8, which has seven times as many columns as rows. Also, notice that the factorization approach appears relatively more efficient as the problem size increases, eventually surpassing the dual approach. The largest advantage of the factorization may be, however, that the operations in (12) for each S_I can be performed in parallel.

Yang and Zenios [117] have implemented the factorization in (9) in parallel on a Connection Machine CM-5 with up to 64 processors. Due to memory efficiencies in the parallel version, they actually achieve superlinear speedups in parallel compared

Problem	*No. of Scenarios*	*Speed-ups over:*	
		Primal	*Dual*
SC205.02	8	2.156	0.315
SC205.03	16	7.101	0.331
SC205.04	32	26.753	0.379
SC205.05	64	> 30	0.404
SCRS8.02	8	1.571	0.224
SCRS8.03	16	9.396	0.263
SCRS8.04	32	> 30	0.150
SCAGR7.02	8	4.426	0.395
SCAGR7.03	16	15.840	0.439
SCAGR7.04	32	> 30	0.507
SCAGR7.05	64	> 30	0.727
SCTAP.02	8	3.164	0.850
SCTAP.03	16	12.825	1.054
SCTAP.04	32	> 30	1.250
SCSD8.01	4	*	2.691
* - Algorithm for primal problem did not converge.			

Table 1: Speed-ups for the Birge-Qi factorization over primal and dual interior point method solutions.

to serial times. They compared their method with Vanderbei's LOQO interior point solver (Vanderbei [105]) which enables alternative factorizations of a symmetric indefinite system (Vanderbei and Carpenter [106]). Yang and Zenios report that serial times using the factorization in (9) do not generally overtake LOQO except for very large problems (as suggested in Table 1), but that their parallel processing achieves order of magnitude speedups over LOQO on large versions of the test problems from [18]. They also report solutions of problems with up to 18 million variables and almost 3 million constraints.

The symmetric indefinite system considered in LOQO corresponds to the system in primal-dual interior point methods that find primal step directions, $\delta\bar{x} = (\delta x, \delta y)$, and dual step directions $\delta\bar{\pi} = (\delta\pi, \delta\rho)$, to satisfy

$$\begin{pmatrix} -I & D\bar{A}^T \\ \bar{A}D & 0 \end{pmatrix} \begin{pmatrix} \delta\bar{x} \\ \delta\bar{\pi} \end{pmatrix} = \begin{pmatrix} v \\ w \end{pmatrix}, \tag{26}$$

where v and w depend on the iterate. The primal method can be viewed as eliminating for $\delta\bar{x}$ first in (26) while the dual method effectively solves for $\delta\bar{\pi}$ first. Other methods can, however, take direct advantage of the structure in (26) without resorting to either of these extreme cases.

Czyzyk, et al. [35] report on solutions using the symmetric indefinite, augmented system in (26) and a pivot ordering that yields a factorization similar to that (9). Using large versions of the test problems in Birge and Holmes, they report improvement over split variable formulations on all problems except SCRS8 and speedups ranging up to two orders of magnitude over split variable formulations.

For multi-stage problems, Berger et al. [6] provide a similar factorization form, called *tree dissection*, for the augmented system and a possibly quadratic objective. The general idea there is to force pivots along links joining scenarios at different points in time. Their implementation of this strategy achieves several orders of magnitude improvements over LOQO's default setting, and as much as an order of magnitude improvement over LOQO forced to use a "dual first" strategy (and the subsequent $\bar{A}^T\bar{A}$ form). They were also able to solve significantly larger problems than other methods with as many as one million constraints and almost two million variables.

From these developments, it appears that special purpose factorizations offer particular advantages for interior point methods for stochastic programs. The availability of parallel processing and capabilities for performing necessary routines in parallel may determine which form that factorization should take. Virtually all of the results report improvements for large problems over direct simplex-based techniques. Decomposition methods, however, have often achieved even greater efficiencies. We discuss these methods in the next section.

3.2 Decomposition Approaches

The general idea behind decomposition methods for (2S-SLPR) and (MS-SLPR) is that a smaller problem in x (or x_1) alone can be solved with some approximation of the recourse function, $\mathcal{Q}(x)$. This idea started with a method called *L-Shaped* by

Van Slyke and Wets [104] that is a form of Benders' [5] decomposition. From that, various extensions have been suggested. We discuss these below.

3.2.1 The L-Shaped Method

Consider the general formulation in (2S-SLPR). The basic idea of the L-shaped method is to approximate the nonlinear term in the objective of these problems. A general principle behind this approach is that, since the nonlinear objective term (the recourse function) involves a solution of all second-stage recourse linear programs, we would like to avoid numerous function evaluations for it. We therefore use that term to build a master problem in x but we only evaluate the recourse function exactly as a subproblem.

To make this approach possible, we assume again that the random vector $\boldsymbol{\xi}$ has finite support. Consider the formulation in (6). The dual-block-angular structure led to the name, *L-shaped Method*, for the decomposition method we describe here. The method is essentially a Dantzig-Wolfe decomposition (inner linearization) [40] of the dual or a Benders' decomposition (outer linearization) [5] of the primal. This method, due to Van Slyke and Wets [104] for stochastic programming, also considers feasibility questions of particular relevance for stochastic programs.

L-shaped Method

Step 0. Set $s = t = \nu = 0$.

Step 1. Set $\nu = \nu + 1$. Solve the linear program (27)-(29).

$$\begin{array}{lrlll} \min & z = c^T x + \theta & & & (27) \\ \text{s.t.} & Ax & = b, & & \\ & D_\ell x & \geq d_\ell, & \ell = 1, \ldots, s; & (28) \\ & E_\ell x + \theta & \geq e_\ell, & \ell = 1, \ldots, t; & (29) \\ & x \geq 0, & \theta \in \Re. & & \end{array}$$

Let (x^ν, θ^ν) be an optimal solution. If no constraint (29) is present, θ^ν is set equal to $-\infty$ and not considered in the computation of x^ν.

Step 2. For $k = 1, \ldots, N$ solve the linear program

$$\begin{array}{llll} \min & w' = \ e^T v^+ + e^T v^- & & (30) \\ \text{s.t.} & Wy + Iv^+ - Iv^- & = h_k - T_k x^\nu, & (31) \\ & y \geq 0, \quad v^+ \geq 0, & v^- \geq 0, & \end{array}$$

where $e^T = (1, \ldots, 1)$, until, for some k, the optimal value $w' > 0$. In this case, let σ^ν be the associated dual multipliers and define

$$D_{s+1} = \sigma^\nu T_k \tag{32}$$

and

$$d_{s+1} = \sigma^\nu h_k \tag{33}$$

to generate a constraint (called a *feasibility cut*) of type (28). Set $s = s + 1$, add to the constraint set (28), and return to Step 1. If, for all k, $w' = 0$, go to Step 3.

Step 3. For $k = 1, \ldots, K$ solve the linear program

$$\begin{array}{ll} \min & w = q_k^T y \\ \text{s.t.} & Wy = h_k - T_k x^\nu, \\ & y \geq 0. \end{array} \tag{34}$$

Let π_k^ν be the simplex multipliers associated with the optimal solution of Problem k of type (34). Define

$$E_{t+1} = \sum_{k=1}^{K} p_k \cdot \pi_k^\nu T_k \tag{35}$$

and

$$e_{t+1} = \sum_{k=1}^{K} p_k \cdot \pi_k^\nu h_k. \tag{36}$$

Let $w^\nu = e_{t+1} - E_{t+1} x^\nu$.
If $\theta^\nu \geq w^\nu$, stop, x^ν is an optimal solution. Otherwise, set $t = t + 1$, add to the constraint set (29), and return to Step 1.

The method solves an approximation of (2S-SLPR) using an outer linearization of $\mathcal{Q}$. Two types of constraints are sequentially added: (i) feasibility cuts (28) determining $\{x | \mathcal{Q}(x) < +\infty\}$ and (ii) *optimality cuts* (29) which are linear supports of $\mathcal{Q}$ on its domain of finiteness.

This method has also been generalized to multiple stages (see Birge [11]). The basic idea of the Nested L-Shaped or Benders' decomposition method is to place cuts on the period t value function, $\mathcal{Q}_{t+1}(x_t)$, and to add other cuts to achieve an x_t that has a feasible completion in the future. The cuts represent successive linear approximations of $\mathcal{Q}_{t+1}$. Due to the polyhedral structure of $\mathcal{Q}_{t+1}$, this process converges to an optimal solution in a finite number of steps.

In general, for every stage $t = 1, \ldots, H-1$, we have $k = 1, \ldots, N^t$ possible outcomes or scenarios that are joined to an *ancestor* scenario, $a(k)$ at $t-1$, and a set of *descendant* scenarios at $t+1$. We then have the following master problem that generates cuts to stage t and proposals for stage $t+1$:

$$\begin{array}{llll} \min & c_{t,k} x_{t,k} + \theta_{t,k} & & (37) \\ \text{s. t.} & W_t x_{t,k} & = h_{t,k} - T_{t-1,k} x_{t-1,a(k)}, & (38) \\ & D_{t,k}^j x_{t,k} & \geq d_{t,k}^j, j = 1, \ldots, r_{t,k}, & (39) \\ & E_{t,k}^j x_{t,k} + \theta_{t,k} & \geq e_{t,k}^j, j = 1, \ldots, s_{t,k}, & (40) \\ & x_{t,k} \geq 0, & & (41) \end{array}$$

where $x_{t-1,a(k)}$ is the current solution from the ancestor scenario, $a(k)$, and where for $t = 1$, we interpret $b = h_1 - T_0 x_0$ as initial conditions of the problem.

The nested decomposition method iterates among problems of the form (37-41) from stage t to $t+1$ passing forward proposals, x^t, and from $t+1$ to t passing backward prices in the form of cuts. With suitable choices of these movements, the method terminates finitely.

Many alternative strategies are possible in this algorithm for determining the next subproblem (37-41) to solve. For feasible solutions, one approach explores all scenarios at t before deciding to move to $t-1$ or $t+1$. For feasible iterations, the algorithm often proceeds in a consistent direction either forward (to $t+1$) or backward (to $t-1$) as long a movement in that direction is possible. This approach is the "fast forward-fast back" procedure proposed by Wittrock [115] for deterministic problems and implemented by Gassmann [48] for stochastic problems. One may alternatively enforce a move from t to $t-1$ ("fastback") or from t to $t+1$ ("fastforward") whenever it is possible. From experiments conducted by Gassmann [48] and Birge et al. [16], the fast-forward-fast-back sequencing protocol seems generally to work better than either of these alternatives.

This decomposition method also allows significant parallel computation. Ariyawansa and Hudson [1] show that the two-stage model scales relatively well on parallel processors when the serial methods do not take particular advantage of similarities among scenarios. Sequential solutions may, however, take advantage of a solution for a single subproblem (34) to solve others. The methods, called *trickling* (Gassmann [48]) or *bunching* (see Wets [113]), effectively generate a tree of optimal bases for subproblems (34). In some cases, these approaches achieve significant efficiencies.

Other efficiencies are often possible by replacing θ in (27,29) with $\sum_{k=1}^{N} p_k \theta_k$ and then placing cuts on each of the θ_k variables using only the kth subproblem prices, π_k. For a given number of main problem solutions, this *multicut* approach (from Birge and Louveaux [20]) produces a more complete characterization of the polyhedral structure of the graph of $\mathcal{Q}(x)$, resulting in generally fewer major iterations. The expense of an expanded main problem, however, sometimes overcomes this efficiency (in particular, for small problems).

Comparisons on these appproaches appear in Birge et al. [16]. Figure 1 (from [16]) presents comparisons for solution times of four problems with varying structures (see references in [16]). Note that the scales represent the logarithms of times in CPU seconds and the number of deterministic equivalent variables. The deterministic equivalent problems were solved on an IBM RS/6000 320H using IBM's OSL primal simplex algorithm (IBM [53]). Gassmann's MSLiP (Gassmann [48]) solved the problems with both single cuts and no trickling except at the last stage (MSLiP-SC) and multicuts and trickling throughout the subproblem tree (MSLiP-MC). ND-UM refers to a nested decomposition code developed at the University of Michigan, with times reflecting single cuts and no bunching or trickling. ND-Bunch times give some indication of bunching advantages on the STOCHFOR problem.

Figure 1 also presents the results from a parallel implementation under *Best Parallel*. This version implemented the nested decomposition method in parallel over a network of RS/6000s using the PVM (Geist et al. [49]) protocol. Large problem efficiencies (fractions of available CPU times used on all processors) exceeded 50% in all cases with an average of greater than 75% efficiency for the large problems.

Figure 1: Solution Time Comparisons.

In all cases, the deterministic equivalent problems became too large to solve directly on the available workstations. Of the examples here, the direct simplex method appears only superior (by a small amount) for the smaller STORM problems. Other studies (see Ruszczyński [96]), however, indicate that direct methods lose efficiency significantly on this problem as the number of scenarios increases. In all problems except SCFXM, the decomposition method times are approximately linear in the number of variables. For SCFXM, the growth is much less than linear due to the decomposition's ability to find a solution quickly that also solves most of the scenarios.

The results also show that trickling or bunching and multicuts generally offer greater efficiencies (except on SCFXM where a single scenario dominates the others). Other efficiencies are also possible in some problems by adding a *regularizing term*, $\frac{1}{2}\|x - x^{\nu}\|^2$, to the objective in (27). This approach from Ruszczyński [94, 95, 96] appears to offer advantages when initial cuts force the iterates, x^{ν}, to oscillate widely after initialization. The method can also be extended directly to multiple stages and again offers many possibilities for parallel computation (see Ruszczyński [95] and implementation by Korycki [63]).

The regularized method also introduces possibilities for nonlinear objectives in the decomposition framework. Louveaux [66] provides one approach for multistage problems with quadratic objectives. Following the deterministic approach for convex objectives in O'Neill [80], Pereira and Pinto [81, 82] developed a method with nonlinear objectives. Noël and Smeers [79] extended the basic method to treat nonlinear constraints within a time period. Further developments appear in Birge and Rosa [23] with solutions of examples with over 4000 variables involving nonlinearities and over 1400 nonlinear constraints.

Extensions of these cutting plane methods are also possible in problems with integer variables. If all integer variables are in the first stage, then the decomposition procedures apply directly (see Wollmer [116]). When integer variables appear in future stages, however, methods become more difficult because the recourse function, $\mathcal{Q}(x)$, loses convexity. One remedy is to force all integer variables (repeated for different scenarios) into a first stage, as done by Bienstock and Shapiro [8]. Alternatively, for two-stage problems, Laporte and Louveaux [64] provide valid cutting procedures that adapt the L-shaped method to problems with integer variables.

While these methods all apply to simplex solutions of (27-29) and (37-41) (or at least to a solution with optimal values to check convergence), interior point methods can also be adapted to take advantage of the problem structure without direct solution of the full deterministic equivalent problem. One approach, the *Algorithm of Cutting Planes from Analytic Centers*, appears in Bahn et al. [2]. They generate cuts to reduce a main problem feasible region and then move toward the center of that region with successive subproblem solutions and improved objective values. Their results confirm the potential for improvements over direct solvers and show possibilities for improvement over the simplex-based methods on certain problems.

3.2.2 Decomposition Methods with Sampling

As mentioned earlier, stochastic programming formulations often include some approximation of an underlying probability distribution. When that distribution is known, it is particularly common to use a sample approximation. A potential disadvantage of sampling, however, is that some effort might be wasted on an inaccurate approximation. An approach to avoid these problems is to use sampling within another algorithm without complete optimization. A natural candidate is to embed sampling into the L-shaped method described in the previous section. We consider two such approaches, one based on many samples to derive cuts (see Dantzig and Glynn [37], Dantzig and Infanger [38]) and the other based on a single sample to derive less accurate cuts that eventually drop away (Higle and Sen [50]).

The first approach by Dantzig, Glynn and Infanger is to sample $Q(x,\xi)$ in the L-shaped method instead of actually computing $\mathcal{Q}(x)$. Given an iterate x^k, the result is an estimate, $\mathcal{Q}^\nu(x^k) = (\frac{1}{\nu}(\sum_{i=1}^{\nu} Q(x^k,\xi^i))$, and an estimate of $\nabla\mathcal{Q}(x^k)$ as $\bar{\pi}_k^\nu = (\frac{1}{\nu}\sum_{i=1}^{\nu}\pi_k^i)$ where $\pi_k^i \in \partial Q(x^k,\xi^i)$. Now, for Q convex in x, one obtains

$$Q(x,\xi^i) \geq Q(x^k,\xi^i) + \pi_k^i(x - x^k), \tag{42}$$

for all x. We also have that

$$\mathcal{Q}^\nu(x) = (\frac{1}{\nu})(\sum_{i=1}^{\nu} Q(x,\xi^i)) \geq \mathcal{Q}^\nu(x^k) + \bar{\pi}_k(x - x^k) = LB_k^\nu(x), \tag{43}$$

where $\sqrt{\nu}$ times right hand side in (43) is asymptotically normally distributed (see, e.g., Shapiro [98]) with a mean value,

$$\sqrt{\nu}(\mathcal{Q}(x^k) + \nabla\mathcal{Q}(x^k)^T(x - x^k)), \tag{44}$$

which is a lower bound on $\sqrt{\nu}\mathcal{Q}(x)$, and a variance, which (assuming Q is almost surely differentiable with respect to x relative to the distribution P) is

$$\rho^k(x) = Var_Q(x^k)+(x-x^k)^T \int_\Xi (\nabla Q(x,\xi)-\nabla \mathcal{Q}(x))(\nabla Q(x,\xi)-\nabla \mathcal{Q}(x))^T P(d\xi))(x-x^k). \tag{45}$$

Thus, we place a cut on $\mathcal{Q}(x)$ as the right hand side of (43) as

$$\mathcal{Q}(x) \geq \mathcal{Q}^\nu(x^k) + \bar{\pi}_k(x - x^k) - \epsilon_k(x),$$

where $\epsilon_k(x)$ is an error term with zero mean and variance equal to $\frac{1}{\nu}\rho^k(x)$. Now, at iteration K, the L-shaped method involves the solution of:

$$\begin{array}{llll} \min & c^T x & +\theta & \\ \text{s. t.} & Ax & & = b, \\ & D^l x & & \geq d^l, l = 1, \ldots, s; \\ & E^k x & +\theta & \geq e^k, k = 1, \ldots, K; \\ & x \geq 0, & & \end{array} \tag{46}$$

where D^l, d^l is a feasibility cut as in (28), $E^k = -\bar{\pi}_k$, and $e^k = \mathcal{Q}^\nu(x^k) + \bar{\pi}_k(-x^k)$, where we count x^k iterations only when a finite $\mathcal{Q}^\nu(x^k)$ is found. Note that the generation of feasibility cuts occurs whenever ξ^i is sampled and $Q(x^k, \xi^i)$ is ∞.

It is shown in Birge and King [19] that, with probability at least α^2,

$$c^T x^{K+1} + \theta^{K+1} - \frac{1}{\sqrt{\nu}}\bar{\sigma}^{\max}\Phi^{-1}(\alpha^{\frac{1}{K}}) \leq \min_{Ax=b, x\geq 0}\{c^T x + \mathcal{Q}(x)\}, \tag{47}$$

where $\bar{\sigma}^{\max}$ is the maximum $\rho^k(x)$ over feasible x, and

$$\min_{Ax=b,x\geq 0}\{c^T x + \mathcal{Q}(x)\} \leq c^T x^{K+1} + \mathcal{Q}^\nu(x^{K+1}) + \Phi^{-1}(\alpha)(\frac{1}{\nu}\rho^{K+1}(x^{K+1}))^{0.5}. \tag{48}$$

Hence, (47) and (48) provide an α^2 confidence interval around the feasible minimum of $c^T x + \mathcal{Q}(x)$. If this interval becomes sufficiently small, one can stop with a high confidence solution.

Other approaches may also be used. Infanger [52] makes several assumptions that can lead to tighter confidence intervals on the optimal value. Solutions of large problems appear in Dantzig and Infanger [38] with these assumptions.

An alternative, called *stochastic decomposition*, by Higle and Sen [50] is to generate many cuts with small samples and to adjust these cuts to drop as the algorithm continues processing. They assume complete recourse and a known (probability one) lower bound on $Q(x, \xi)$ (e.g., 0). They also assume that the set of dual solutions to the recourse problem (34) is bounded, and that $\{x|Ax = b, x \geq 0\}$ and Ξ are also compact.

With these assumptions, the basic stochastic decomposition method generates iterates, x^k, and observations, ξ^k, in the following way.

Stochastic Decomposition Method

Step 1. Set $k = 0$, $\xi^0 = \bar{\xi}$, and let x^1 solve

$$\min_{Ax=b,x\geq 0} \{c^T x + Q(x, \xi^0)\}. \tag{49}$$

Step 2. Let $k = k + 1$ and ξ^k be an independent sample generated from $\boldsymbol{\xi}$. Find $\mathcal{Q}^k(x^k) = \frac{1}{k}\sum_{t=1}^{k} Q(x^k, \xi^t) = \frac{1}{k}\sum_{t=1}^{k} \pi_t^k(\xi^t - Tx^k)$. Let $E^k = \frac{1}{k}\sum_{t=1}^{k} \pi_t^k T$ and $e^k = \frac{1}{k}\sum_{i=1}^{k} \pi_t^k \xi^t$.

Step 3. Update all previous cuts by $E^t \leftarrow \frac{k-1}{k} E^t$ and $e^t \leftarrow \frac{k-1}{k} e^t$ for $t = 1, \ldots, k-1$.

Step 4. Solve the L-shaped master problem as in (27-29) to obtain x^{k+1}. Go to 1.

This method differs slightly from the basic method in Higle and Sen [50] in our assuming π_t^k to be optimal dual solutions in each iteration. Higle and Sen allow a restricted set of dual optima that may decrease the solution effort (with perhaps fewer effective cuts).The main convergence result is contained in the following theorem.

Theorem 2 *Assuming complete recourse, $Q(x, \xi) \geq 0$, bounded dual solutions to (34), K_1 and Ξ compact, there exists a subsequence, $\{x^{k_j}\}$, of the iterates of the Stochastic Decomposition Method such that every limit point of $\{x^{k_j}\}$ solves the recourse problem (2S-SLPR).*

Proof: See Higle and Sen [50]. □

One difficult in this basic method is that convergence to an optimum may only occur on a subsequence. To remedy this, Higle and Sen suggest retaining an incumbent solution which changes at low objective function record values. The incumbent is updated each time a sufficient decrease in the kth iteration objective value is obtained. They also show that the sequence of incumbents contains a subsequence with optimal limit points and show how this subsequence can be identified. Various approaches may be used for practical stopping conditions. Their results show good convergence behavior on a variety of problems, especially those arising in network design (see Sen, Doverspike and Cosares [97]).

3.3 Lagrangian-based Approaches

The methods given above are all motivated by the linear formulations, although extensions include nonlinear problems. Lagrangian-based methods, however, have their motivation directly in nonlinear problems and may provide more flexibility than the decomposition approaches. We describe the main approaches below.

Consider the general nonlinear stochastic program given in (4). We can consider a variety of approaches to (4) based on available nonlinear programming methods. For example, we may consider gradient projection, reduced gradient methods, and straightforward penalty type procedures. These methods generally assume, however, that gradients of functions, $E[f_{t+1}(\omega, x^t(\omega), x_{t+1}(\omega))]$ are available and relatively inexpensive to acquire. This is not often the case in stochastic programs since each evaluation may involve solving several problems. Lagrangian approaches have been proposed to avoid this problem by disaggregating the objective terms in (4).

The basic idea behind the lagrangian approaches is that only the nonanticipativity constraints (5) link the scenarios. If these constraints are relaxed, then the solution reduces to separate problems for each realization of ξ. To see how the procedure develops, assume a multiplier, π, (defined on an appropriate dual space as in, for example, Rockafellar and Wets [90]) to obtain a dual problem to (4):

$$\max_{\pi(\xi)} w = \theta(\pi), \tag{50}$$

where

$$\begin{array}{rl} & \theta(\pi) = \\ \inf_{x \in X} z = & \mathrm{E}[\sum_{t=0}^{H} f_{t+1}(\xi, x^t(\xi), x_{t+1}(\xi))] \\ & +E[\sum_{t=0}^{H} \pi^t(\xi)(I - \Pi_t)x^t(\xi)], \end{array} \tag{51}$$

X represents all constraints in $X_t(\xi)$ and π^t corresponds to the first t period components of π.

As an example, we consider the two-stage case with finite realizations of ξ. In this case, if we suppose the x_0 are data in f_1, x_1 becomes x, and x_2 becomes y, then (4) becomes:

$$\min z = \{\sum_{k=1}^{N} p_k(f_1(x_k) + f_2(x_k, y_k)) | x_k - \sum_{k=1}^{N} p_k x_k = 0, k = 1, \ldots, N\}. \tag{52}$$

The dual program is then:

$$\max_{\pi} [\theta(\pi) = \min_{x,y} \sum_{k=1}^{N} p_k[(f_1(x_k) + f_2(x_k, y_k)) + \pi_k \cdot (x_k - \sum_{k=1}^{N} p_k x_k)]]. \tag{53}$$

We state the standard duality result (for a proof, see, e.g., Bazaraa, Sherali and Shetty [3], Theorem 6.2.4) in the finite distribution case in the following theorem.

Theorem 3 *Suppose the stochastic nonlinear program (52) with all functions convex has a finite optimal value, and suppose* $\Omega = \{1, \ldots, K\}$ *with* $P\{\omega = k\} = p_k$*, then* $z \geq w$ *for every feasible* $x, y_1, \ldots, y_K$ *in (52) and* $\pi_1, \ldots, \pi_K$ *feasible in (53), and their optimal values coincide,* $z^* = w^*$*.*

□

We can now describe a basic gradient method for the dual problem. For our purposes, we assume that (53) always has a unique solution.

Lagrangian Dual Ascent Method

Step 0. Set $\pi^0, \nu = 0$ and go to Step 1.

Step 1. Given $\pi = \pi^\nu$ in (53), find a solution be $(x_1^\nu, \ldots, x_K^\nu, y_1^\nu, \ldots, y_K^\nu)$.

Step 2. If $x_k - \sum_{k=1}^N p_k x_k = 0, k = 1, \ldots, N$, stop (optimal). Otherwise, let $\hat{\pi}_k = x_k - \sum_{k=1}^N x_k$ and go to Step 3.

Step 3. Let λ^ν minimize $\theta(\pi^\nu + \lambda\hat{\pi})$ over $\pi^\nu + \lambda\hat{\pi} \geq 0, \lambda \geq 0$. Let $\pi^{\nu+1} = \pi^\nu + \lambda^\nu\hat{\pi}$, $\nu = \nu + 1$, and go to Step 1.

Assuming the unique solution property, this algorithm always produces an ascent direction in θ. The algorithm either converges finitely to an optimal solution or, assuming a bounded set of optima, produces an infinite sequence with all limit points optimal. With multiple optima in (53), some nondifferentiable procedure must be used. In this case, one could consider finding the maximum norm subgradient to be assured of ascent or one could use various bundle type methods (see, e.g., Lemaréchal [65] and Kiwiel [62]).

The basic hope for computational efficiency in the dual ascent procedure is that the number of dual iterations will be small compared to the number of function evaluations that might be required by directly attacking (4). Substantial time may be spent solving (53) but that should be somewhat easier than solving (4) since the linking constraints appear as objective terms instead of hard constraints. Overall, however, this type of procedure generally is slow due especially to a single-point linearization of θ. This observation has led to other types of lagrangian approaches to (4) that use more global or second order information.

Rockafellar and Wets [91] suggest one such procedure for a special case of (52) where f_1 is equivalent to a convex quadratic function over a convex region, and where f_2 is equivalent to a quadratic function subject to linear constraints. Their approach is, however, to restrict the search not just to a single search direction but to allow optimization over a low dimensional set in the dual space. An implementation of their method (Lagrangian Finite Generation Method for Linear-Quadratic Stochastic Linear Programs) is described in King [59]. The method was used successfully to solve practical water management problems concerning Lake Balaton in Hungary (Somlyódy and Wets [99]).

Another method for speeding convergence is to enforce strictly convex objective terms. Rockafellar and Wets discuss methods for adding quadratic terms to esnure positive definite Hessians. In this way, the finite generation method becomes a form of augmented lagrangian procedure.

In a general augmented lagrangian approach (see, e.g., Bertsekas [7]), one adds a penalty $\frac{r}{2}\|x_k - \sum_{k=1}^N p_k x_k\|^2$ to each term k of $\theta(\pi)$ and performs the iterations with a fixed step size such that $\pi_k^{\nu+1} = \pi_k^\nu + r p_k(x_k - \sum_{k=1}^N p_k x_k)$. The advantage (see, e.g., the discussion in Dempster [41]) is that Newton type steps can be applied since the method maintains a nonsingular Hessian. The result should generally be that convergence becomes superlinear in terms of the dual objective without a significantly greater computational burden over pure dual ascent.

The computational experience reported by Dempster suggests that this procedure may be quite efficient with a new variable, x_0, substituting for $\sum_{k=1}^N p_k x_k$ to solve for:

$$\hat{\theta}(\pi) = \min_{x,y} f_1(x_0) + \sum_{k=1}^{N} p_k[f_2(x_k, y_k) + \pi_k^\nu \cdot (x_k - x_0) + \frac{r}{2}\|x_k - x_0\|^2]. \tag{54}$$

Iterations alternate between searches in x_0 and then separable solutions in each x_k, y_k. In this way, the augmented lagrangian approach of solving (54) to find a dual ascent Newton type direction achieves superlinear convergence in dual iterations. The only problem may come in the time to construct the search directions through solutions of (54).

This method is similar to the *Progressive Hedging Algorithm* (PHA) of Rockafellar and Wets [92] which actually achieves a full separation of the separate scenario problems for each iteration, resulting in considerably less work per iteration (but possibly an increased number of iterations). PHA appears to offer many computational advantages at least for structured problems as reported by Mulvey and Vladimirou [74]. A related approach is the extension of the row action algorithm of Censor and Lent [31] by Nielsen and Zenios [77, 78], which has particular efficiencies for network constraints. The key to these methods' success is that individual subproblem structure is maintained throughout the algorithm.

The basic progressive hedging method begins with a nonanticipative solution $\hat{x}^\nu$ and a multiplier π^ν. The nonanticipative (but not necessarily feasible) solution is used in place of x_0 in (54). The first period constraints are also split into each x_k. In this way, we obtain a subproblem:

$$\inf z = \sum_{k=1}^{N} p_k[f_1(x_k) + f_2(x_k, y_k) + \pi_k^\nu \cdot (x_k - \hat{x}^\nu) + \tfrac{r}{2}\|x_k - \hat{x}^\nu\|^2] \tag{55}$$

Now, (55) splits directly into subproblems for each k so these can be treated separately.

Supposing that $(x_k^{\nu+1}, y_k^{\nu+1})$, $k = 1, \ldots, N$, solves (55), we obtain a new nonanticipative decision by taking the expected value of $x^{\nu+1}$ as $\hat{x}^{\nu+1}$ and step in π by $\pi^{\nu+1} = \pi^\nu + (x^{\nu+1} - \hat{x}^{\nu+1})$.

The steps then are simply stated as below.

Progressive Hedging Algorithm

Step 0. Suppose some nonanticipative x^0, some initial multiplier π^0, and $r > 0$. Let $\nu = 0$. Go to 1.

Step 1. Let $(x_k^{\nu+1}, y_k^{\nu+1})$ for $k = 1, \ldots, K$ solve (55). Let $\hat{x}_k^{\nu+1} = \sum_{l=1}^{K} p_l x_l^{\nu+1}$ for all $k = 1, \ldots, K$.

Step 2. Let $\pi_k^{\nu+1} = \pi_k^\nu + r(x_k^{\nu+1} - \hat{x}_k^{\nu+1})$. If $\hat{x}^{\nu+1} = \hat{x}^\nu$ and $\pi^{\nu+1} = \pi^\nu$, then, stop; $\hat{x}^\nu$ and π^ν are optimal. Otherwise, let $\nu = \nu + 1$ and go to 1.

The convergence of this method is based on Rockafellar's proximal point method [89]. The basis for this approach is not dual ascent but the contraction of the pair, $(\hat{x}^{\nu+1}, \pi^{\nu+1})$, about a saddle point. The key is that the algorithm mapping can be described as $(Px^{\nu+1}, \pi^{\nu+1}/r) = (I - T)^{-1}(Px^\nu, \pi^\nu/r)$, where T is a *maximal* *monotone operator* and P is the diagonal matrix of probabilities corresponding to x_k and π_k, i.e, where $P_{(k-1)n_1+i,(k-1)n_1+i} = p_k$ for $i = 1, \ldots, n_1$ and $k = 1, \ldots, K$. The result is the following theorem proven in Rockafellar and Wets [92].

Theorem 4 *The progressive hedging algorithm, applied to (52) with the same conditions as in Theorem 3, converges to an optimal solution pair, (x^*, π^*), (or terminates finitely with an optimal solution) and, at each iteration that does not terminate in Step 2,*

$$\|(P\hat{x}^{\nu+1}, \pi^{\nu+1}/r) - (Px^*, \pi^*/r)\| < \|(P\hat{x}^{\nu}, \pi^{\nu}/r) - (Px^*, \pi^*)/r\|. \tag{56}$$

□

Mulvey and Vladimirou [74] report on comparisons for generalized networks with linear and nonlinear objectives. Their results indicate that PHA solution times were generally better than direct solution with nonlinear codes, such as MINOS (Murtagh and Saunders [75]), and, in most cases, interior point methods. The results were not overwhelmingly positive but they mention the advantage that PHA may have for situations with limited memory and potential for parallel processing.

Nielsen and Zenios [77, 78] report significant improvements with their parallel implementation over serial methods. Their results show single order of magnitude speedups with parallel implementation (which is comparable to serial decomposition results), but they also obtain solutions to larger problems than they could solve serially.

The general structure of the augmented lagrangian methods and PHA allows for a variety of conditions on the objective functions (and implicitly defined constraints). In fact, when the number of scenarios increases in augmented lagrangian formulations, the duality gap for the integer problem actually goes to zero (Birge and Dempster [15]). While the augmented lagrangian can be extended easily in this way to integer variable problems, PHA may not converge in certain integer cases (see Takriti, Birge and Long [101]). In structured power generation scheduling problems, PHA was, however, able to obtain solutions quite efficiently for integer problems despite the lack of guaranteed convergence.

Mulvey and Ruszczyński [73] have also developed a variant of the augmented lagrangian method called *Diagonal Quadratic Approximation* (DQA). Their approach is to write the nonanticipativity constraints with a permutation order, $\sigma_k, k = 1, \ldots, N$, as $x_k - x_{\sigma(k)}, k = 1, \ldots, N$. Then, they approximate the $\|x_k - x_{\sigma(k)}\|^2$ terms in the objective by using a current iterate, $\hat{x}_k$. The result is that the original augmented lagrangian problem then decomposes entirely by k into subproblems:

$$\inf z_k = f_1(x_k) + f_2(x_k, y_k) + (\pi^k - \pi^{\sigma^{-1}(k)}) \cdot (x_k) + \frac{r}{2}[\|x_k - \hat{x}_{\sigma(k)}\|^2 + \|x_k - \hat{x}_{\sigma^{-1}(k)}\|^2], \tag{57}$$

where $\sigma^{-1}(k)$ refers to the scenario j such that $\sigma(j) = k$. This method also allows for significant parallelization. Mulvey and Ruszczyński report results that outperform direct procedures for linear problems but do not appear as efficient as decomposition methods. The DQA method is, however, completely generalizable to nonlinear problems as in standard augmented lagrangian approaches. Its primary advantage may, therefore, be for loosely coupled computing environments to solve complex nonlinear problems that decompose relatively easily.

4 Recent Applications

Stochastic programming has seen great gains since its original applications in airline fleet assignment by Ferguson and Dantzig. Our aim in this section is to show the advantages obtained in solving stochastic problems in place of deterministic models. Examples of these values appear in all sectors of the economy as the following examples demonstrate (see King [60] for other areas).

4.1 Finance

Financial problems inherently involve uncertainty. These problems are especially attractive for stochastic formulations because deterministic approximations are quite obviously flawed. For example, using a deterministic expected return results in investments only in the highest risk (and also highest expected return) asset, but few people are willing to follow such a strategy. The essence of a stochastic program is to develop a hedging strategy that mixes investments in a way that a strictly deterministic approximation cannot reproduce.

As an example, we consider the Russell-Yasuda Kasai Model (Cariño et al. [29]), which won the second prize in the 1993 Franz Edelman Awards for Management Science Achievement. In that model, decisions are made for a Japanese insurance company on how best to invest in assets to meet a random liability stream over time. The investment returns are also random. Additional complications include legal restrictions about the use of income to meet liabilities.

The overall Russell-Yasuda Kasai model has the structure in (MS-SLPR) with a piecewise linear convex objective. A decomposition method was implemented to solve realistically sized models.

For such financial models, deterministic solutions clearly provide inadequate solutions since they ignore risk. Static portfolio models, based on Markowitz's model [71], have, however, been used in many cases. Since these models consider the asset side without liabilities, their implementations may result in significant transaction costs and inopportune liquidation of assets.

The solution of Russell-Yasuda Kasai model improved significantly over the static portfolio procedures that had been used previously. The authors report that the Russell-Yasuda Kasai model yielded additional income of $79 million in its first two years of use.

4.2 Transportation

Transportation models remain a prominent area for application of stochastic programming models. From Dantzig and Ferguson's model, several other areas have developed. An area of particular focus has been in dynamic vehicle allocation (DVA) (see, e.g., Powell [84, 85], Frantzeskakis and Powell [46], Powell and Chung [86])).

The basic problem in DVA is to route a set a vehicles (notably trucks or freight cars) to meet demands along routes and to position them for anticipated future loads. The overall goal is to maximize returns over given horizons. The model again fits the form of (MS-SLPR) but the wide distribution on load demands makes approximation schemes critically important.

Powell and his colleagues have developed a range of approximations that effectively approximate the value function, $\mathcal{Q}_t(x)$, at each time period and yield a form of dynamic programming approximation method. Their results with realistic vehicle fleet sizes of up to 1000 vehicles indicate approximation errors that are often below ten percent. Other experiments (Powell [85]) show that a combination of their methods with a dispatcher's decision making yields 10 to 15% gains over purely manual dispatches.

4.3 Manufacturing

The complexity and inherent uncertainty in both demand and supply aspects of manufacturing make this area also particularly fruitful for stochastic programming models. Eppen et al. [43] provide a particularly interesting example for a capacity planning model at General Motors. The goal was to determine capacity for various products at a number of plants to maximize an expected profit objective with a downside risk constraint. The model fits the form in (2S-SLPR) with the addition of the risk constraint imposing an additional linking constraint across scenarios.

This model was to aid a high level planning decision (in contrast to the operational level vehicle allocation model). A limited number of scenarios was sufficient to represent the distribution on demands that were critical for the decision. The result of their stochastic model was an increase of over 6% in expected profit over two potential deterministic scenarios and complete avoidance of downside risk that occurred in another potential scenario. In this case, no individual deterministic scenario could provide the optimal stochastic programming (hedging) solution.

4.4 Power Systems

Power system models have been a common area of application and source of developments in stochastic programming methods. Pereira and Pinto [81, 82] use decomposition procedures for models of the Brazilian power system which includes a number of reservoirs that must be controlled with uncertain inputs from precipitation as well as uncertain loads and other sources. A similar model, SOCRATES (Jacobs et al. [54]), is used at Pacific Gas and Electric for coordination of their wide range of hydrological power sources.

As an example of the potential of these models, Takriti, Birge, and Long [103] consider a model of the Michigan power system. This model is designed for daily scheduling and, therefore, explicitly includes binary variables for an individual unit's state of operation (on or off). Their procedure uses the progressive hedging method to achieve convergence to near optimal solutions quickly. The results for a sample week showed potential savings over a deterministic procedure of almost $150,000 in generation costs for that week alone.

4.5 Economic Policy

Stochastic programming models have often been used in economic policy considerations as well. For example, Manne [70] considered the US decision on whether to begin or delay a breeder reactor program. The result showed a low cost of waiting for more information that may have influenced the decision to delay.

As another example, Birge and Rosa [24] consider a model of the US economy with random returns on investment in new technologies with reduced carbon emissions. They consider various methods of limiting carbon emissions in the future due to environmental concerns. The stochastic nature of investment in this situation again can have a large effect on the preferred policy. Deterministic models all yield solutions which concentrate investment on a given *winner* technology. The stochastic program solution hedges and produces acceptable results in a variety of circumstances. The overall value of the stochastic solution is as high in this case as 2% of the total economic activity in the U.S.

5 Conclusions and Future Directions

This paper has provided a brief overview of developments in stochastic programming computation. Due to the size and complexity of these problems and the vast range of potential applications, many more developments can be expected in the future. We give some indication of the possibilities here.

Despite computational gains, approximations seem necessary in stochastic programs for the foreseeable future. The key factors to consider here are uses of the structure of the problems. It is particularly useful in practice to distinguish, through problem structure, those uncertain quantities that cannot be ignored from those that might easily be replaced with mean values. The ability to perform such preprocessing steps can be of particular benefit for stochastic programs (see Wallace and Wets [109, 110]). Other aspects of problem structure might be used to determine how sensitive solutions are to changes in the distribution and whether given solutions might be robust (or close to optimal) over a range of distributions.

Optimization routines may benefit particularly from procedures tailored to particular problem structures. For example, discovering where a model might have particularly weak interactions between different parts might allow efficient decomposition across those links. This might also allow for more distributed forms of solution that would not require frequent synchronization.

A particular area for practical stochastic programming developments would be in the area of discrete variable methods. The integer procedures mentioned above have limitations that cannot meet all circumstances. Further integer programming developments within the decomposition schemes would be especially beneficial.

Overall, the greatest need may be in developments relating to practical problem implementations. Some areas that have not been explored or that may have significant potential include problems in real-time control applications and design. Some initial investigations of stochastic programming methods for on-line use (Birge and Wets [27], Takriti [101]) indicate potential but implementation requires quicker, more reliable, and efficient methods. The area of design also has potential (see, e.g., Birge [14]) due to large potential losses and uncertainties in product uses and manufacture. Little progress on truly practical size problems has, however, been made.

Progress continues rapidly in all of these areas and the methods given above. To keep a record of results to date and to provide access to standard problems and examples for testing, a World Wide Web site[3] has been established with instructions for obtaining test problems and for reporting computational results. The results from various methods on the test set problems are updated on the page to provide current information about the various techniques. We encourage everyone with computational experience in stochastic programming to participate by accessing the information for suggestions and by submitting problems and results for others' benefit.

References

[1] K.A. Ariyawansa and D.D. Hudson, 1991. Performance of a benchmark parallel implementation of the Van Slyke and Wets algorithm for two-stage stochastic programs on the Sequent/Balance, *Concurrency Practice and Experience* 3, 109-128.

[2] O. Bahn, O. du Merle, J.-L. Goffin, and J.-P. Vial, 1994. A cutting plane method from analytic centers for stochastic programming, Research Working Paper 94-02-04, Faculty of Management, McGill University, Montreal, PQ, Canada, Original: December 1992, Revised: January 1994.

[3] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, 1993. *Nonlinear Programming: Theory and Algorithms*, Wiley, New York.

[4] E.M.L. Beale, 1955. On minimizing a convex function subject to linear inequalities, *J. Royal Statistical Society, Series B* 17, 173-184.

[5] J. F. Benders, 1962. Partitioning procedures for solving mixed-variables programming problems, *Numerische Mathematik* 4, 238-252.

[6] A.J. Berger, J.M. Mulvey, E. Rothberg, and R.J. Vanderbei, 1995. Solving multistage stochastic programs using tree dissection, Technical Report SOR 92-5, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ.

[3] `http://www.engin.umich.edu/~ dholmes/post.html`

[7] D.P. Bertsekas, 1982. *Constrained Optimization and Lagrange Multiplier Methods,* Academic Press, New York.

[8] D. Bienstock and J.F. Shapiro, 1988. Optimizing resource acquisition decisions by stochastic programming, *Management Science* 34, 215–229.

[9] J.R. Birge, 1980. Solution Methods for Stochastic Dynamic Linear Programs, Ph.D. Dissertation, Stanford University, and Technical Report SOL 80-29, Systems Optimization Laboratory, Stanford University, December 1980.

[10] J.R. Birge, 1982. The value of the stochastic solution in stochastic linear programs with fixed recourse, *Mathematical Programming* 24, 314-325.

[11] J.R. Birge, 1985. Decomposition and partitioning methods for multi–stage stochastic linear programs, *Operations Research* 33, 989-1007.

[12] J.R. Birge, 1988. The relationship between the L-shaped method and dual basis factorization for stochastic linear programming, in: Y. Ermoliev, R. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer, Berlin, pp. 267-272.

[13] J.R. Birge, 1995. Option methods for incorporating risk into linear planning models, Technical Report 95-8, Department of Industrial and Opertions Engineering, University of Michigan, Ann Arbor, MI, May 1995.

[14] J.R. Birge, 1995. Models and model value in stochastic programming, *Annals of Operations Research*, to appear.

[15] J.R. Birge and M.A.H. Dempster, 1995. Stochastic programming approaches to stochastic scheduling, Technical Report 95-12, Department of Industrial and Opertions Engineering, University of Michigan, Ann Arbor, MI, July 1995.

[16] J.R. Birge, C.J. Donohue, D.F. Holmes, and O.G. Svintsiski, 1994. A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs, Technical Report 94-1, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI, also to appear in *Mathematical Programming.*

[17] J.R. Birge, R.M. Freund, and R.J. Vanderbei, 1992. Prior reduced fill-in in the solution of equations in interior point algorithms, *Operations Research Letters* 11, 195-198.

[18] J.R. Birge and D.F. Holmes, 1992. Efficient solution of two-stage stochastic linear programs using interior point methods, *Computational Optimization and Applications* 1, 245-276.

[19] J.R. Birge and A.J. King, 1995. Unbiased methods for sampled cuts in stochastic programming, Technical Report, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

[20] J.R. Birge and F.V. Louveaux, 1988. A multicut algorithm for two-stage stochastic linear programs, *European Journal of Operations Research* 34, 384-392.

[21] J.R. Birge and L. Qi, 1988. Computing block-angular Karmarkar projections with applications to stochastic programming, *Management Science* 34, 1472–1479.

[22] J.R. Birge and L. Qi, 1995. Continuous approximation schemes for stochastic programs, *Annals of Operations Research* 56, 15–38.

[23] J.R. Birge and C.H. Rosa, 1995. Parallel decomposition of large-scale stochastic nonlinear programs, *Annals of Operations Research*, to appear.

[24] J.R. Birge and C.H. Rosa, 1995. Modelling investment uncertainty in the costs of global CO_2 emission policy, *European Journal of Operations Research* 83, 466–488.

[25] J.R. Birge and S.W. Wallace, 1988. A separable piecewise linear upper bound for stochastic linear programs, *SIAM Journal on Control and Optimization* 26, 725-739.

[26] J.R. Birge and R. J–B. Wets, 1986. Designing approximation schemes for stochastic optimization problems, in particular, for stochastic programs with recourse, *Mathematical Programming Study* 27, 54-102.

[27] J.R. Birge and R. J–B. Wets, 1986. On-line solution of linear programming using sublinear functions, Technical Report 86-25, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

[28] J.R. Birge and R. J–B. Wets, 1989. Sublinear upper bounds for stochastic programs with recourse, *Mathematical Programming* 43, 131-149.

[29] D.R. Cariño, T. Kent, D.H. Meyers, C. Stacy, M. Sylvanus, A.L. Turner, K. Watanabe, and W.T. Ziemba, 1994. The Russel- Yasuda Kasai Model: An asset/liability model for a Japanese insurance company using multistage stochastic programming, *Interfaces* 24(1), 29-49.

[30] T. Carpenter, I. Lustig, and J. Mulvey, 1991. Formulating stochastic programs for interior point methods, *Operations Research* 39, 757-770.

[31] Y. Censor and A. Lent, 1981. An iterative row-action method for interval convex programming, *J. Optimization Theory and Applications* 343, 321-353.

[32] A. Charnes and W. W. Cooper, 1959. Chance-constrained programming, *Management Science* 5, 73-79.

[33] I.C. Choi, C.L. Monma, and D.F. Shanno, 1994. Further development of a primal-dual interior point method, *ORSA Journal on Computing* 2, 304–311.

[34] E. Chu, A. George, J. Liu, and E. Ng, 1984. SPARSPAK: Waterloo Sparse Matrix Package User's Guide for SPARSPAK-A, Research Report CS-84-36, Department of Computer Science, University of Waterloo, Waterloo, Ontario.

[35] J. Czyzyk, R. Fourer, and S. Mehrotra, 1993. A study of the augmented system and column-splitting approaches for solving two-stage stochastic linear programs by interior-point methods, Technical Report 93-05, Department of Industrial and Management Sciences, Northwestern University, Evanston, Illinois.

[36] G.B. Dantzig, 1955. Linear programming under uncertainty, *Management Science* 1, 197-206.

[37] G.B. Dantzig and P. Glynn, 1990. Parallel processors for planning under uncertainty, *Annals of Operations Research* 22, 1-21.

[38] G.B. Dantzig and G. Infanger, 1991. Large-scale stochastic linear programs—importance sampling and Benders decomposition, in: C. Brezinski and U. Kulisch, eds., *Computational and applied mathematics, I (Dublin, 1991)*, North-Holland, Amsterdam, pp. 111– 120.

[39] G. B. Dantzig and A. Madansky, 1961. On the solution of two–stage linear programs under uncertainty, in: J. Neyman, ed., *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 1961, pp. 165–176.

[40] G.B. Dantzig and P. Wolfe, 1960. The decomposition principle for linear programs, *Operations Research* 8, 101-111.

[41] M.A.H. Dempster, 1988. On stochastic programming II: dynamic problems under risk, *Stochastics* 25, 15-42.

[42] J. Dupačová and R. J-B Wets, 1988. Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems, *Annals of Statistics* 16, 1517-1549.

[43] G.D. Eppen, R. K. Martin, and L. Schrage, 1989. A scenario approach to capacity planning, *Operations Research* 37, 517-527.

[44] Y. Ermoliev, 1983. Stochastic quasigradient methods and their applications to systems optimization, *Stochastics* 9, 1–36.

[45] A. Ferguson and G. B. Dantzig, 1956. The allocation of aircraft to routes: an example of linear programming under uncertain demands, *Management Science* 3, 45–73.

[46] L. Frantzeskakis and W.B. Powell, 1990. A successive linear approximation procedure for stochastic, dynamic vehicle allocation problems, *Transportation Science* 24, 40–57.

[47] A. Gaivoronski, 1988. Stochastic quasigradient methods and their implementation, in: Y. Ermoliev, R. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, pp. 313-351.

[48] H. I. Gassmann, 1990. MSLiP: A computer code for the multistage stochastic linear programming problem, *Mathematical Programming* 47, 407-423.

[49] A. Geist, A. Beguelin, J. Dongarra, W. Jang, R. Mancheck, and V. Sunderam, 1993. PVM3 User's Guide and Reference Manual, Report ORNL/TM-12187, Oak Ridge National Laboratory, Department of Energy, Oak Ridge, Tennessee.

[50] J. Higle and S. Sen, 1991. Stochastic decomposition: an algorithm for two stage linear programs with recourse, *Mathematics of Operations Research* 16, 650-669.

[51] J.K. Ho and E. Loute, 1981. A set of staircase linear programming test problems, *Mathematical Programming* 20, 245-250.

[52] G. Infanger, 1991. Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs; Extended version: Including results of large-scale problems, Technical Report SOL 91-6, Systems Optimization Laboratory, Department of Operations Reserach, Stanford University, Stanford, CA.

[53] International Business Machines Corp., 1991. Optimization Subroutine Libary Guide and Reference, Release 2, document SC23-0519-02, International Business Machines Corp.

[54] J. Jacobs, G. Freeman, J. Grygier, D. Morton, G. Schultz, K. Staschus, J. Stedinger, and B. Zhang, 1993. Stochastic optimal coordination of river-basin and thermal electric systems (SOCRATES): A system for scheduling hydroelectric generation under uncertainty, Technical Report, Pacific Gas and Electric Company, San Francisco, CA, February 1993.

[55] P. Kall, 1976. *Stochastic Linear Programming*, Springer-Verlag, Berlin.

[56] P. Kall, 1979. Computational methods for solving two-stage stochastic linear programming problems, *Journal of Applied Mathematics and Physics* 30, 261-271.

[57] P. Kall, A. Ruszczyński, and K. Frauendorfer, 1988. Approximations in stochastic programming, in: Y. Ermoliev, R. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, pp. 33-64.

[58] N. Karmarkar, 1984. A new polynomial-time algorithm for linear programming, *Combinatorica* 4, 373-395.

[59] A. King, 1988. Finite generation method, in: Y. Ermoliev, R. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, pp. 295-312.

[60] A. King, 1988. Stochastic programming problems: examples from the literature, in: Y. Ermoliev, R. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, pp. 543-567.

[61] A. King and R. T. Rockafellar, 1993. Asymptotic theory for solutions in generalized M-estimation and stochastic programming, *Mathematics of Operations Research* 18, 148–162.

[62] K.C. Kiwiel, 1983. An aggregate subgradient method for nonsmooth convex minimization, *Mathematical Programming* 27, 320-341.

[63] J. Korycki, 1994. On a distributed implementation of a decomposition method for stochastic optimization problems, Technical Report # 635, Department of Mathematical Sciences, Clemson University, Clemson, SC, August 1994.

[64] G. Laporte and F.V. Louveaux, 1993. The integer L-shaped method for stochastic integer programs with complete recourse, *Operations Research Letters* 13, 133–142.

[65] C. Lemaréchal, 1978. Bundle methods in nonsmooth optimization, in: *Nonsmooth optimization* (Proc. IIASA Workshop), Pergamon, Oxford-Elmsford, N.Y., pp. 79–102.

[66] F.V. Louveaux, 1980. A solution method for multistage stochastic programs with recourse with application to an energy investment problem, *Operations Research* 28, 889-902.

[67] I.J. Lustig, R.E. Marsten, and D.F. Shanno, 1991. Computational experience with a primal-dual interior point method for linear programming, *Linear Algebra and its Application* 152, 191–222.

[68] I.J. Lustig, R.E. Marsten, and D.F. Shanno, 1994. Interior point methods for linear programming: Computational state of the art, *ORSA Journal on Computing* 6, 1–14.

[69] A. Madansky, 1960. Inequalities for stochastic linear programming problems, *Management Science* 6, 197-204.

[70] A. S. Manne, 1974. Waiting for the breeder, in: *Review of Economic Studies Symposium*, 1974, pp. 47-65.

[71] H.M. Markowitz, 1952. Portfolio selection, *Journal of Finance* 7, 77–91.

[72] K. Marti, 1988. *Descent Directions and Efficient Solutions in Discretely Distributed Stochastic Programs,* Lecture Notes in Economics and Mathematical Systems, Vol. 299, Springer–Verlag, Berlin.

[73] J. M. Mulvey, and A. Ruszczyński, 1995. A new scenario decomposition method for large scale stochastic optimization, *Operations Research* 43, 477–490.

[74] J.M. Mulvey and H. Vladimirou, 1991. Applying the progressive hedging algorithm to stochastic generalized networks, *Annals of Operations Research* 31, 399-424.

[75] B.A. Murtagh and M.A. Saunders, 1983. MINOS 5.0 User's Guide, Technical Report SOL 83-20, Systems Optimization Laboratory, Stanford University, Stanford, CA.

[76] J.L. Nazareth and R. J-B.Wets, 1986. Algorithms for stochastic programs: the case of nonstochastic tenders, *Mathematical Programming Study* 28, 1–28.

[77] S.S. Nielsen and S.A. Zenios, 1993. Proximal minimizations with D-functions and the massively parallel solution of linear stochastic network programs, *Int. J. Supercomputing and Applications* 7, 349-364.

[78] S.S. Nielsen and S.A. Zenios, 1993. A massively parallel algorithm for nonlinear stochastic network problems, *Operations Research* 41, 319–337.

[79] M.-C. Noël and Y. Smeers, 1987. Nested decomposition of multistage nonlinear programs with recourse, *Mathematical Programming* 37, 131-152.

[80] R.P. O'Neill, 1976. Nested decomposition of multistage convex programs, *SIAM J. Control and Optimization* 14, 409-418.

[81] M.V.F. Pereira and L.M.V.G. Pinto, 1985. Stochastic optimization of a multireservoir hydroelectric system – a decomposition approach, *Water Resources Research* 21, 779-792.

[82] M.V.F. Pereira and L.M.V.G. Pinto, 1991. Multi-stage stochastic optimization applied to energy planning, *Mathematical Programming* 52, 359-375.

[83] G.Ch. Pflug, 1988. Stepsize rules, stopping times and their implementation in stochastic quasigradient algorithms, in: Y. Ermoliev, R. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, pp. 353–372.

[84] W.B. Powell, 1988. A comparative review of alternative algorithms for the dynamic vehicle allocation program, in: B. Golden, A. Assad, eds., *Vehicle Routing: Methods and Studies,* North-Holland, 1988.

[85] W.B. Powell, 1990. Real-time optimization for truckload motor carriers, *OR/MS Today* 17 (2), 28–33.

[86] W.B. Powell nad R. K.-M. Cheung, 1993. A network recourse decomposition method for dynamic networks with random arc capacities, Technical Report, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, August 1993.

[87] A. Prékopa, 1971. Logarithmic concave measures with application to stochastic programming, *Acta. Sci. Math. (Szeged)* 32, 301–316.

[88] H. Raiffa and R. Schlaifer, 1961. *Applied Statistical Decision Theory*, Harvard University, Boston.

[89] R. T. Rockafellar, 1976. Monotone operators and the proximal point algorithm, *SIAM Journal on Control and Optimization* 14, 877-898.

[90] R.T. Rockafellar and R. J-B. Wets, 1976. Stochastic convex programming: Basic duality, *Pacific Journal of Mathematics* 63, 173-195.

[91] R.T. Rockafellar and R. J-B. Wets, 1986. A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming, *Mathematical Programming Study* 28, 63-93.

[92] R.T. Rockafellar and R. J-B. Wets, 1991. Scenarios and policy aggregation in optimization under uncertainty, *Mathematics of Operations Research* 16, 119-147.

[93] W. Römisch and R. Schultz, 1991. Distribution sensitivity in stochastic programming, *Mathematical Programming* 50, 197-226.

[94] A. Ruszczyński, 1986. A regularized decomposition for minimizing a sum of polyhedral functions, *Mathematical Programming* 35, 309-333.

[95] A. Ruszczyński, 1993. Parallel decomposition of multistage stochastic programing problems, " *Mathematical Programming* 58, 201-228.

[96] A. Ruszczyński, 1993. Regularized decomposition of stochastic programs: algorithmic techniques and numerical results, Working Paper WP-93-21, International Institute of Applied Systems Analysis, Laxenburg, Austria, April 1993.

[97] S. Sen, R.D. Doverspike, and S. Cosares, 1994. Network planning with random demand, *Telecommunications Systems* 3, 11–30.

[98] A. Shapiro, 1991. Asymptotic analysis of stochastic programs, *Annals of Operations Research* 30, 169-186.

[99] L. Somlyódi and R. J-B Wets, 1988. Stochastic optimization models for lake eutrophication management, *Operations Research* 36, 660-681.

[100] B. Strazicky, 1980. Some results concerning an algorithm for the discrete recourse problem, in: M.A.H. Dempster, ed., *Stochastic Programming*, Academic Press, New York.

[101] S. Takriti, 1994. *On-Line Solution of Linear Programs with Varying Right-Hand Sides*, Ph.D. dissertation, University of Michigan, Ann Arbor, Michigan.

[102] S. Takriti, J.R. Birge, and E. Long, 1994. Lagrangian solution techniques and bounds for loosely–coupled mixed-integer stochastic programs, Technical Report, Department of Industrial and Opertions Engineering, University of Michigan, Ann Arbor, MI, December 1994.

[103] S. Takriti, J.R. Birge, and E. Long, 1995. A stochastic model for the unit commitment problem, *IEEE Transactions on Power Systems*, to appear.

[104] R. Van Slyke and R. J-B. Wets, 1969. L–shaped linear programs with application to optimal control and stochastic programming, *SIAM Journal on Applied Mathematics* 17, 638–663.

[105] R.J. Vanderbei, 1992. LOQO user's manual, Technical Report SOR 92-5, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ.

[106] R.J. Vanderbei and T.J. Carpenter, 1993. Syummetric indefinite systems for interior point methods, *Mathematical Programming* 58, 1-32.

[107] A. Wald, 1950. *Statistical Decision Functions*, Wiley, New York.

[108] S.W. Wallace, 1986. Solving stochastic programs with network recourse, *Networks* 16, 295–317.

[109] S.W. Wallace and R. J-B Wets, 1989. Preprocessing in stochastic programming: the case of uncapacitated networks, *ORSA Journal on Computing* 1, 252–270.

[110] S.W. Wallace and R. J-B Wets, 1992. Preprocessing in stochastic programming: the case of linear programs, *ORSA Journal on Computing* 4, 45–59.

[111] R. J–B. Wets, 1974. Stochastic programs with fixed recourse: the equivalent deterministic problem, *SIAM Review* 16, 309–339.

[112] R. J-B Wets, 1983. Solving stochastic programs with simple recourse, *Stochastics* 10, 219-242.

[113] R. J–B Wets, 1988. Large-scale linear programming techniques in stochastic programming, in: Y. Ermoliev, R. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer, Berlin, 1988.

[114] R. J-B Wets, 1990. Stochastic programming, in G.L. Nemhauser, A.H.G. Rinnooy Kan, M.J. Todd, eds., *Optimization*, Handbooks in operations research and management science ; Vol. 1, North-Holland, Amsterdam.

[115] R.J. Wittrock, 1983. Advances in a nested decomposition algorithm for solving staircase linear programs, Technical Report SOL 83-2, Systems Optimization Laboratory, Stanford University, Stanford, CA.

[116] R. Wollmer, 1980. Two stage linear programming under uncertainty with 0-1 integer first stage variables, *Mathematical Programming* 19, 279-288.

[117] D. Yang and S.A. Zenios, 1995. A scalable parallel interior point algorithm for stochastic linear programming and robust optimization, Report 95-07, Department of Public and Business Administration, University of Cyprus, Nicosia, Cyrpsu, February 1995.