A Sublinear Approximation Method for Stochastic Programming

JOHN R. BIRGE
Industrial and Operations Engineering
The University of Michigan

ROGER J-B. WETS
Department of Mathematics
University of California at Davis

Technical Report #86-26

July 1986
A Sublinear Approximation Method for Stochastic Programming

1. Introduction

We consider the following stochastic program with recourse:

\[ \text{find } x \in \mathbb{R}^n, \chi \in \mathbb{R}^m \text{ such that } Ax=b, \ Tx-\chi=0, \ x \geq 0, \quad (1.1) \]

and \( z = cx + \psi(\chi) \) is minimized,

where \( \psi(\chi) = \mathbb{E}[\psi(\chi, \xi)] \) and \( \psi(\chi, \xi) \) is the recourse function defined by

\[ \psi(\chi, \xi) = \inf_{y \in \mathbb{R}_{+}^m} \{ qy \mid Wy = \xi - \chi \}. \quad (1.2) \]

The random \( m_2 \)-vector \( \xi \) is defined on a probability space \( (\mathbb{E}, \mathcal{F}, P) \). The vectors \( b \in \mathbb{R}^m, \ c \in \mathbb{R}^n, \ q \in \mathbb{R}^m, \ A \in \mathbb{R}^{m \times m}, \ T \in \mathbb{R}^{m \times n}, \ W \in \mathbb{R}^{m \times n}, \) are deterministic. In more general models, \( q \) and \( T \) may be stochastic, but we confine our attention to the case of random right-hand sides in (1.2).

Difficulties in performing the multiple integration to evaluate \( \psi(\chi) \) make the solution of (1.1) especially complex. Solution procedures that do not assume special structure, therefore, involve some approximation to \( \psi(\chi) \) and its derivatives. One type of procedure is to sample from \( \xi \) randomly and to use sample information to guide an optimization algorithm. These stochastic quasi-gradient methods (see Ermoliev [1983] and Ermoliev and Gaivoronski [1986]) have asymptotic convergence properties, but they are limited by a lack of computable bounds. Other approximation procedures for (1.1) generally rely on discretizations of \( \xi \) (Hwang, Ziemba, and Ben-Tal [1977], Kall and Stoyan [1982], Birge and Wets [1986a]). These approaches are limited by the size of the resulting linear programs.

Various methods have been suggested for solving these large-scale linear programs (Wets [1986]). These methods include basis factorization (Strazicky [1980], Kall [1979]), inner linearization (Dantzig and Madansky [1961]), and
A Sublinear Approximation Method for Stochastic Programming

John R. Birge*  
Industrial and Operations Engineering  
University of Michigan  
Ann Arbor, MI 48109

Roger J-B. Wets*  
Mathematics  
University of California  
Davis, CA 95616

Abstract:

Separable sublinear functions are used to approximate the recourse function of a stochastic program. The resulting problem's objective involves the inf-convolution of convex functions. A dual of this problem is formulated to obtain an implementable solution procedure. The resulting convex program only requires the evaluation of line integrals. Computational results indicate that it provides good quick bounds on the stochastic program objective value.

Keywords: stochastic programming, sublinear function, simple recourse problem, recourse model, duality, approximation.

*This research has been partially supported by the National Science Foundation.
outer linearization (Van Slyke and Wets [1969]). Implementations are described in Kall and Keller [1983] and Birge [1985, 1986]. In general, the number of realizations in the discretization is limited to numbers in the hundreds. This discretization level is not always enough to provide close bounds on the objective value. Solution times can also become extremely long. The situation is better, however, if (1.1) has special structure.

The stochastic program with simple recourse is a special structure instance of (1.1) in which $W = (I, -I)$, where $I$ is an $m_x \times m_x$ identity matrix. This allows for the use of general nonlinear optimization procedures (Nazareth and Wets [1986]) and combinations of linear and nonlinear procedures (Qi [1986]). Separability reduces the $m_x$-dimensional integration to $m_x$ independent one-dimensional integrals; it renders possible the direct computation of the values and the derivatives of $\pi$ for use in optimization.

Our approach is to approximate the recourse function by functions similar to simple recourse functions. We generalize and extend the ray function approximation in Birge and Wets [1986a]. The result is a separable function that allows a type of monotropic optimization (Rockafellar [1984]). It also only requires line integration in its evaluation. The convex hull of several of these functions is used in the optimization procedure. In our experience, the solution of this approximating problem provides quick solutions that are close in value to the optimum. It may be especially valuable in providing initial solution values (as in Birge and Wets [1984]) that may be used in further optimization procedures (Nazareth and Wets [1986]).

The simple recourse problem and its properties are described in Section 2. Section 3 describes the sublinear approximation procedure and its relation to the simple recourse problem. Section 4 explains how the sublinear approximation is solved using its dual, and Section 5 discusses our
implementation. Section 6 presents computational results. Section 7 discusses possible extensions.

2. The Simple Recourse Problem

The simple recourse problem has a separable recourse function that is written as \( \psi(x) = \sum_{i=1}^{m_2} \psi_i(x_i) \) where \( \psi_i(x_i) = \int \psi_i(x_i, \xi_i) P_i(d\xi_i) \),

\( P_i \) is the marginal distribution function of \( \xi_i \) and

\[
\psi_i(x_i, \xi_i) = \inf \{ q_i^+ y_i^+ + q_i^- y_i^- \mid y_i^+ - y_i^- = \xi_i - x_i, y_i^+ \geq 0, y_i^- \geq 0 \} \tag{2.1}
\]

\[
= \begin{cases} 
q_i^+(\xi_i - x_i) & \text{if } x_i \leq \xi_i, \\
q_i^-(x_i - \xi_i) & \text{if } x_i \geq \xi_i.
\end{cases}
\]

Detailed properties of the simple recourse problem are given in Wets [1966, 1974a] and Parikh [1968].

The simple recourse function components \( \psi_i(x_i) \) can then be written as

\[
\psi_i(x_i) = q_i^+ x_i - q_i^- x_i + q_i^+ x_i P_i(x_i) - q_i^+ \int_{-\infty}^{x_i} \xi_i P_i(d\xi_i), \tag{2.2}
\]

where \( q_i = q_i^+ + q_i^- \). The functions \( x \mapsto \psi_i(x) \) and \( x_i \mapsto \psi_i(x_i) \) are continuous convex functions on \( \mathbb{R}^{m_2} \) and \( \mathbb{R} \), respectively (Wets [1974a]). The subdifferential at \( x_i \) is given by

\[
\partial \psi_i(x_i) = \{ \pi \mid -q_i^- + q_i^+ \leq \pi \leq -q_i^- + q_i^+ P_i(x_i) \}, \tag{2.3}
\]

where \( P_i^+(x) = \lim_{y \uparrow x} P_i(y) \) (Wets [1974a]). From (2.3) if \( P_i \) is continuous, then \( \psi_i \) is differentiable.

The development of our sublinear approximation relies upon the use of conjugate functions. To find the conjugate of \( \psi_i \), first define

\[
G_i(x) = \{ y \mid P_i^+(y) \leq x \leq P_i(y) \}. \tag{2.4}
\]

The conjugate of \( \psi_i \) is written

\[
\psi_i^*(v_i) = \sup_{x_i} \{ v_i x_i - \psi_i(x_i) \}. \tag{2.5}
\]
PROPOSITION 2.1. The conjugate function of \( \Psi_i \) defined in (2.1) is given by

\[
\Psi_i^*(v_1) = \begin{cases} 
-q_i^+ \xi_1 + (v_1 + q_i^+) y - q_i^+ P_i(y) \\
+q_i^+ \int_1^y \xi_1 P_i(d\xi_1), \text{ if } -q_i^+ \leq v_1 \leq q_i^+ \\
\text{otherwise,}
\end{cases}
\]

where \( y \in G_i\left(\frac{v_1 + q_i^+}{q_i} \right) \).

Proof. From (2.5) and (2.1), we have

\[
\Psi_i^*(v_1) = \sup_{x_1} \left\{ v_1 x_1 - q_i^+ \xi_1 + q_i^+ x_1 - q_i x_1 P_i(x_1) \right\}
+q_i^+ \int_1^y \xi_1 P_i(d\xi_1)
-\]

\[
= -q_i^+ \xi_1 + \sup \left\{ (v_1 + q_i^+) x_1 - q_i x_1 P_i(x_1) \right\}
+q_i^+ \int_1^y \xi_1 P_i(d\xi_1)
-\]

\[
= \sup \left\{ (v_1 + q_i^+) x_1 - q_i x_1 P_i(x_1) \right\}. \tag{2.7}
\]

From (2.7), it is clear that \( \Psi_i^*(v_1) = \leftrightarrow \) if \( v_1 < -q_i^+ \) or \( v_1 > q_i^+ \). We, therefore, assume that \( -q_i^+ \leq v_1 \leq q_i^+ \). Using (2.3), we find the set of \( x_1 \) that attains the supremum in (2.7) is \( \left\{ x_1 \mid P_i(x_1) \leq \frac{v_1 + q_i^+}{q_i} \leq P_i(x_1) \right\} = G_i\left(\frac{v_1 + q_i^+}{q_i} \right) \). Substitution of \( y \in G_i\left(\frac{v_1 + q_i^+}{q_i} \right) \) yields the formula in (2.6). \( \hspace{1cm} \blacksquare \)

COROLLARY 2.2. If \( P_i \) is continuous, then

\[
\Psi_i^*(v_1) = \begin{cases} 
-q_i^+ \xi_1 + q_i^+ \int_1^y \xi_1 P_i(d\xi_1), \text{ if } -q_i^+ \leq v_1 \leq q_i^+ \\
\text{otherwise,}
\end{cases}
\]

\[
= \sup \left\{ \frac{v_1 + q_i^+}{q_i} \right\}. \tag{2.8}
\]

Proof: If \( P_i \) is continuous, then, for all \( y \in G_i\left(\frac{v_1 + q_i^+}{q_i} \right) \), \( P_i(y) = \frac{v_1 + q_i^+}{q_i} \).

Substituting this into (2.6) yields the result. \( \hspace{1cm} \blacksquare \)

5
COROLLARY 2.3. If \( \xi_1 \) has the degenerate distribution

\[
P_1(\xi_1) = \begin{cases} 
1 & \xi_1 \geq \xi_1, \\
0 & \text{otherwise},
\end{cases}
\]  

(2.9)

then

\[
\psi_{1*}(v_1) = \begin{cases} 
v_1/\xi_1 & \text{if } -q_1^- \leq v_1 \leq q_1^-, \\
v_1 + q_1^+ \over q_1 & \text{otherwise}.
\end{cases}
\]  

(2.10)

Proof: Note that \( G_1(-\ldots-) = \xi_1 \) if \(-q_1^- \leq v_1 \leq q_1^-\). Substitution in (2.6) again yields the result. \( \blacksquare \)

In addition to \( \psi_{1*} \), we are also interested in its subgradients.

PROPOSITION 2.4. The subdifferential of \( \psi_{1*} \) at \( v_1 \in \text{ri}(\text{dom } \psi_{1*}) \) (the relative interior of the effective domain of \( \psi_{1*} \)) is given by

\[
\partial \psi_{1*}(v_1) = G_1(-\ldots-) \cdot v_1 + q_1^+ \over q_1 .
\]  

(2.11)

Proof: The subgradient may be found directly using (2.8). We can also use Theorem 23.5 from Rockafellar [1970]. If \( v_1 \in \partial \psi_{1*}(y) \), then \( y \in \partial \psi_{1*}(v_1) \).

From (2.3), \( v_1 \in \partial \psi_{1*}(y) \) implies

\[-q_1^- + q_1^- P_1^-(y) \leq v_1 \leq -q_1^- + q_1^- P_1(y),
\]

or

\[
P_1^-(y) \leq v_1 + q_1^+ \over q_1 \leq P_1(y).
\]

Equation 2.11 follows using (2.4). \( \blacksquare \)

3. The Sublinear Approximation

Simple recourse functions are used here to approximate the general recourse function function \( \psi \) as defined in (1.2). We first note that the function \( \phi \) defined by

\[
\phi(\xi - \chi) = \psi(\chi, \xi)
\]  

(3.1)

is sublinear (positively homogeneous and convex). This property allows the
simple recourse function approximation. The function $\phi$ is also polyhedral
(i.e., its epigraph is a polyhedral cone).

Birge and Wets [1986a] introduced a method for approximating $\psi$ by simple
recourse functions. This method was based on solving the linear program:

$$\text{find } y \in \mathbb{R}^n \text{ such that } Wy = e_i, \ y \geq 0,$$

and $qy$ is minimized,

where $e_i$ is the $i$th unit $m_i$-vector. The optimal solution value of (3.2) is
$q_I^+$. If we substitute $-e_i$ for $e_i$, the optimal solution value is $-q_I^-$. By sublinearity,

$$\psi(x, \xi) \leq \psi_I(x, \xi) = \sum_{i=1}^{m_2} \psi_{I(i)}(x_1, \xi_1),$$

where

$$\psi_{I(i)}(x_1, \xi_1) = \begin{cases} q_{I(i)}^+(\xi_1 - x_1), & \xi_1 \geq x_1 \\ q_{I(i)}^-(\xi_1 - x_1), & \xi_1 < x_1. \end{cases}$$

The function $\psi_I$ is a simple recourse function.

By integration in (3.3), we have

$$\psi(x) \leq \psi_I(x),$$

where

$$\psi_I(x) = \int_{\Xi} \psi_{i}(x, \xi) P(d\xi)$$

$$= \sum_{i=1}^{m_2} \left\{ \int_{\xi_1 \geq x_1} q_{I(i)}^+(\xi_1 - x_1) P(d\xi_1) \\ + \int_{\xi_1 < x_1} q_{I(i)}^-(\xi_1 - x_1) P(d\xi_1) \right\}.$$  

Note that $\psi_I$ is separable in the components of $x$ and only line integration is
required in its computation.

Other approximations are obtained by considering directions other than
$\pm e_i$ in (3.2). Let $h_1, \ldots, h_K \in \mathbb{R}^2$ positively span $\mathbb{R}^2$ (i.e., $\mathbb{R}^2 = \text{pos}[h_1, \ldots, h_K]$). Substitute $h_j, j=1, \ldots, K$, for $e_i$ in (3.2) and let the
optimal solutions be \( q_{H(j)}, j = 1, \ldots, K \). We then have
\[
\psi(x, \xi) \leq \inf \{ \sum_{j=1}^{K} \lambda_j q_{H(j)} \mid \sum_{j=1}^{K} \lambda_j h_j = \xi - x, \lambda_j \geq 0, j = 1, \ldots, K \}. \tag{3.5}
\]
If \( H = [h_1, \ldots, h_K] \) includes all of the columns of \( W \), then (3.5) becomes an equality (see Birge and Wets [1986b]).

A difficulty in using (3.5) for approximating \( \psi \) is that in general an optimization must still be performed inside the integral. This solution is immediate, however, if \( H \) is a minimal set, i.e., a positive linear basis for \( \mathbb{R}^{m_2} \). A convenient choice for such a set is to use a linear basis \( D = [d_1, \ldots, d_{m_2}] \) for \( \mathbb{R}^{m_2} \) and \( -D = [-d_1, \ldots, -d_{m_2}] \), so that \( \text{pos} [D, -D] = \mathbb{R}^{m_2} \). In this case, \( \xi - x = \sum_{i=1}^{m_2} (D^{-1})_i (\xi - x) d_i \). The approximation is
\[
\psi(x, \xi) \leq \psi_D(x, \xi) = \sum_{i=1}^{m_2} \psi_D(i)(x, \xi) \tag{3.6}
\]
where
\[
\psi_D(i)(x, \xi) = \begin{cases} q_D^+(i) (D^{-1})_i (\xi - x) & \text{if } (D^{-1})_i (\xi - x) \geq 0, \\ q_D^-(i) (D^{-1})_i (x - \xi) & \text{if } (D^{-1})_i (x - \xi) < 0, \end{cases} \tag{3.7}
\]
and \( q_D^+(i) \) and \( q_D^-(i) \) are the optimal solution values of (3.2) with \( d_i \) and \( -d_i \), respectively, substituted for \( e_i \) and \( -e_i \).

Several different bases are used in our approximation of \( \psi \). For \( D = \{D^1, \ldots, D^L\} \), a set for linear bases for \( \mathbb{R}^{m_2} \),
\[
\psi(x, \xi) \leq \inf \{ \sum_{j=1}^{L} \lambda_j \psi_j(x, \xi) \mid \sum_{j=1}^{L} \lambda_j \xi_j = \xi, \sum_{j=1}^{L} \lambda_j x_j = x, \lambda_j \geq 0, \text{ for all } j \}, \tag{3.8}
\]
the function obtained by taking the convex hull of the epigraphs of \( \psi_Dj \), \( j = 1, \ldots, L \). Of course, if \( D \) includes all linear bases in \( W \), then (3.8) is satisfied as an equality. Although it might be simpler than (3.5), inequality (3.8) is still difficult to use computationally, again because of
the minimization required inside the integral. A weaker, but usable
inequality is obtained by reversing the infimum and integration. For this, we
define
\[
\Psi_D(x) = \frac{m_2}{m_1} \sum_{i=1}^{m_2} \Psi_{D(i)}(x), \text{ where}
\]
\[
\Psi_D(x) = \left\{ \begin{array}{ll}
q_D^+ (D^{-1})_{i,1} (\xi-x) P(d\xi) \\
(D^{-1})_{i,1} (\xi-x) \geq 0 \\
q_D^- (D^{-1})_{i,1} (\xi-x) P(d\xi) . \\
(D^{-1})_{i,1} (\xi-x) < 0.
\end{array} \right.
\]  
(3.9)

**Proposition 3.1.** Let \( D \) be a set of linear bases of \( \mathbb{R}^{m_2} \), then
\[
\psi(x) \leq \inf \{ \Psi_D(x), D \in D \}.
\]  
(3.10)

**Proof:** Integration of (3.6) yields
\[
\Psi(x) \leq \Psi_D(x)
\]  
(3.11)

for any \( D \in D \). So, \( \int \lambda^i \Psi(x^i) \geq \int \lambda^i \Psi_D(x^i) \), for any \( \lambda^i \geq 0 \). Letting \( \sum \lambda^i = 1 \) and \( x = \sum \lambda^i x^i \) yields (3.10) by the convexity of \( \Psi \).

Equality in (3.10) can only be guaranteed in very special cases, even if \( D \) includes all linear bases from \( W \). To see this simply observe that if \( D \) is rich enough, then
\[
\int \inf \Psi_D = \int \inf \Psi_D \geq \inf \Psi_D
\]

with strict inequality except in degenerate cases such as: \( D \) is a singleton, the \( \Psi_D \)'s are linear, the probability measure is degenerate, etc. But as suggested by Proposition 3.1, \( \inf \{ \Psi_D(x), D \in D \} \) always provide us with an upper bound, that is relatively easy to compute, even when other procedures might fail, see below. We have also observed that the solution obtained from solving the "stochastic" program

\[
\text{find } x \in \mathbb{R}^{n_1}, \chi \in \mathbb{R}^{m_2} \text{ such that } Ax = b, \ Tx = \chi, \ x \geq 0,
\]

and \( z = cx + \inf \{ \Psi_D, D \in D \}(\chi) \) is minimized,
instead of (1.1), is usually a very good approximation of the optimal solution, much better than may be expected from the relatively lax inequality (3.10). Our experience shows that the function \( \text{co} \{ \psi_D, D \in D \} \) is "parallel" to \( \psi \), i.e.,

\[
\psi(x) = \text{co} \{ \psi_D(x), D \in D \}.
\] (3.12)

The following argument, heuristic in nature, does give some justification for this observation. Suppose \((x^0, \chi^0)\) is a feasible solution of (1.1) that is near the optimal solution. Suppose \(D\) contains all bases (in \(W\)) that correspond to basic optimal solutions, for any possible pair \((\xi, \chi)\), of the linear program that defines \(\psi(\chi, \xi)\), cf. Wets [1974b] for the Basis Decomposition Theorem. Also, suppose that for some \(D_0 \in D\),

\[
\text{co} \{ \psi_D, D \in D \} (x^0) = \psi_{D_0}(x^0).
\]

Then

\[
\psi(\text{co} \psi_D)(x^0) = \psi_{D_0}(x^0)
\]

\[
= \int \psi_{D_0}(\chi^0, \xi) P(d\xi)
\] (3.13)

Observe that \(\psi_{D_0}(\chi^0, \xi) = -q^+_{D_0}\) if \(\xi \in \Xi_0\), where

\[
\Xi_0 = \{ \xi \mid D^{-1}_0 \xi \geq D^{-1}_0 x^0 \}.
\]

Since \(x^0\) is nearly optimal, one can reasonably expect to have \(\Xi_0\), or more generally the region surrounding \(\Xi_0\), contains most of the probability. Hence the subgradients of \(\psi_{D_0}\) at \(x^0\) are approximated by \(-q^+_{D_0}\). On the other end (Section 7, Wets [1974b]), excluding possibly some boundary cases,

\[
\psi(x^0) = \int \psi(\chi^0, \xi) P(d\xi)
\]

\[
= \sum_{D \in D_0} \psi_D P[D^{-1}_0 \xi \geq D^{-1}_0 x^0]
\] (3.14)
where \( \pi_D \) are the multipliers associated with the basis \( D \). Most of the probability mass lying in \( \xi_0 \) or its neighborhood will result in having \( \partial \psi(x^0) \) reasonably well approximated by \( -\pi_D \). But from the construction of the \( \{ q_D^+, q_D^- \} \) it follows that \( q_D^+ = \pi_D \). This would imply that both \( \partial \phi_0(\psi_D) \) and \( \partial \psi \) at \( x^0 \) are near \( -\pi_D \).

As an example, consider the stochastic program:

\[
\text{find } x, y_1^+, y_1^-, y_2^+, y_2^-, y_3 \geq 0 \text{ such that } \quad \begin{align*}
    x & \leq 1 \\
    t_1 x & + y_1^+ - y_1^- + y_3 = \xi_1 \\
    t_2 x & + y_2^+ - y_2^- + y_3 = \xi_2
\end{align*}
\]

(3.15)

to minimize \( z = cx + E_\xi [y_1^+ y_1^- y_2^+ y_2^- y_3] \).

where \( \xi_i \) is uniformly distributed on \([0,1]\), for \( i=1,2 \). The regions for the optimal bases of the recourse problem are shown in Figure 1. We use three bases \( D^1, D^2, D^3 \) to develop approximating functions \( \psi_D^1, \psi_D^2 \) and \( \psi_D^3 \), where

\[
D^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D^2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad D^3 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.
\]

Figure 1. Optimal Subproblem Basis Regions
A comparison of the three approximations and the exact value of $\psi$ is given in Figure 2, where $\chi_2 = 0.1$. Note that the slope of the convex hull of the approximating functions closely approximates the slope of $\psi$.

Figure 2. Comparison of $\psi$ Values

4. Dualization and Solution Procedures

Several questions must be answered in order to solve (1.1) with $\text{co}(\psi_D | D \in D)$ substituted for $\psi$. The first concern is that finding the convex hull of a set of functions is itself a difficult task. The second problem is to evaluate $\psi_D$ efficiently. A third area involves the choice of $D$. We address each of these problems in this section.
After substituting for \( v \), (4.1) becomes:

\[
\text{find } x \in \mathbb{R}^n, \; z \in \mathbb{R}^m \text{ such that } Ax = b, \; Tx - z = 0, \; x \geq 0, \tag{4.1}
\]

and \( z = cx + \sup_{D \in \mathcal{D}} \{ \psi_D(x) \mid D \in \mathcal{D} \} \) is minimized.

Instead of solving (4.1), we consider a dual program to (4.1). The dual program has a computational advantage because the convex hull operation is replaced by a supremum.

**PROPOSITION 4.1.** A dual program to (4.1) is given by

\[
\text{find } \pi \in \mathbb{R}^n, \; \omega \in \mathbb{R}^m \text{ such that } \pi A + \omega T \leq c \text{ and } \omega = \sup_{D \in \mathcal{D}} \{ \psi_D^*(x) \mid D \in \mathcal{D} \} \tag{4.2}
\]

where \( \psi_D^* \) is the conjugate function of \( \psi \) and where the optimal value of (4.2), \( w^* = z^* \), the optimal value of (4.1).

**Proof:** A general dual of (4.1) (see, e.g., Rockafellar [1974], Geoffrion [1971]) can be written as

\[
\begin{align*}
\text{Max } & \{ \inf_{x, \omega} cx + g(x) + \omega (b - Ax) + \pi (x - Tx) \} \tag{4.3} \\
& \pi, b \quad x \geq 0, \; x
\end{align*}
\]

where \( g(x) = \text{co}\{ \psi_D(x) \mid D \in \mathcal{D} \} \). Since (4.1) involves linear constraints, the optimal values of (4.3) and (4.1) are the same. We can rewrite (4.3) as

\[
\begin{align*}
\text{Max } & \{ \inf_{x, \omega} (c - \pi A - \pi T)x - (\pi x - g(x)) + \omega b \}, \tag{4.4} \\
& \pi, \omega \quad x \geq 0, \; x
\end{align*}
\]

which is equivalent to

\[
\text{find } \pi \in \mathbb{R}^n, \; \omega \in \mathbb{R}^m \text{ such that } \pi A + \omega T \leq c \text{ and } \omega = \sup_{D \in \mathcal{D}} \{ \psi_D^*(x) \mid D \in \mathcal{D} \} \tag{4.5}
\]

By Theorem 16.5 of Rockafellar [1970],

\[
(\text{co} \{ \psi_D(x) \mid D \in \mathcal{D} \})^* = \sup \{ \psi_D^* \mid D \in \mathcal{D} \}, \tag{4.6}
\]

yielding (4.2).

A solution of the dual program (4.2) is simpler to compute than a solution of (4.1) because \( g^*(-\pi) \) is much easier to evaluate than \( g(x) \). We
must, however, obtain an expression for \( \Psi^*_D \). First, let \( P_D(1) \) be the distribution function of \( \xi_i = (D^{-1})_i (x - \chi) \) and let \( \xi_1 = \int \xi_1 P_D(1)(\xi_1) \). Also define

\[
G_D(1)(x) = \{ y \mid P_D(1)(y) \leq x \leq P_D(1)(y) \}.
\]

**PROPOSITION 4.2.** The conjugate function of \( \Psi^*_D \) is given by

\[
\Psi^*_D(v) = \sum_{i=1}^{m_2} \Psi^*_D(1)(v),
\]

where

\[
\Psi^*_D(1)(v) = \begin{cases} 
- q^+_D(1) \xi_1 + (vD) \xi_1 q^+_D y - q^-_D(1) y P_D(1)(y) \\
+ q^+_D y \xi_1 P_D(1)(d\xi_1), 
\end{cases}
\]

if \( - q^+_D(1) \leq vD \leq q^-_D(1) \),

otherwise,

where \( q_D(1) = q^+_D(1) + q^-_D(1) \) and \( y \in G_D(1) \left( \frac{vD - \xi_1}{q_D(1)} \right) \).

**Proof:** From (3.9), observe that

\[
\Psi_D(1)(x) = \int \Psi^S_1 ((D^{-1})_1, \xi_1) P_D(1)(d\xi_1),
\]

where \( \Psi^S_1 \) is a simple recourse function as in (2.1) with \( q_D^+ \) and \( q_D^- \) substituted for \( q_1^+ \) and \( q_1^- \) respectively. From (4.9), it follows that

\[
\Psi_D(x) = \sum_{i=1}^{m_2} \Psi^S_1 ((D^{-1})_1) = \Psi^S(D^{-1}x),
\]

where \( \Psi^S_1 \) has the form of the simple recourse function in (2.2) and \( \Psi^S \) is the simple recourse function defined for the random variables \( \xi_1, \ldots, \xi_{m_2} \) in place of \( \xi_1, \ldots, \xi_{m_2} \).

The dual of \( \Psi_D \) is then (see Theorem 16.3 Rockafellar [1970])

\[
\Psi^*_D = (\Psi^S D^{-1})^* = D^{-1}(\Psi^S)^*,
\]

where

\[
D^{-1}(\Psi^S)^*(v) = (\Psi^S)^*(vD).
\]

Applying (4.11) and (4.12) to (2.6) yields the result.

A subgradient of \( \Psi^*_D \) is used in the optimization procedure. It is also calculated from the subdifferential of \( (\Psi^S)^* \).
PROPOSITION 4.3. The subdifferential of $\Psi_D^*$ at $v \in \text{int}(\text{dom} \Psi_D^*)$ is

$$\partial(\Psi_D^*)(v) = \{ y \in \mathbb{R}^{m_2^*}, y_i \in G_D, i = 1, \ldots, m_2 \} \ \text{(4.13)}$$

Proof: If $w^*$ is a subgradient of $(\Psi^S)^*$ at $w$, then

$$(z - w) \cdot w^* + (\Psi^S)^*(w) \leq (\Psi^S)^*(z). \ \text{(4.14)}$$

From (4.11) and (4.12) if $z = wD$ and $w = vD$, (4.14) becomes

$$(v - u) Dw^* + \Psi_D^*(v) \leq \Psi_D^*(u).$$

From Proposition 2.4, $w^* = (w_1^*, \ldots, w_{m_2}^*)$ where $w_i^* \in G_i$ (---), so $Dy$ in (4.13) is a subgradient of $\Psi_D^*$ at $v$. A reverse argument shows that any subgradient of $\Psi_D^*$ has the form $Dy$ in (4.12), proving the result. ■

The expressions for $\Psi_D^*$ and $\partial \Psi_D^*$ in (4.7-8) and (4.13) are used in an optimization procedure for (4.2). A difficulty is that, even when each $\Psi_D^*$ is differentiable (i.e., the distribution function of $\xi_i$ is strictly increasing on its support, but not necessarily continuous), the objective function in (4.2) is not necessarily differentiable. Nondifferentiable methods can be applied to this program. One possibility is to use a minimum norm subgradient as a gradient (see Wolfe [1975] and Nazareth and Wets [1986]). This approach is applicable when $\Psi_D^*$ is differentiable because the subdifferential of $g^*(\pi)$

$$\text{sup}_{D \in D'} \Psi_D^*(\pi)$$

is

$$\partial g^*(\pi) = \text{co}\left(\Psi_D^*(\pi) \mid D \in D'\right), \ \text{(4.15)}$$

where $D' = \{ D \in D \mid \Psi_D^*(\pi) = g^*(\pi) \}$. The set in (4.15) is then polyhedral, so the vector $v$ which minimizes $||v||$ over $v \in \partial g^*(\pi)$ can be found easily.

When each $\Psi_D^*$ is differentiable, we use an alternative to these nondifferentiable methods. We transform (4.2) into a smooth optimization problem with nonlinear constraints (as suggested, for example, in Gill, Murray, and Wright [1981]). The new problem becomes
\[ \begin{align*}
\text{find } & \sigma \in \mathbb{R}^{m_1}, \pi \in \mathbb{R}^{m_2}, \theta \in \mathbb{R} \text{ such that} \\
& \sigma^a + \pi^T \leq c \\
& \nu^*_D(-\pi) \leq \theta, \text{ for all } D \in \mathcal{D}, \text{ and} \\
& w = \sigma b - \theta \text{ is maximized.}
\end{align*} \quad (4.16) \]

General methods for optimization problems with nonlinear constraints can be applied to (4.16). In our examples, we use the MINOS computer code (Murtagh and Saunders [1980]) to solve (4.16).

5. Implementation Considerations

A. Distribution Approximation

In solving (4.2) or (4.16) one needs to find \( P_D(1) \), the distribution function of \( \xi_i \), and \( G_D(1) \), the inverse function. If the random variables \( \xi_j \) are normally distributed with means, \( \mu_j \), and variances, \( \sigma_j^2 \), then \( \xi_i = (D^{-1})_i \). \( \xi - \chi \) is also normally distributed with mean \( \Sigma \sum_{j=1}^{m_2} (D^{-1})_{ij} (\mu_j - \chi_j) \) and variance \( \Sigma \sum_{j=1}^{m_2} (D^{-1})_{ij}^2 \sigma_j^2 \). (Note that degenerate random variables can also be included among the \( \xi_i \) with \( \xi_i \) remaining normally distributed.) Other special cases are possible, such as when \( 0 < (D^{-1})_{ij} \mu_j = (D^{-1})_{ik} u_k \) for all \( j \) and \( k \) where the \( \xi_j \) are exponentially distributed. In this case, \( (D^{-1})_i \xi \) would have an Erlangian distribution of order \( m_z \) with mean, \( m_z (D^{-1})_{ij} \mu_j \). In general, however, the distribution of \( \xi_i \) must be approximated.

In our implementation, we approximate \( \xi_i \) with a normal distribution by fitting the mean and variance of the random variable. This approach is, of course, exact if each \( \xi_i \) is normally distributed. It is also an appropriate approximation for many types of unimodal distributions and allows for rapid computation. Alternative approaches are discussed in the Appendix.

B. Basis Choice

Given a problem form \((4.2) \text{ or } (4.16)\) and a method for finding \( P_D(1) \), the set of bases \( \mathcal{D} \) must still be chosen. For the problem of approximating \( \psi \), it
appears that the matrices to include in $D$ should be chosen so that the level sets of $\Psi_D$ cover high probability regions of the level sets of $\psi$. This coverage, however, depends on $\chi$, so the choices should be good for a range of values of $\chi$ (that would ideally include the optimal value of $\chi$). The use of the identity as a starting basis appears to have good coverage, especially when the optimal $\chi$ is close to $\xi$.

We implemented several basis generation procedures that started with the identity as the first basis and then included additional bases. Generating random bases from the set of all bases proved inefficient because the corresponding functions $\Psi_D$ often did not improve the solution. Bases were then generated from the set of optimal bases for some $\xi$ by solving (1.2) for varying values of $\xi$ and $\chi$. A patterned choices of values for $\xi$ (using $\xi_i \pm 3\sigma_i$ for each $i$) proved slightly more efficient than random selection of $\xi_i$ and was, therefore, used in the experiments described below. Two different choices for $\chi$ were implemented. On the $k$th solution of (4.16) with new bases added to $D$, we used $\chi = \chi^{k-1}$ or $-\chi^{k-1}$ where $\chi^{k-1}$ was the optimal value of $\chi$ from the $(k-1)$th solution of (4.16). In our experiments, $\chi = -\chi^{k-1}$ proved more effective because it included a broader class of bases by exploring different regions of $\chi$.

6. Numerical Results

Formulation 4.16 was used with the MINOS/AUGMENTED (MINOS Version 4.0) computer program for nonlinearly constrained problems. This implementation on The University of Michigan's Amadahl 5860 Computer found optimal solutions (for (4.16)) for all of the test problems tried. Other approaches encountered difficulties as discussed in the Appendix.

The appropriate use of tolerances on constraint satisfaction was especially important in our implementation. A value of $10^{-5}$ was used for ROW
TOLERANCE in the MINOS SPECS file. This allowed some flexibility in satisfying the constraints without creating large infeasibilities. To avoid infinite values of \( \psi_D^* \), the constraints
\[
-q_i^{D^+} \leq v_{D,i} \leq q_i^{D^-}
\]  
were added to (4.16). A penalty term was included in the subgradient definition from (4.13) when (5.1) was satisfied as an equality (within the tolerance). This made the subgradient definition consistent at the boundary of the effective domain of \( \psi_D^* \).

Problem testing initially involved simple recourse problems which were solved exactly by the formulation in (4.16). After conducting this check, the method was applied to general recourse problems. The results reported here apply to the small energy decision problem in Louveaux [1986]. In this problem, \( m_1 = 2 \), \( n_1 = 4 \), \( m_2 = 7 \), and \( n_2 = 12 \). Of the seven recourse problem constraints, four are balancing constraints that are fixed at zero and three are demand constraints that are stochastic. The demands were assumed independent and normally distributed. Different problems were generated by varying the means and standard deviations of these random variables (see Table 1).

To check the accuracy of the sublinear approximation method, the test problems were also solved using the L-shaped code, NDSP (Birge [1985]), with upper bounds from the Edmundson-Madansky inequality (Madansky [1959]) and lower bounds from Jensen's inequality. After the kth solution of the problem with these bounds, the bounds were refined for each \( \xi_1 \) by partitioning the interval of the approximation that contained \( \chi_1^{k-1} \), the ith component of the optimal \( \chi \) found at the (k-1)th iteration. The NDSP program allows up to 125 realizations of the random vector \( \xi \) or four iterations of this bounding procedure. The results for the upper bounds and CPU second times for each iteration appear in Table 2 under the NDSP columns. The upper and lower
bounds on the final iteration value appear in Table 3. Note that in Problem 8 after 33.80 CPU seconds, NDSP terminated because a limit of 100 L-shaped algorithm iterations (corresponding to 100 cutting planes) had been performed. Execution was not continued because of large row residuals caused by instability in the bases including these cuts (see Birge [1986] for a discussion of this phenomenon).

The sublinear approximation results appear in Tables 2 and 3 with columns corresponding to the choices of \( x = x^{k-1} \) for finding new bases as noted above. The algorithm is terminated when the addition of new bases does not change the solution. This is indicated by "B" in Table 2. The upper bound result of solving (4.16) and the CPU seconds to obtain this result appear in Table 2. The computation times include some iteration logging that is comparable but not identical to iteration log times included in the CPU times for NDSP.

Table 3 provides bounds on the objective values

\[
Z(NDSP) = c x_{ND} + \Psi(x_{ND})
\]

and

\[
Z(SLA) = c x_{SL} + \Psi(x_{SL}),
\]

where \((x_{ND}, x_{ND})\) is the last solution found by NDSP and \((x_{SL}, x_{SL})\) is the last solution found by the Sublinear Approximation (SLA) method. Upper bounds are found using the Edmundson-Madansky inequality, and lower bounds are found using the Jensen inequality. The relative differences between the means of the upper and lower bounds given by NDSP and the sublinear approximation method are also given in Table 3.

From Table 2, it appears that the sublinear approximation method provides upper bounds that are close to the NDSP values within times that are comparable with the NDSP times. Table 3 indicates how well the \( x \) and \( \chi \) solutions from the sublinear approximation method perform in (1.1). The upper and lower bounds result from upper and lower approximations of \( \Psi \). In Table 3, note that the mean values of the upper and lower bounds on \( Z(SLA) \) are at most 5.8% worse than the NDSP values and are at best 12.5% better.
A key property to note in examining Table 2 is that the sublinear approximation times are comparable for all problems. The NDSP approximations, however, become much worse as additional randomness is included into the problem. The true advantage of the sublinear approximation method is evident in Problems 7-9. Here, the sublinear approximation method obtains relatively good bounds (within 25% of the best lower bound) in fractions of the times for NDSP.
Table 1. Problem Parameters

<p>| Problem | Random Variables Parameters |  |  |  |  |  |
|---------|-----------------------------|---|---|---|---|
|         | $\mu_1$ | $\sigma_1$ | $\mu_2$ | $\sigma_2$ | $\mu_3$ | $\sigma_3$ |
| 1       | 5.0     | 1.0     | 4.0     | 0.0     | 3.0     | 0.0     |
| 2       | 5.0     | 0.0     | 4.0     | 1.0     | 3.0     | 0.0     |
| 3       | 5.0     | 0.0     | 4.0     | 0.0     | 3.0     | 1.0     |
| 4       | 5.0     | 1.0     | 4.0     | 1.0     | 3.0     | 0.0     |
| 5       | 5.0     | 1.0     | 4.0     | 0.0     | 3.0     | 1.0     |
| 6       | 5.0     | 0.0     | 4.0     | 1.0     | 3.0     | 1.0     |
| 7       | 5.0     | 1.0     | 4.0     | 1.0     | 3.0     | 1.0     |
| 8       | 2.0     | 1.0     | 4.0     | 1.0     | 3.0     | 1.5     |
| 9       | 4.0     | 1.0     | 5.0     | 1.0     | 5.0     | 1.0     |</p>
<table>
<thead>
<tr>
<th>Problem</th>
<th>Iteration</th>
<th>NDSP Upper Bound</th>
<th>CPUs(^a)</th>
<th>Sublinear Approximation (x = x^{k-\eta}) Upper Bound</th>
<th>CPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>395.</td>
<td>0.41</td>
<td>452.</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>367.</td>
<td>0.75</td>
<td>385.</td>
<td>1.79</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>366.</td>
<td>1.12</td>
<td>383.</td>
<td>2.38</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>366.</td>
<td>1.27</td>
<td>b</td>
<td>2.75</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>345.</td>
<td>0.38</td>
<td>398.</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>335.</td>
<td>0.71</td>
<td>345.</td>
<td>1.48</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>335.</td>
<td>0.79</td>
<td>b</td>
<td>1.71</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>335.</td>
<td>0.88</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>291.</td>
<td>0.42</td>
<td>304.</td>
<td>1.44</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>289.</td>
<td>0.47</td>
<td>289.</td>
<td>2.03</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>b</td>
<td>--</td>
<td>b</td>
<td>2.42</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1180.</td>
<td>0.42</td>
<td>572.</td>
<td>3.37</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>454.</td>
<td>1.01</td>
<td>440.</td>
<td>5.78</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>425.</td>
<td>1.85</td>
<td>b</td>
<td>6.22</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>424.</td>
<td>2.88</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1140.</td>
<td>0.50</td>
<td>478.</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>505.</td>
<td>1.19</td>
<td>407.</td>
<td>2.14</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>388.</td>
<td>1.97</td>
<td>402.</td>
<td>3.82</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>377.</td>
<td>4.30</td>
<td>b</td>
<td>4.37</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1100.</td>
<td>0.37</td>
<td>425.</td>
<td>2.17</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>473.</td>
<td>0.73</td>
<td>350.</td>
<td>7.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>357.</td>
<td>1.05</td>
<td>b</td>
<td>7.53</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>347.</td>
<td>1.54</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2060.</td>
<td>0.51</td>
<td>617.</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1100.</td>
<td>2.09</td>
<td>470.</td>
<td>3.56</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>488.</td>
<td>9.59</td>
<td>b</td>
<td>4.09</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>466.</td>
<td>26.44</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2360.</td>
<td>0.61</td>
<td>511.</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1350.</td>
<td>2.17</td>
<td>359.</td>
<td>2.82</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>491.</td>
<td>11.63</td>
<td>b</td>
<td>3.52</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>c</td>
<td>33.80</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2420.</td>
<td>0.48</td>
<td>626.</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>878.</td>
<td>1.49</td>
<td>476.</td>
<td>5.07</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>475.</td>
<td>6.94</td>
<td>b</td>
<td>5.62</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>471.</td>
<td>13.52</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

\(a\) - CPU seconds

\(b\) - No improvement

\(c\) - Iteration limit exceeded
Table 3. Upper and Lower Bounds on Last Solution Value.

<table>
<thead>
<tr>
<th>Problem</th>
<th>NDSP Upper Bound</th>
<th>NDSP Lower Bound</th>
<th>Sublinear Upper Bound</th>
<th>Sublinear Lower Bound</th>
<th>(\chi = \chi^{k-1}) Mean Difference*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>366.</td>
<td>363.</td>
<td>371.</td>
<td>370.</td>
<td>0.016</td>
</tr>
<tr>
<td>2</td>
<td>335.</td>
<td>334.</td>
<td>343.</td>
<td>343.</td>
<td>0.026</td>
</tr>
<tr>
<td>3</td>
<td>289.</td>
<td>289.</td>
<td>289.</td>
<td>289.</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>424.</td>
<td>419.</td>
<td>438.</td>
<td>436.</td>
<td>0.037</td>
</tr>
<tr>
<td>5</td>
<td>377.</td>
<td>374.</td>
<td>390.</td>
<td>389.</td>
<td>0.037</td>
</tr>
<tr>
<td>6</td>
<td>347.</td>
<td>345.</td>
<td>350.</td>
<td>349.</td>
<td>0.010</td>
</tr>
<tr>
<td>7</td>
<td>466.</td>
<td>438.</td>
<td>494.</td>
<td>462.</td>
<td>0.058</td>
</tr>
<tr>
<td>8</td>
<td>491.</td>
<td>316.</td>
<td>369.</td>
<td>337.</td>
<td>-0.125</td>
</tr>
<tr>
<td>9</td>
<td>471.</td>
<td>433.</td>
<td>486.</td>
<td>465.</td>
<td>0.052</td>
</tr>
</tbody>
</table>

\*\(-\{(UB(SLA) + LB(SLA))/2\} - \{(UB(NDSP) + LB(NDSP))/2\}\)/\{(UB(NDSP) + LB(NDSP))/2\}
7. Discussion and Extensions

The results in Section 5 indicate that the sublinear approximation method may be a good approximation procedure to use in complex stochastic programs. It provides good bounds on the optimal objective value quickly. The solutions obtained from it are also generally close in value to the solutions obtained by other solution procedures. In addition, the sublinear approximation can be applied to problems with many random variables where other methods cannot be applied. This ability for general stochastic programs sets the sublinear approximation apart from all other methods.

The sublinear approximation method only provides upper bounds on the optimal objective value. A lower bound would be required to determine how good the solution is and whether additional optimization is required. The Jensen inequality lower bound was used in Section 5 to achieve this. Other possibilities and uses of this inequality are described in Birge and Wets [1986a]. One of these alternatives that would be natural in this context is outer linearization to bound the function from below, an approach also used in Marti [1975].

The upper bounding procedure described above can also be made more applicable by including more distributions and improving the solution procedure. Alternatives are discussed in the Appendix.

The solution of (4.2) is still at best an approximation (and upper bound) of the solution of the original problem (1.1). There are several alternatives for improving the approximation of (1.1) after the solution of (4.2). One alternative is to find a lower bound on the value of $z$ corresponding to the (dual) solution of (4.2). A cutting plane can be formed from this and used to initiate an L-shaped algorithm as in NDSP. The L-shaped algorithm can then be followed in a sequential approximation procedure (Birge [1983]) using the objective function in (4.2) as an upper bound to check
optimality. When this bound ceases to improve, other upper bounding methods (such as in Birge and Wets [1986c]) could be used.

Another possibility for improving the sublinear approximation is to evaluate the recourse function exactly over groups of random variables for which integration with respect to these variables is tractable. For example, suppose $m_2=3$,

$$\Psi_2(x, \xi) = \min \{ qy \mid Wy = (\xi_1 - x_1, \xi_2 - x_2, 0)^T, y \geq 0 \}$$

$$= \pi_1 \xi_1 + \pi_2 \xi_2$$

where $(\xi_1, \xi_2) = (\xi_1 - x_1, \xi_2 - x_2) \in \mathbb{R}^i$, $i=1, \ldots, r$, and we can evaluate

$$\Psi_2(x_1, x_2) = \sum_{i=1}^{r} \int_{\mathbb{R}^i} [\pi_1 \xi_1 + \pi_2 \xi_2] P(d\xi_1, d\xi_2).$$

(6.2)

We then combine the approximation in (6.2) with a simple recourse function approximation, $\Psi_3(x_3)$, evaluated with respect to $z_i = \xi_i - x_i$. The result is another upper bound on $\Psi$, where

$$\Psi(x) \leq \Psi_2(x_1, x_2) + \Psi_3(x_3).$$

(6.3)

Again combinations of approximations as in (6.3) can be used to approximate $\Psi$ from above. The practicality of this approach of course relies upon the possibility to evaluate integrals as in (6.2) efficiently. For large groups of random variables, this effort may negate any advantages from this approximation scheme.

Another possibility for improving the sublinear approximation is to use the functions, $\Psi_{dj}(x, \xi)$, as in the approximation of $\psi(x, \xi)$ in (3.8). Instead of integrating each $\Psi_{dj}$ separately, we can approximate the integral of

$$\int_{\mathbb{R}^L} \psi_{dj}(x, \xi), \quad j=1, \ldots, L.$$ 

This can be achieved by substituting a discrete approximation for $P$ or by random selection. The use of this approximation as described in Birge and Wets [1986b] avoids the optimization problem in (1.2). Its use in stochastic programming is described in Wets [1985].
A possible extension that maintains the inner linearization procedure is to incorporate the sublinear approximation method into a generalized programming algorithm as in Nazareth and Wets [1986] and the sequential algorithm in Birge [1983]. In this case, after (4.2) is solved to obtain \((\sigma^0, \pi^0)\) with dual solutions \((x^0, x^0)\) and value, \(w^0\), we use a refined upper approximation, \(\Psi^0\), of \(\Psi\) and solve
\[
\min \Psi^0(x) + \pi^0 x \\
\text{x}
\] (6.4)
to obtain \(x^1\). The first master program for the generalized programming approach is then

\[
\text{find } x \in \mathbb{R}^n, \lambda_0, \lambda_1 \in \mathbb{R} \text{ such that}
\]

\[
Ax = b, \\
Tx - \lambda_0 x^0 - \lambda_1 x^1 = 0,
\]
\[
\lambda_0 + \lambda_1 = 1,
\]

and \(z = cx + \lambda_0 w^0 + \lambda_1 \Psi^0(x^1)\) is minimized. Program 6.5 then initiates the generalized programming procedure.

More possibilities for the sublinear approximation procedure appear in the context of parallel computation as described in Wets [1985]. Because the work in evaluating each \(\Psi_{D,i}^0\), for example, as in (4.8) can be conducted separately, the algorithm can immediately benefit from parallel computation. This area and the extensions mentioned above all need to be explored to exploit the sublinear approximation properties fully. The results to date, however, still indicate that this new approximation method provides an efficient way to compute approximate solutions for previously intractable problems.
Appendix. Alternative Approaches

The method given above uses formulation (4.16) with a normal approximation of the random variables $\xi_i$ and a patterned choice of optimal bases for varying values of the random variables. Alternatives are possible for each of these areas. A different formulation can be used, other approximations of the random variables are possible, and different selection rules for choosing bases are admissible.

Several different solution procedures were attempted for Program 4.2. In general, the nondifferentiable techniques encountered line search problems in some examples. The minimum-norm subgradient method appeared to be quite sensitive to numerical errors in the calculation of the values of $\varphi_D^*$. Since these values resulted from numerical integration, error was present, making it difficult to obtain the set $D'$ in (4.15) exactly. This caused the subdifferential to be incomplete, which allowed the procedure to miss finding a descent direction.

A bundling method (Lemaréchal [1978]) was also attempted, using the unconstrained optimization code, M1FC1, of Lemaréchal and Bencora Imbert [1985]. This method has an advantage over the minimum-norm subgradient method in its greater insensitivity to small errors in the function and subgradient values. In our implementation, the linear constraints were added to the objective using a penalty parameter. The method was then not always able to identify an optimum. The two nondifferentiable methods could be improved by increased integration accuracy and careful tolerancing for the minimum-norm subgradient method and by the active set method for linear constraints in the bundle method (Lemaréchal, et al. [1981]).

Other approximation techniques for the distribution of $\xi_i$ assume that $\xi_i$ has a certain distribution and fit the parameters of the
distributions using \((\mathbf{D}^{-1})_i (\mathbf{e}_i - \mathbf{c})\). These parameters may be estimated from the moments of \(\mathbf{z}_i\) that can be calculated. Another type of parameter estimation can be used with an assumption of a piecewise linear distribution for \(\mathbf{z}_i\). The goal is to obtain the best such approximation. An advantage of this approach in the context of (4.2) is that the resulting approximate functions \(\mathbf{y}^*_D\) are piecewise quadratic. The methods in Wets [1975] for simple recourse problems can then be used in solving (4.2). Procedures for finding these approximations are discussed in Birge and Wets [1986a].

The choice of bases presents an even greater number of possible alternatives. One possibility is use all bases which are optimal for (1.2) for some \(\mathbf{e} \in \mathbf{Z}\). Let this set be

\[
D^1 = \{ D | D = [\mathbf{W}_1, ..., \mathbf{W}_m] \\
D \text{ is invertible, } D^{-1}(\mathbf{e}_i - \mathbf{c}) \geq 0 \\
\text{for some } \mathbf{e} \in \mathbf{Z}, \text{ and, for} \\
\pi^D = q(D)D^{-1}, \, \pi^D \mathbf{w} \leq q \}
\]

and let the set of all bases in \(W\) be

\[
D^2 = \{ D | D = [\mathbf{W}_1, ..., \mathbf{W}_m] \text{ and } D \text{ is invertible} \}.
\]

At first, it might appear that \(D^1\) would be a sufficient set of bases to use in characterizing \(\mathbf{co}(\mathbf{y}_D | D \in D^2)\). This would be a sufficient set for equality in (3.8), but it is not sufficient to characterize \(\mathbf{co}(\mathbf{y}^D | D \in D^2)\) as the following example shows.

Consider the following expansion of Example 3.15:

\[
\text{find } x, y_1^+, y_1^-, y_2^+, y_2^-, y_3, y_4, y_5, y_6, \geq 0 \text{ such that } \\
x \leq 1 \\
t_1 x + y_1^+ - y_1^- + y_3 - y_4 - y_5 + y_6 = \xi_1 \\
t_2 x + y_2^+ - y_2^- + y_3 + y_4 - y_5 - y_6 = \xi_2
\]
to minimize \( z = cx + \sum_{i=1}^{5} \xi_i [y_i^+ + y_2^+ + y_3^- + y_4 + y_5 + y_6] \),

where the random variables, \( \xi_i, i=1,2 \), are normally distributed with means 0 and variances, \( \sigma_i^2, i=1,2 \). The optimal bases are illustrated by the regions in Figure 3,

![Figure 3. Optimal basis regions](image)

where

\[
D_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad (A.4)
\]

and the other bases correspond to the boundary lines as illustrated. Note that the basis

\[
\bar{\pi} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

(A.5)

corresponds to \( \tilde{\pi} = [1, 1]\bar{\pi}^{-1} = [1, 1] \) which is not dual feasible, so \( \bar{\pi} \notin D^1 \).
Consider the recourse function

\[ \Psi_{D_1}(x) = \Psi(\hat{x}_1, \hat{x}_2) = \int_{\xi_1 \geq \hat{x}_1} (\xi_1 - \hat{x}_1) P_{D_1}(d\xi_1) + \int_{\xi_1 < \hat{x}_1} (\hat{x}_1 - \xi_1) P_{D_1}(d\xi_1) \]
\[ + \int_{\xi_2 \geq \hat{x}_2} (\xi_2 - \hat{x}_2) P_{D_1}(d\xi_2) + \int_{\xi_2 < \hat{x}_2} (\hat{x}_2 - \xi_2) P_{D_1}(d\xi_2), \]

where \( \xi_1 = \xi_1 - \xi_2, \xi_2 = \xi_2, \hat{x}_1 = x_1 - x_2 = (D_1^{-1}) x, \) and \( \hat{x}_2 = x_2 = (D_1^{-1} x). \) The partial derivatives of \( \Psi \) with respect to \( \hat{x}_1 \) and \( \hat{x}_2 \) are

\[ \frac{\partial \Psi}{\partial \hat{x}_1} = -P_{D_1} \{ \xi_1 \geq \hat{x}_1 \} + (1 - P_{D_1} \{ \xi_1 \geq \hat{x}_1 \}), \]

and

\[ \frac{\partial \Psi}{\partial \hat{x}_2} = -P_{D_1} \{ \xi_2 \geq \hat{x}_2 \} + (1 - P_{D_1} \{ \xi_2 \geq \hat{x}_2 \}). \]

From (A.7),

\[ \min \Psi(\hat{x}_1, \hat{x}_2) = \Psi(0,0), \]

because \( P_{D_1} \{ \xi_1 \geq \hat{x}_1 \} = P_{D_1} \{ \xi_2 \geq \hat{x}_2 \} = 1/2. \) Since \( \xi_1 \) is normally distributed with mean 0 and variance \( 2\sigma^2 \) and \( \xi_2 = \xi_2, \) we have

\[ \Psi_{D_1}(0,0) = \Psi(0,0) = \frac{(2 + \sqrt{2})\sigma}{\sqrt{\pi}}. \]

By symmetry, we also have

\[ \Psi_{D_1}(0,0) = \Psi(0,0). \]

If we use \( \bar{D} \), however, we can substitute \( \xi_1 \) for \( \xi_1 \) in (4.22). At \( (0,0) \), we then obtain

\[ \Psi_{\bar{D}}(0,0) = \frac{(2\sqrt{2})\sigma}{\sqrt{\pi}}. \]

From (A.8-11),

\[ \Psi_{\bar{D}}(0,0) < \min_{x \in D} \Psi_{\bar{D}}(x) \leq \infty \{ \Psi_{\bar{D}}(0,0) \mid D \notin D^1 \}. \]
The inequality in (4.28) shows that $\mathcal{D}^1$ is not a sufficient set to describe $\text{co}\{\psi_D \mid D \in \mathcal{D}^2\}$. The reason for this is illustrated in Figure 4. The area of the level set of $\psi$ covered by $\psi_D$ has higher probability than the area covered by $\psi_{D_1}$ or any other $\psi_D$, $D \in \mathcal{D}^1$.

Using all bases in $\mathcal{D}^2$ would, however, be inefficient for computation. Instead, it appears best to begin with a limited set of bases and then to add bases to improve the approximation. This sequential choice would be driven by the characteristics of the problem. One possibility for this procedure is to begin with a simple approximation involving, for example, the identity as in $\psi_D$ above. New bases could be added to approximate the level curve of $\psi$. A procedure for doing this is described in Wets [1985] and Birge and Wets [1986b]. This procedure, however, only tries to improve the approximation of $\psi(x, \xi)$ as much as possible. This process may miss the best approximation of $\psi$ as the preceding example illustrates.
7. References


Parikh, S.C., [1968], Lecture Notes on Stochastic Programming, University of California, Berkeley.


Wets, R. J-B., [1974a], Stochastic Programming, Lecture Notes, University of Kentucky, Lexington.


