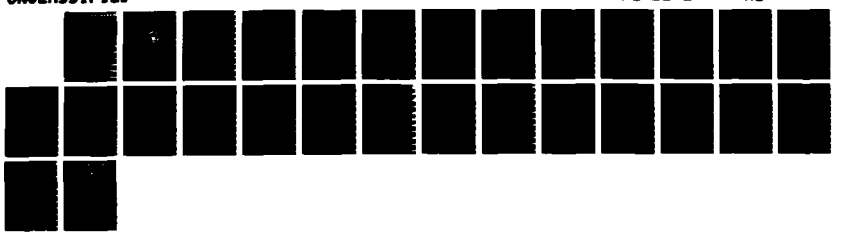
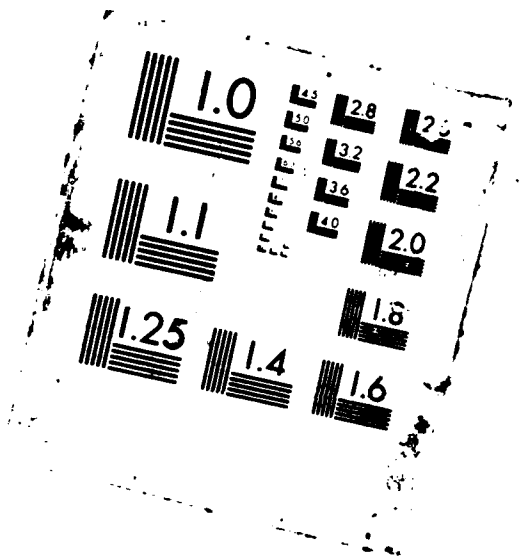
AD-A178 694 A SEPARABLE PIECEWISE LINEAR UPPER BOUND FOR STOCHASTIC 1/1
LINEAR PROGRAMS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY
CA J R BIRGE ET AL. FEB 87 NPS55-67-001

UNCLASSIFIED F/G 12/1 NL

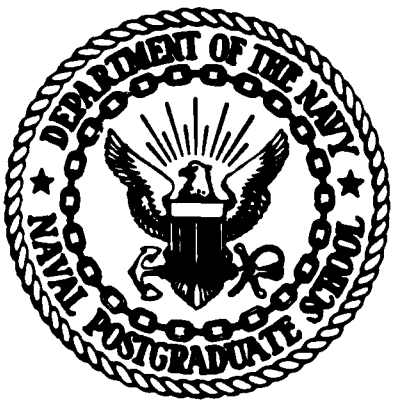



1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0

NPS55-87-001

NAVAL POSTGRADUATE SCHOOL

Monterey, California



A SEPARABLE PIECEWISE LINEAR UPPER BOUND FOR STOCHASTIC LINEAR PROGRAMS

JOHN R. BIRGE
STEIN W. WALLACE

FEBRUARY 1987

Prepared for:
Naval Postgraduate School
Monterey, CA 93943-5000

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NPS55-87-001	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (*If applicable*) Code 55	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b. ADDRESS (*City, State, and ZIP Code*)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (*Include Security Classification*)

A SEPARABLE PIECEWISE LINEAR UPPER BOUND FOR STOCHASTIC LINEAR PROGRAMS

12. PERSONAL AUTHOR(S)

Birge, John R. and Wallace, Stein W. (Chr. Michelsen Institute)

13a. TYPE OF REPORT	13b TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15 PAGE COUNT
Technical	FROM _____ TO _____	1987 February	23

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	stochastic programming, upper bounds, convex functions, integration

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

Stochastic linear programs require the evaluation of an integral in which the integrand is itself the value of a linear program. This integration is often approximated by discrete distributions that bound the integral from above or below. A difficulty with previous upper bounds is that they generally require a number of function evaluations that grows exponentially in the number of variables. We give a new upper bound that requires operations that only grow polynomially in the number of random variables. We show that this bound is sharp if the function is linear and give computational results to illustrate its performance.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL John R. Birge	22b TELEPHONE (*Include Area Code*) (408)646-2594 22c OFFICE SYMBOL Code 55

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office 1986—606-243

A Separable Piecewise Linear Upper Bound for Stochastic Linear Programs

John R. Birge*

Department of Industrial and Operations Engineering

The University of Michigan

Ann Arbor, Michigan 48109

USA

Stein W. Wallace

Department of Science and Technology

Chr. Michelsen Institute

N-5036 Fantoft, Norway

Abstract : Stochastic linear programs require the evaluation of an integral in which the integrand is itself the value of a linear program. This integration is often approximated by discrete distributions that bound the integral from above or below. A difficulty with previous upper bounds is that they generally require a number of function evaluations that grows exponentially in the number of variables. We give a new upper bound that requires operations that only grow polynomially in the number of random variables. We show that this bound is sharp if the function is linear and give computational results to illustrate its performance.

Keywords: stochastic programming, upper bounds, convex functions, integration.

* - This author's work was supported in part by Office of Naval Research Grant N00014-86-K-0628,by Dalhousie University during a visit in the Department of Mathematics, Statistics, and Computing Science, and by the National Research Council under a Research Associateship at the Naval Postgraduate School, Monterey, California.

1. Introduction

Stochastic linear programs can be formulated for a variety of applications. Some examples include airline scheduling (Ferguson and Dantzig [1956]), financial planning (Kusy and Ziemba [1986]), energy modeling (Birge [1987]) and water resource planning (Prékopa and Szantai [1978]). The basic model we consider here is the stochastic linear program with recourse in the following general form:

$$\min_x\{c^T x + Q(x) | Ax = b, x \geq 0\}$$

where

$$Q(x) = \int Q(x, \xi, \phi) P(d\xi, d\phi)$$

and the *recourse function* is defined as

$$Q(x, \xi, \phi) = \min\{q^T y | Wy = \xi - Tx, u + \phi \geq y \geq 0\},$$

where $x \in \Re^{n_1}$, $y \in \Re^{n_2}$, $b \in \Re^{m_1}$, and (ξ, ϕ) is a random vector on the probability space $(\Re^{m_2+n_2}, \mathcal{F}, \mathrm{P})$ with support, $\Xi \times \Phi$. The vectors, c, q, and u, and matrices, A, W, and T are dimensioned correspondingly. The fundamental problem in stochastic programming is to evaluate the integral of Q. In this paper, we describe a method for finding an upper bound on Q that requires a polynomial number of operations in the number of random variables.

Previous results in bounding expressions for Q are described in Birge and Wets [1986a]. The bounds are based on the convexity and positive homogeneity of Q. The first result is due to Jensen [1906]'s inequality which provides a lower bound on Q. The usefulness of this lower bound is that it requires an evaluation of Q at one point (the mean of the random variables) and has been found to be generally sharp in some practical examples (see, e.g., Hausch and Ziemba [1983]). Madansky [1959] provided an upper bound following Edmundson [1956] that is based on the theory of moment spaces and amounts to weighting the extreme points of the support of the random variables. Ben-Tal and Hochman [1972] and Huang, Ziemba and Ben-Tal [1977] refined this bound for independent random variables. Dupačová [1976] formulated a bound of the same general type for dependent random variables that was extended to unbounded ranges and non-polyhedral sets in Gassmann and Ziemba [1986]. Frauendorfer [1986] provided a sharper bound in the bounded range,

dependent variable case, and Birge and Wallace [1986] gave a bound and method for refinement for special cases of dependent random variables.

The upper bounds mentioned above all have the property that they are solutions to moment problems with varying conditions. Dupačová's work on minimax solutions (Žáčková [1966]) led to these conclusions and to the use of the generalised moment problem. Ermoliev, et al. [1987] provided a general programming framework for solving the general problem. It is used in Birge and Wets [1987] for bounds with piecewise linear approximations on moment constraints and in Cipra [1985] with first and second moment constraints.

The problem with each of these bounds is that they require an exponentially increasing number of function evaluations as the number of random variables increases. An alternative for this situation was given by the ray approximation procedure in Birge and Wets [1986a]. This uses the sublinearity property of the recourse function to obtain a separable function that majorises Q. This approach is generalised in Birge and Wets [1986b]. Wallace [1987b], on the other hand, formulated a procedure that applies to problems in which the recourse function involves the solution of a network problem. Our procedure is a combination and generalization of these two basic approaches. The algorithm we give provides a separable piecewise linear function that bounds Q throughout the support of the random variables and can be easily evaluated.

Section 2 presents our basic algorithm and the separable piecewise linear upper bound (*SPLU*). Its properties are described in Section 3. Section 4 gives an illustrative small example and provides comparison with the upper bound of Edmundson and Madansky. Extensions of the basic algorithm and conclusions are given in Section 5.

2. The Basic Algorithm

We give a general method for finding an upper bound on the expected value of the value of a linear program with random right-hand sides and random upper bounds on the variables. To simplify notation and to establish general results, we consider the following system :

$$\begin{aligned} A_1 x &= b_1 + \xi \\ A_2 x &= b_2 \qquad (1) \\ 0 \leq x &\leq c + \phi \end{aligned}$$

where $A_1 \in \Re^{m_1 \times n}$, $A_2 \in \Re^{(m-m_1)\times n}$, $(A_1|A_2)^T = A$ is the coefficient matrix, $(b_1|b_2)^T = b$ is the fixed part of the right-hand side, c is the fixed part of the bounds on the variables, ξ is the random availability of resources and ϕ is the random part of the variable capacities, where $\phi \geq 0$. We assume that there is a positive probability that $\phi = 0$. Next define $Q(\xi, \phi)$ by

$$Q(\xi, \phi) = \min\{q^T x | (1)\}. \tag{2}$$

Finally define $\chi(\xi, \phi, d^-, d^+)$ as the set of x-vectors satisfying

$$\begin{aligned} A_1 x &= \xi \\ A_2 x &= 0 \\ d^- \leq x &\leq \phi + d^+. \end{aligned} \tag{3}$$

Our goal is to find an upper bound on $Q(\xi, \phi)$, or, more precisely, on $EQ(\xi, \phi)$. We do this by finding a separable piecewise linear function $U(\xi, \phi)$ defined by

$$U(\xi, \phi) = Q(\bar{\xi}, 0) + H(\phi) + \sum_{i=1}^{m_1} \begin{cases} q^T x^{i+}(\xi_i - \bar{\xi}_i) \text{if} \xi_i \geq \bar{\xi}_i \\ q^T x^{i-}(\bar{\xi}_i - \xi_i) \text{if} \xi < \bar{\xi}_i \end{cases},$$

where $\bar{\xi}_i = E\xi_i$, and $H(\phi)$ is a piecewise linear function in ϕ.

Algorithm 1

Step 0: Find $Q(\bar{\xi}, 0)$ with optimal solution x^0, where

$$x^0(i) = \begin{cases} e_i^T B_0^{-1}(b + \bar{\xi}) & \text{if } i \text{ is basic,} \\ 0 & \text{if } i \text{ is nonbasic at lower bound,} \\ c(i) & \text{if } i \text{ is nonbasic at upper bound.} \end{cases}$$

Assume for simplicity that the first m variables are basic. Let $x^{i+} = (B_0^{-1}e_i, 0, 0, \ldots, 0)$ and $x^{i-} = (-B_0^{-1}e_i, 0, 0, \ldots, 0)$ where $i = 1, 2, \ldots, m_1$. Let

$$\alpha^1(i) = \max - x^0(i) - \sum_{j=2}^{m_1} x^{j+}(i) y^{j+} - \sum_{j=2}^{m_1} x^{j-}(i) y^{j-}$$

subject to

$$\begin{aligned} y^{j+} - y^{j-} &= \xi_j - \bar{\xi}_j \\ \xi_j^{\min} &\leq \xi_j \leq \xi_j^{\max}, j = 2, \ldots, m_1. \end{aligned}$$

Let

$$\beta^1(i) = \min - x^0(i) - \sum_{j=2}^{m_1} x^{j+}(i)y^{j+} - \sum_{j=2}^{m_1} x^{j-}(i)y^{j-} + c(i)$$

subject to

$$\begin{aligned} y^{j+} - y^{j-} &= \xi_j - \bar{\xi}_j \\ \xi_j^{\min} &\leq \xi_j \leq \xi_j^{\max}, j = 2, \ldots, m_1, \end{aligned}$$

for all $i = 1, \ldots, n$.

If $\alpha^1(i) > 0$ for some i or $\beta^1(i) < 0$ for some i, let $x^{i+} = \alpha^1(i) = (0, \ldots, 0)$, and $\beta^1(i) = x^0(i) + c(i)$ for all $i = 1, \ldots, m_1$ and go to Step 1 with $r = 1$.

Otherwise, check

$$e(i) = \max - x^{1+}(i)y^+ - x^{1-}(i)y^-$$

subject to

$$y^+ - y^- = \xi_1 - \bar{\xi}_1$$

$$\xi_1^{\min} \leq \xi_1 \leq \xi_1^{\max}$$

and

$$f(i) = \min - x^{1+}(i)y^+ - x^{1-}(i)y^-$$

subject to

$$y^+ - y^- = \xi_1 - \bar{\xi}_1$$

$$\xi_1^{\min} \leq \xi_1 \leq \xi_1^{\max}.$$

If $\alpha^1 + e \leq 0$ and $\beta^1 + f \geq 0$, then $Q(\xi, \phi)$ is linear in ξ, go to Step 4.

Otherwise, let $r = 1$ and go to Step 1.

Step 1: If $\xi_r^{\max} < +\infty$, solve

$$\min\{q^T x \mid \chi[(\xi_r^{\max} - \bar{\xi}_r)e_r, 0, \alpha_r, \beta_r]\} = q^T x^{r+}(\xi_r^{\max} - \bar{\xi}_r).$$

Else (let $\beta_*^r(i) = \infty$ if $\beta^r(i) = +\infty$, $\beta_*^r(i) = 0$ otherwise) and solve

$$\min\{q^T x \mid \chi(e_r, 0, 0, \beta_*^r)\} = q^T x^{r+}.$$

If $\xi_r^{\min} > -\infty$, solve

$$\min\{q^T x \mid \chi[(\xi_r^{\min} - \bar{\xi}_r)e_r, 0, \alpha_r, \beta_r]\} = q^T x^{r+}(-\xi_r^{\min} + \bar{\xi}_r).$$

Else (let $\beta_*^r(i) = \infty$ if $\beta^r(i) = +\infty$, $\beta_*^r(i) = 0$ otherwise) and solve

$$\min\{q^T x \mid \chi(-e_r, 0, 0, \beta_*^r)\} = -q^T x^{r-}.$$

If Step 1 was entered with $x^{i\pm} = (0, \ldots, 0)$ for all i, go to Step 2; otherwise, go to Step 4.

Step 2: For $i = 1, \ldots, n$, solve

$$\alpha^{r+1}(i) = \max - x^0(i) - \sum_{j \neq r+1} x^{j+}(i) y^{j+} - \sum_{j \neq r+1} x^{j-}(i) y^{j-}$$

subject to

$$\begin{aligned} y^{j+} - y^{j-} &= \xi_j - \bar{\xi}_j \\ \xi_j^{\min} &\leq \xi_j \leq \xi_j^{\max}, j \neq r+1, \end{aligned}$$

$$\beta^{r+1}(i) = \min - x^0(i) - \sum_{j \neq r+1} x^{j+}(i) y^{j+} - \sum_{j \neq r+1} x^{j-}(i) y^{j-} + c(i)$$

subject to

$$\begin{aligned} y^{j+} - y^{j-} &= \xi_j - \bar{\xi}_j \\ \xi_j^{\min} &\leq \xi_j \leq \xi_j^{\max}, j \neq r+1. \end{aligned}$$

Step 3: If $r < m$, let $r = r + 1$ and go to Step 1.

Otherwise, go to Step 4.

Step 4: Find

$$\alpha^*(i) = \max - x^0(i) - \sum_{j=1}^{m_1} x^{j+}(i) y^{j+} - \sum_{j=1}^{m_1} x^{j-}(i) y^{j-}$$

subject to

$$\begin{aligned} y^{j+} - y^{j-} &= \xi_j - \bar{\xi}_j \\ \xi_j^{\min} &\leq \xi_j \leq \xi_j^{\max}, j = 1, \ldots, m_1, \end{aligned}$$

$$\beta^*(i) = \min - x^0(i) - \sum_{j=1}^{m_1} x^{j+}(i) y^{j+} - \sum_{j=1}^{m_1} x^{j-}(i) y^{j-} + c(i)$$

subject to

$$\begin{aligned} y^{j+} - y^{j-} &= \xi_j - \bar{\xi}_j \\ \xi_j^{\min} &\leq \xi_j \leq \xi_j^{\max}, j = 1, \ldots, m_1, \end{aligned}$$

for $i = 1, \ldots, n$.

Let $x^* = \text{argmin}\ \{q^T x \mid \chi(0, \phi^{\max}, \alpha^*, \beta^*)\}$. Find a conformal realisation of x^* (Rockafellar [1984,p. 455]),so that

$$x^* = \sum \alpha_k x_k^* \text{ with } \alpha_k > 0,$$

such that $x^*(i) > 0 \Rightarrow x_k^*(i) \geq 0$ and $x^*(i) < 0 \Rightarrow x_k^*(i) \leq 0$, and $x^*(i) = 0 \Rightarrow x_k^*(i) = 0$. An algorithm for finding such a realisation is the "painted index algorithm" in Rockafellar [1984,p.476]. Paint all columns A_j of A such that

$$A_j \text{ is } \begin{cases} \text{white} & \text{if } x^*(j) > 0, \\ \text{black} & \text{if } x^*(j) < 0, \\ \text{red} & \text{if } x^*(j) = 0. \end{cases}$$

Let $k = 1$. Pivot until a Tucker-tableau is reached in which there is a compatible column. This will always be possible in our case. Let the compatible column be A'_j, and let F be the set of indices for the basic columns in the final Tucker tableau. We now have that

$$\sum_{i \in F} A_i A'_j(i) + A_j = 0.$$

If A_j is white, let

$$x_k^*(i) = \begin{cases} A'_j(i) & \text{if } i \in F, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

If A_j is black, reverse all signs in x_k^*. (Note that the sign convention in a Tucker tableau is opposite of the convention in the standard simplex tableau.)

Let $\alpha_k = \min\{x^*(i)/x_k^*(i), x_k^*(i) \neq 0\}$, $x^*(i) = x^*(i) - \alpha_k x_k^*(i)$ and re-paint every column for which $x^*(i) = 0$ red.

If $x^* \neq 0$, let $k = k + 1$ and repeat. Otherwise, go to Step 5 with the conformal realisation $\sum_{k=1}^{K} \alpha_k x_k^*$.

Step 5: Using the cost coefficients $q^T x^{i\pm}$, find $\mathrm{E}_\xi U(\xi, \phi)$. This amounts to performing m_1 simple line integrals.

Step 6: If $x^*(i) > 0$ (so that $x_k^*(i) \geq 0, \forall k$), we are using a variable $x(i)$ with random capacity $\beta^*(i) + \phi_i (\geq \phi_i)$. If $x^*(i) < 0$, we are using a variable $x(i)$ with deterministic capacity $\alpha^*(i) (< 0)$. We shall in the following assume that each variable $x(i)$,such that $x^*(i) \neq 0$, has associated with it a random arc capacity ϕ_i^*.

If $x^*(i) < 0$, we have $\Pr\{\phi_i^* = \alpha^*(i)\} = 1$, if $x^*(i) > 0, \phi^* = \phi + \beta^*(i)$. For each $k = 1, \ldots, K$, let $q_k = \sum q(i)x_k^*(i) (\leq 0)$. Sort the primal supports x_k^* such that $q_1 \leq q_2 \leq \cdots \leq q_K$. Let $k = 1$, $\rho = 0$ (where ρ will become $\mathrm{E}H(\phi)$).

Step 7: Let $P = \{i \mid x_k^*(i) \neq 0\}$. Consider the random variable

$$\beta_k = \max\{0, \min_P\{\phi_i^*/x_k^*(i)\}\}, \beta_k \in [0, \alpha_k].$$

Find $\mathrm{E}q_k\beta_k$. (This work amounts to increasing the capacity of each conformal flow until the first variable capacity is met. This continues on each conformal flow. Details are given for the network case in Wallace [1987b].) Let $\rho = \rho + \mathrm{E}q_k\beta_k$, and $\phi^* = \phi^* - \alpha_k x_k^*$. If $k = K$ or if $q_{k+1} = 0$, stop with $\rho = \mathrm{E}H(\phi)$, otherwise let $k = k + 1$ and repeat Step 7.

End.

The value obtained in Algorithm 1 is indeed an upper bound on the expected linear program value.

Theorem. The value **SPLU**$= \mathrm{E}_{\xi,\phi}[U(\xi, \phi)]$ obtained in Algorithm 1 is an upper bound on $\mathcal{Q} = \mathrm{E}_{\xi,\phi}[Q(\xi, \phi)]$.

Proof: The proof requires only showing that $x = x^0 + \sum(x^{j+}(\xi_j - \bar{\xi})^+ + x^{j-}(\bar{\xi} - \xi_j)^+) + \sum(\beta_k q_k x_k^*)$ is feasible in $\chi(\xi, \phi^*, 0, c)$. This is obtained by noting that the definitions of $x^{r\pm}$, α^r, and β^r in Steps 0 to 2 maintain feasibility for ϕ^*.■

The algorithm as described above is our basic version. We prove certain properties of it in the next section. In Section 5, we present alternative versions of some of the steps in Algorithm 1.

3. Properties of the Upper Bound

The purpose of this section is to show that the upper bound presented in this paper has some desirable properties and to relate the procedure to other bounding methods.

3.1 Exact Bounds for Linear Problems

All other bounds used in stochastic programming are exact whenever $Q(\xi, \phi)$ is linear in ξ and ϕ over the support of the random variables. This is true of the Madansky upper bound, the piecewise linear upper

bound in the pure network case (Wallace [1987b]), the linear upper bound on the expected max flow in a network (Wallace [1987a]), the Jensen lower bound, and the sublinear approximation in Birge and Wets (if the random variables have unbounded support). All acceptable bounds should have this property.

Property 1: The bound SPLU given by Algorithm 1 is exact if $Q(\xi, \phi)$ is a linear function.

Proof: Assume $Q(\xi, \phi)$ is linear in ξ and ϕ and that the reduced cost of a non-basic variable is always different from zero (a dual non- degeneracy assumption). Then $EU(\xi, \phi) = EQ(\xi, \phi)$. Of course, if Q is linear, it can be written as

$$Q(\xi, \phi) = Q(\bar{\xi}, 0) + \sum_{k=1}^{m_1} f_k(\xi_k - \bar{\xi}_k) + \sum_j h_j \phi_j.$$

Clearly, Step 0 provides us with $Q(\bar{\xi}, 0)$. Also, if Q is linear, $\alpha^1 + e \leq 0$, $\beta^1 + f \geq 0$ in Step 0,since the basis corresponding to $Q(\bar{\xi}, 0)$ is feasible for all $\xi \in \Xi$. Hence, $f_k = qx^{k+} = -qx^{k-}$. Therefore, if Q is linear, the algorithm will discover the coefficients of ξ in Step 0 and then go to Step 4.

Let us define a variable i to be stochastic if $\phi_i^{\max} > 0$, otherwise, it is deterministic. Consider the conformal realization of $x^* = \sum \alpha_k x_k^*$. First note that x_k^* is an elementary vector (Rockafellar [1984, p.453]). This means that there is no way to split x_k^* into two or more other vectors where at least one has fewer non-zeroes than x_k^*.

Assume there exists an elementary vector y such that $y(i) \neq 0$ for more than one stochastic random variable. Then fix the value of ϕ_i at 0 for all variables except for those with $y(i) \neq 0$. Then, Q would not be linear. (Compare with the random variable β in Step 7.) Hence, if Q is linear, there is no elementary vector with more than one stochastic variable.

Now, assume that we have found two elementary vectors y_1 and y_2 , such that they share the stochastic variable i (i.e., $y_1(i) \neq 0, y_2(i) \neq 0$). Also assume that $q_1/y_1(i) \neq q_2/y_2(i)$. (The variable q_i defined as in Step 6.) Let all $\phi_j = 0$ for $i \neq j$. Then Q is not linear in variable i, because the marginal gain of increasing ϕ_i is not the same in both elementary vectors. Hence, two elementary vectors can only share a stochastic variable if $q_1/y_1(i) = q_2/y_2(i)$. (This corresponds to two circuits in a pure network that have the same cost and share an arc with a random capacity.) Of course, $h_i = q_1/y_1(i)$.

Hence, if Q is linear, no elementary vector x_k^* has more than one stochastic variable and two elementary

vectors can only share a stochastic variable if they have the same cost (in the sense described above). Since Step 6 only creates elementary vectors x_k^*, the random variable β_k in Step 7 is linear in its single random variable. Hence, our method produces the exact solution.■

3.2 The Bound is Polynomial

The Edmundson-Madansky bound requires that $Q(\xi, \phi)$ be solved in all extreme cases of ξ and ϕ. There are $2^{m_1+n_1}$ such points; hence, the method is exponential in the number of stochastic variables. Only for very moderate values of n_1 and m_1 is it possible to apply this bound.

The major goal of this paper is therefore to find a good upper bound that can be computed in a number of operations that is polynomial rather than exponential *in the number of random variables.*

Property 2: Algorithm 1 calculates SPLU in a number of operations that is polynomial in the number of random variables.

Proof: The amount of work is in the worst case:

Step 0: 1 LP (α^1, β^1 can be found by inspection).

Step 1: $2m_1$ LPs.

Step 4: 1 LP to find x^*. The conformal realisation is independent of n_1 and m_1. (The worst case is n LP's, $n \leq n_1$.)

Step 5: The integration is a constant amount of work for each random variable.

Step 7: Finding $Eq_k\beta_k$ amounts to checking the $m_1 * \max\{m_i\}$ (in the worst case) possible values of β_k. The value m_i is the total number of possible values for ϕ_i. This has to be done not more than n times (since the number of zeroes increases by one for each k).

Hence, the algorithm is linear in n_1 and m_1.■

3.3 Relation to Networks

The method presented in this paper is closely related to the network method in Wallace [1987b]. The

major difference is in Step 1, where we only solve two networks in the network case and not $2m_1$ as here.

Below is a short network interpretation of some of the vectors and scalars used in the algorithm to help in its understanding.

Step 0: The variable x^{i+} shows how the flow changes on the basic arcs as the supply at node i is increased by one unit or (the demand is decreased). Hence,

$$x^{i+}(j) = \begin{cases} +1 & \text{if arc } j \text{ is a forward arc on the path from node } i \text{ to the slack node,} \\ -1 & \text{if arc } j \text{ is a reverse arc,} \\ 0 & \text{if arc } j \text{ is not on the path.} \end{cases}$$

x^{i-} is similarly defined for increased demand (or decreased supply).

$\alpha^1(i) > 0$ implies that with the chosen set of paths $(x_i^{\pm})$ there are supply/demand combinations that give a negative flow on arc i , even when we disregard node 1.

$\beta^1(i) > 0$ implies that with the chosen paths, there are supply/demand combinations that overuse arc i even when we do not consider node 1.

Step 1: $x_i^{\pm}$ are still paths, but not along a basis. Both basic and non–basic arcs are used. If Step 1 finishes successfully, we have actually replaced the original network by a star-shaped network (where the slack node is in the center of the star). The arc going from the center node to node i has unit cost qx_i^-, the arc in the other direction has unit cost qx_i^+. The way we have used α^r and β^r has guaranteed that whatever combination we get of supply and demand, sending that flow along the paths $x_i^{\pm}$ would be feasible and cost the same as in the star–shaped network.

Hence, we have found an upper bounding simple recourse problem (Wets [1983]) . In stochastic programming this approximation depends on the actual value of the first stage decisions (as in the recourse function in the introduction). Hence, in some sense, it is a local approximation.

Step 4: $\alpha^* \leq 0$ shows how much flow can be sent along the original arcs in the negative direction without making that total flow negative (whatever the supply/demand is). Similarly, β^* shows how much is left of the capacity in the arcs in the worst case.

x^* is just a circulation in the network, and x_k^* are circuits of minimal length (in terms of the number of arcs in them). α_k shows how much flow the circuit can take (or, more precisely, how much flow it has been allotted.)

Step 7: β_k **is a random variable describing the capacity of circuit** k.

3.4 Relation to Sublinear Approximations

The separable piecewise linear upper bound is also a generalisation of the ray function approximation in Birge and Wets [1986a] and its extension in the sublinear approximation in Birge and Wets [1986b]. These procedures find the value of $Q(\xi, \phi)$ in different coordinate directions to again obtain a separable function that can easily be integrated. The approach in Birge and Wets [1986b] uses varying choices of the coordinate system that leads to an extension of the SPLU bound given here. This extension would involve solving for x^{j+} and x^{j-} in different directions so that a variety of bounds could be obtained.

The ray function approximation amounts to solving for

$$q^T x^{j+} = \min\{q^T x \mid Ax = e^j, x \geq 0\}$$

and

$$q^T x^{j-} = \min\{q^T x \mid Ax = -e^j, x \geq 0\}.$$

These values of x^{j+} and x^{j-} are then used in $U(\xi, \phi)$ as in SPLU. The extension is to use the elements of other coordinate systems in place of $\pm e^j$ in the definitions (i.e., use some vectors d^j that form a basis for $\Re^n$). This procedure can be used in Algorithm 1 to obtain an alternative bound.

The sublinear approximation with varying directions has been found to produce accurate approximations in a variety of examples. The advantage of the SPLU bound is that it applies to bounded regions so it may be used on partitions of the support of the random variable in a refinement procedure in solving a stochastic program. Algorithm 1 also incorporates the procedures for handling random bounds that often arise in practical examples.

3.5 Finiteness

There is no guarantee that our upper bound is finite, i.e., that all linear programs that must be solved are feasible. An infinite bound of course results if $EQ(\xi, \phi) = +\infty$, i.e. the problem itself is infeasible, but

it can also be that $EQ(\xi,\phi) < +\infty$, whereas $EU(\xi,\phi) = +\infty$. This is not always avoidable. We note that the only other polynomial upper bound, the ray approximation, is never better than our bound (assuming the possible extensions mentioned above), and that exponential bounds may be necessary in some cases.

3.6 Partitioning

When approximations, such as the one in this paper, are used in two-stage stochastic programming, a comparison is made with a lower bound (EL) usually based on Jensen's inequality. Then, if EU - EL is too large (according to some rule), the support rectangle (for independent random variables) is partitioned into smaller rectangles called *cells*, and the bounding procedures are applied to these cells, which in turn are weighted by their probability.

Hence, whenever a partition is called for, one must decide which cell to partition and, along which coordinate direction, to perform the partition. With an upper bounding method that can take on the value $+\infty$ even for a feasible problem, one should clearly partition the cell where $EU = +\infty$, along the coordinate direction that was being treated when the infeasibility was discovered. This provides a dynamic scheme in which the algorithm is applied on each cell until either an infinite value is obtained or the difference between lower and upper bounds is above the acceptable threshold. A partition is made in either instance. Partition strategies are discussed in Birge and Wets [1986a], Birge and Wallace [1986] and Frauendorfer and Kall [1986].

4. Examples

In this section, we first present a small example to illustrate the bound. We then give computational results on a larger problem from energy modeling (Louveaux[1987]).

4.1 A Problem with Two Random Variables

The first example is a problem with two random variables and without random capacities. We wish to find bounds on $EQ(\xi)$ where

$$Q(\xi) = \min\ x_1 + x_2 + x_3 + x_4 + 10x_5 + 10x_6 \tag{4.1}$$

subject to

$$\begin{array}{rrrrrrrr} x_1 & +3x_2 & +x_3 & & -x_5 & & = \xi_1 & (4.2) \\ 3x_1 & +x_2 & & +x_4 & & -x_6 & = \xi_2 & (4.3) \end{array}$$

$$x_1, \ldots, x_6 \geq 0,$$

where ξ_1 and ξ_2 are uniformly distributed on $[1, 4]$. This problem is illustrated in Figure 1, where $A_{.i}$ refers to the ith column in the constraint matrix of (4.2-4.3). We follow Algorithm 1 step by step.

Step 0:

(i) Find $Q(\bar{\xi}) = 1.25$.

$x^0 = (0.625, 0.625, 0, 0, 0, 0)$.

$x^{1+} = (-0.125, 0.375, 0, 0, 0, 0); x^{1-} = (0.125, -0.375, 0, 0, 0, 0)$.

$x^{2+} = (0.375, -0.125, 0, 0, 0, 0); x^{2-} = (-0.375, 0.125, 0, 0, 0, 0)$.

$\alpha^1(1) = -0.0625; \alpha^1(2) = -0.4375$;(Note : $\beta^r(i) = +\infty$.)

Now $e(1) = 0.1875 > -\alpha^1(1) = 0.0625$, so go to Step 1. Note in Figure 1 that we have essentially moved along the vertical line through $\bar{\xi}$. The bound α^1 recorded the (negative) minimum multiples of the vectors $A_{.1}$ and $A_{.2}$ for points along that line. The value $e(1)$ recorded the greatest change in the multiple of $A_{.1}$ from the multiple for $\bar{\xi}$ for other points along the horizontal line through $\bar{\xi}$. The function is not linear because this change is greater than the minimal multiple ($\alpha_1(1)$) for movement in the vertical direction.

Step 1.

Solve $\min\{q^T x \mid \chi[1.5e_1, 0, (-0.0625, -0.4375, 0, 0, 0, 0), \infty)]\} = 1.125 = (0.75) * (1.5) = q^T x^{1+}(\xi_1^{\max} - \bar{\xi}_1)$, where $x^{1+} = (-0.0625, 0.1875, 1.0, 0, 0, 0)$, and

$\min\{q^T x \mid \chi[-1.5e_1, 0, (-0.0625, -0.4375, 0, 0, 0, 0), \infty)]\} = 1.375 = (0.9167) * (1.5) = q^T x^{1-}(\bar{\xi}_1 - \xi_1^{\min})$, where $x^{1-} = (-0.0625, -0.4375, 0, 0.625, 0.125, 0)$. Next go to Step 4.

Step 4:

We can skip this since there are no random bounds. Step 5 then is the terminal step.

Step 5:

Here we compute $EU(\xi) =$

$$Q(\bar{\xi}) + \int_{\xi_1 \geq \bar{\xi}_1} x^{1+}(\xi_1 - \bar{\xi}_1) dF(\xi_1) + \int_{\xi_1 < \bar{\xi}_1} x^{1-}(\bar{\xi}_1 - \xi_1) dF(\xi_1)$$

$$+ \int_{\xi_2 \geq \bar{\xi}_2} x^{2+}(\xi_2 - \bar{\xi}_2) dF(\xi_2) + \int_{\xi_2 < \bar{\xi}_2} x^{2-}(\bar{\xi}_2 - \xi_2) dF(\xi_2)$$

$$= 1.25 + 0.5(0.75)(0.75) + (0.5)(0.9167)(0.75) + 0.5(0.25)(0.75) + (0.5)(-0.25)(0.75) = 1.875.$$

End.

So, we have SPLU = 1.875. We compare this with the Edmundson-Madansky (EM) bound. In this example, the EM bound assigns equal weights to the values of $Q(\xi)$ at each of the extreme points of Ξ. Hence,

$$EM = 0.25 * (Q(1,1) + Q(1,4) + Q(4,1) + Q(4,4)) = 1.625.$$

The EM bound is better than the SPLU bound but this difference may be eliminated by refinements of the SPLU bound. We describe possible refinements in Section 5.

4.2 Computational Results for an Energy Model

The usefulness of the SPLU bound is best demonstrated on a practical example in which the number of random variables varies. We wish specifically to observe the performance of SPLU relative to the EM bound as the number of random variables increases. The performance is measured in the sharpness of the bound and the computational effort. As a practical example, we consider the small energy model in Louveaux [1987]. We do not consider random bounds because that is directly analogous to the network case discussed in Wallace [1987b].

In this example, we have four technologies which can be used to satisfy three demands at varying costs. High cost "backstop" technologies are also available to satisfy demand so the problem is feasible for any demand realisation. The randomness occurs in the capacity of the technologies and the demands. This allows from one to seven random variables. The examples were also chosen with varying ranges (narrow,medium, and wide) on the random variables resulting in twenty-one sets of examples. We assume uniform distributions. This assumption favors bounds (such as the Edmundson-Madansky bound) that place weights at extreme values since other distributions generally have more mass around the center of the support.

The experiments were conducted on the Amdahl 5860 at The University of Michigan Computing Center. The SPLU and EM bounds were both implemented in FORTRAN codes using the same linear programming routine LPM-1 (Pfefferkorn and Tomlin [1976]). Each bound was computed for each of the twenty-one test problems. The Jensen inequality lower bound was also computed to determine the values of the upper bounds relative to the lower bounds. The results are given in Table 1.

The results in Table 1 show that the polynomial bound SPLU does not generally provide as accurate a bound as the EM bound, but that as the number of random variables increases the computational time in SPLU increases much less rapidly than the time for EM. In these examples, the growth of time for SPLU is indeed approximately linear (gaining ten milliseconds for each random variable), while the time for EM approximately doubles as each new random variable is introduced. This demonstrates that the real advantage of the SPLU bound is in problems with a large number of random variables where the EM bound cannot be computed. These results are comparable with the results in Wallace [1987b] for networks.

Refinements are also possible to reduce the error in SPLU. In Section 5, a refinement scheme using parametric linear programming is introduced. The use of different coordinate directions is another possibility as mentioned above. For unbounded ranges, the resulting sublinear approximation values were reduced up to thirty percent from the coordinate direction values for a similar set of test problems (Birge and Wets [1986b]).

PROBLEM[1]	TIME[2]		VALUE		
	EM	SPLU	EM	SPLU	Jensen
1-NAR	9	10	182.75	182.75	182.75
1-MED	10	18	220.50	220.25	220.00
1-WID	10	21	385.50	341.50	297.50
2-NAR	16	20	183.38	183.06	182.75
2-MED	17	26	220.50	220.50	220.00
2-WID	22	35	389.85	389.10	297.50
3-NAR	22	28	183.38	183.38	182.75
3-MED	25	38	221.38	222.50	220.00
3-WID	33	40	433.60	439.58	297.50
4-NAR	51	41	184.09	185.50	182.75
4-MED	44	41	227.22	255.18	220.00
4-WID	47	45	434.26	469.50	297.50
5-NAR	94	52	184.19	186.44	182.75
5-MED	87	51	227.41	278.38	220.00
5-WID	75	54	434.35	499.18	297.50
6-NAR	163	54	185.58	192.49	182.75
6-MED	193	61	235.91	303.35	220.00
6-WID	149	72	443.52	524.30	297.50
7-NAR	329	64	186.23	215.48	182.75
7-MED	366	70	236.27	328.35	220.00
7-WID	304	76	444.12	549.30	297.50

1 - Number of random variables - range of random variables.

2 - CPU milliseconds.

Table 1. Results for Edmundson-Madansky and SPLU bounds.

5. Extensions and Conclusions

The SPLU bound can be refined in a variety ways. The use of other coordinate directions may be possible, but it is best used when linear transformations of the random variables have a known distributional form as is the case for normally distributed random variables. As mentioned above, a common procedure is to partition the support of the random variables and to apply the bound on each of the partitions. Here we give a parametric programming approach that can obtain more accurate results without partitioning the random variables. The following modifications of Algorithm 1 provide this basic bound.

Algorithm 2

Substitute the following steps into Algorithm 1 to obtain

$$U'(\xi,\phi) = Q(\bar{\xi},0) + H(\phi) + \sum_{\ell_i \geq (<) \bar{\ell}} f_i^{+(-)}((-)\xi_i - (+)\bar{\xi}_i).$$

Step 1':Solve the parametric linear program

$$\min\{q^T x \mid \chi(\epsilon e_r, 0, \alpha^r, \beta^r)\}$$

for $\epsilon \in [0, \xi_r^{\max} - \bar{\xi}_r]$ (or $\epsilon \in [0,\infty)$ if ξ_r is unbounded). This generates a piecewise linear function $f_r^+(\epsilon e_r)$ with break points $\{0, \epsilon_1, \ldots, \epsilon_T\}$, and with slope values, $q^T x_1^{r+}, \ldots, q^T x_T^{r+}$.

Then,solve the parametric linear program

$$\min\{q^T x \mid \chi(-\epsilon e_r, 0, \alpha^r, \beta^r)\}$$

for $\epsilon \in [0, \bar{\xi}_r - \xi_r^{\min}]$ (or $\epsilon \in [0,\infty)$ if ξ_r unbounded.) We then obtain a piecewise linear function f_r^- with breakpoints $\{0, \epsilon^1, \ldots, \epsilon^T\}$, and with slope values, $q^T x_1^{r-}, \ldots, q^T x_T^{r-}$.

Step 2' and Step 4': Substitute $\min_t\{x_t^{j+}(i)\}$ for $x^{j+}(i)$ and $\min_t\{x_t^{j-}(i)\}$ for $x^{j-}(i)$ in the definitions of $\alpha^{r+1}(i)$ and $\alpha^*(i)$ and substitute $\max_t\{x_t^{j+}(i)\}$ for $x^{j+}(i)$ and $\max_t\{x_t^{j-}(i)\}$ for $x^{j-}(i)$ in the definitions of $\beta^{r+1}(i)$ and $\beta^*(i)$.

The changes in Step 1' of Algorithm 2 lead to better bounds if α^r and β^r are the same and $q^T x_1^r < q^T x_T^r$. In Algorithm 2, the approximation obtains as low a value as possible for all ξ_r for changes in the rth direction

given the values found for movement in previous directions. In Algorithm 1, the approximation just uses the extreme values of ξ_r.

This difference can be seen in the example from Section 4.1 which is illustrated in Figure 2. The dashed line corresponds to the function used in Algorithm 1 by using the extreme values. The solid line corresponds to the functions f_r^+ and f_r^-. The new bound is

$$\text{SPLU}' = \text{E}[U'(\xi, \phi)] = 1.449.$$

We note that SPLU' is now below the EM bound value of 1.625.

The bound from Algorithm 2 is not always better than SPLU because the bounds may change for different values of r, i.e. α^{r+1} may increase and β^{r+1} may decrease. Although this difference appears to rarely make SPLU' worse than SPLU according to our limited conmputational experience, it may be advantageous to guarantee that a bound at least as good as SPLU is obtained. This guarantee is accomplished in the following modification of Step 1'.

Step 1''. Solve

$$\min\{q^T x \mid \chi((\xi_r^{\max} - \bar{\xi}_r)e_r, 0, \alpha^r, \beta^r\}$$

$$= q^T \bar{x}.$$

Let

$$\alpha_*^r(i) = \begin{cases} \bar{x}(i) & \text{if } \bar{x} < 0, \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\beta_*^r(i) = \begin{cases} \bar{x}(i) & \text{if } \bar{x} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Then solve the parametric linear program

$$\min\{q^T x \mid \chi((\epsilon e_r)e_r, 0, \alpha_*^r, \beta_*^r)\}$$

to obtain $f_r^+(\epsilon e_r)$ as explained in Step 1'.

This modification of Algorithm 2 results in bounds that are at least as sharp as SPLU and can still benefit from the parametric program as in the example given above. The key benefit of the SPLU bound that the computational effort only grows polynomially with increases in the number of random variables is maintained. We have demonstrated how this improvement results in reduced times on one set of examples and that the greatest value of the SPLU bound may be in cases where the EM and other exponential bounds cannot be reasonably computed. The refinements mentioned above may allow the SPLU bound to be even more useful in the solution of practical stochastic linear programming problems.

References

Ben-Tal, A. and E. Hochman [1972],"More bounds on the expectation of a random variable", *Journal of Applied Probability* 9, 803–812.

Birge, J.R. [1987],"Exhaustible resource models with uncertain returns from exploration investment", in: **Numerical Methods in Stochastic Programming**, R. Wets and Y. Ermoliev, eds., Chapter 27, to appear.

Birge, J.R. and S.W. Wallace [1986],"Refining bounds for stochastic linear programs with linearly transformed independent random variables", *Operations Research Letters* 5, 73-77.

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

Birge, J.R. and R.J-B. Wets [1986b],"A sublinear approximation method for stochastic programming", Technical Report 86-24, Department of Industrial and Operations Engineering, The University of Michigan.

Birge, J.R. and R.J-B.Wets [1987],"Computing bounds for stochastic programming problems by means of a generalised moment problem", *Mathematics of Operations Research*, to appear.

Cipra, T. [1985],"Moment problem with given covariance structure in stochastic programming", *Ekonomicko-matematický obzor*, 21, 66–77.

Dupačová, J. [1974], "Minimax stochastic programs with nonconvex nonseparable penalty functions", in: *Colloquia Mathematica János Bolyai*, A. Prékopa, ed., North-Holland, pp. 303–316.

Edmundson, H.P. [1956],"Bounds on the expectation of a convex function of a random variable", The Rand Corporation, Paper 982.

Ermoliev, Y., A. Gaivoronsky and C. Nedeva [1987],"Stochastic optimisation problems with partially known distribution functions", *SIAM Journal on Control and Optimization*, to appear.

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

Frauendorfer, K. [1986],"Solving S.L.P. recourse problems with arbitrary multivariate distributions - the dependent case", Institut für Operations Research, Universität Zürich, Technical Report.

Frauendorfer,K. and P. Kall [1986],"A Solution Method for SLP recourse problems with arbitrary multivariate distributions - the independent case", Institut für Operations Research der Universität Zürich, Technical Report.

Gassmann, H. and W. T. Ziemba [1986],"A tight upper bound for the expectation of a convex function of a multivariate random variable", *Mathematical Programming Study* 27, 39–53.

Hausch, D. and W.T. Ziemba [1983], "Bounds on the value of information in uncertain decision problems II", *Stochastics* 10, 181-217.

Huang, C., W. Ziemba, and A. Ben-Tal,"Bounds on the expectation of a convex function of a random variable: with applications to stochastic programming", *Operations Research* 25, 315-325.

Jensen, J.L. [1906],"Sur les fonction convexes et les inégalités entre les valeurs moyennes", *Acta Mathematica* 30, 175–193.

Kusy, M. and W.T. Ziemba [1986],"A bank asset and liability management model", *Operations Research* 34, 356-376.

Louveaux, F. [1987],"Optimal investment for electricity generation: a stochastic model and a test problem", in: **Numerical Methods in Stochastic Programming**, R. Wets and Y. Ermoliev, eds., to appear.

Madansky, A. [1959],"Bounds on the expectation of a convex function of a multivariate random variable",*Annals of Mathematical Statistics* 30, 743–746.

Pfefferkorn, C.E. and J.A. Tomlin [1976],"Design of a linear programming system for ILLIAC IV", Technical

Report SOL 76-8, Systems Optimization Laboratory, Stanford University.

Prékopa, A. and T. Szantai [1978], "Flood control reservoir system design using stochastic programming", *Mathematical Programming Study* 9, 138–151.

Rockafellar, T.R. [1984], **Network Flows and Monotropic Programming**, Wiley, New York.

Wallace, S.W. [1987a], "Investing in arcs in a network to maximise the expected max flow", *Networks*, to appear.

Wallace, S.W. [1987b], "A piecewise linear upper bound on the network recourse problem", *Mathematical Programming*, to appear.

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

Žáčková, J. [1966], "On minimax solutions of stochastic linear programming problems", *Časopis pro Pěstování Matematiky* 91, 423–430.

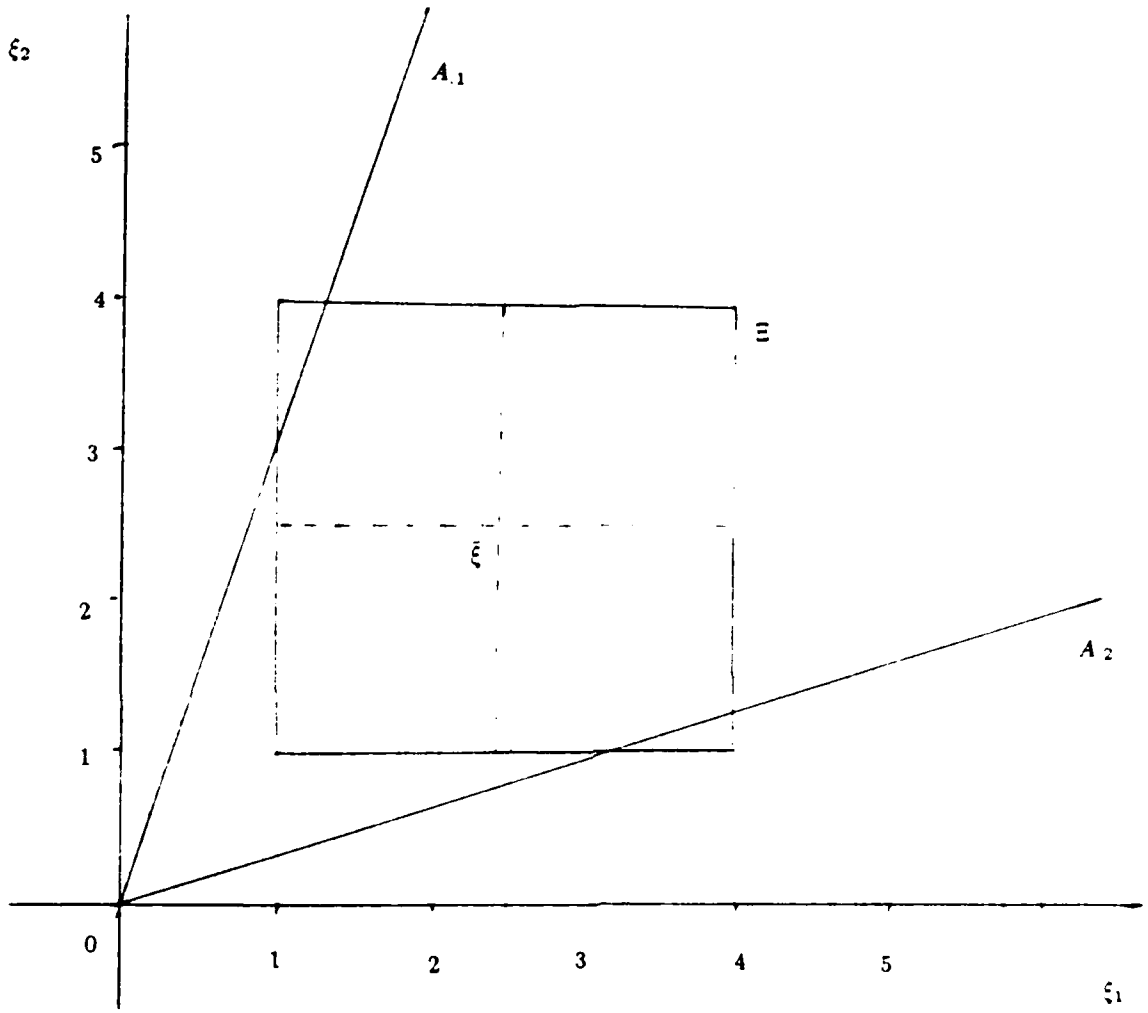


Figure 1. Example with two random variables.

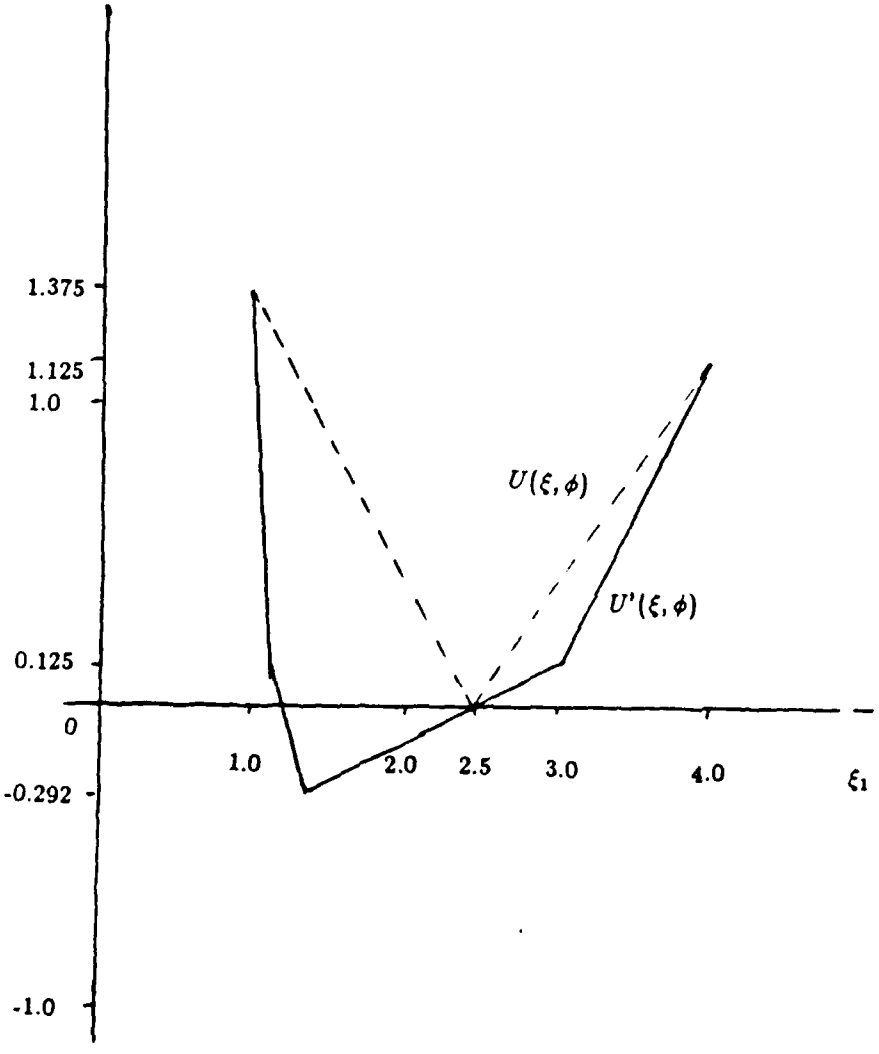


Figure 2. Parametric linear program bound.

DISTRIBUTION LIST

	NO OF COPIES
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Library (Code 0142) Naval Postgraduate School Monterey, CA 93943-5000	2
Director of Research Administration Naval Postgraduate School Monterey, CA 93943-5000	1
Library (Code 55) Naval Postgraduate School Monterey, CA 93943-5000	1
Center for Naval Analyses 2000 Beauregard Street Alexandria, VA 22311	1
Operations Research Center, Room E40-164 Massachusetts Institute of Technology Attn: R. C. Larson and J. F. Shapiro Cambridge, MA 02139	1
Koh Peng Hong Ministry of Defence Blk A, Stockport Road Singapore 0511	1

END

4-87

DTIC