An Exact Penalty Algorithm for Recourse-Constrained Stochastic Linear Programs

Diana Yakowitz

USDA-ARS Southwest Watershed Research Center
2000 E. Allen Road
Tucson, Arizona 85719

ABSTRACT

This paper presents an algorithm for solving stochastic linear programs in which a recourse problem that appears in the constraint set. This algorithm, which is based on a stochastic decomposition (SD) algorithm of Higle and Sen, involves the use of an exact penalty function in the master program. Results from applications to several problems are reported.

1. INTRODUCTION

Stochastic linear programming problems are linear programming problems for which one or more data elements are described by random variables. Stochastic linear programming problems with recourse are problems in which a first-stage decision is made before the random variables are observed. A recourse decision that varies with these observations compensates for any deficiencies that result from the earlier decision. Application areas in which stochastic linear programming problems with recourse have arisen include water resources, waste management, economics, and finance.

Stochastic programs with recourse are generally large-scale optimization problems. Few algorithms address such general large-scale optimization programs and computational implementation and testing of these algorithms have been limited mainly to simple problems with small variable dimension. In certain instances the probability distributions may be unknown and observational data may be the only information available. The algorithm introduced in this paper can be used even in such a case.
A formulation for a two-stage stochastic linear program with recourse is as follows.

$$\min f(x) = c^T x + E[Q(x, \bar{\omega})]$$

s.t. $Ax \leq b$

where

$$Q(x, \bar{\omega}) = \min q^T y$$

s.t. $Wy = \bar{\omega} - Tx, \quad y \geq 0.$

Here $A$ is a known $m_1 \times n_1$ matrix, $c$ and $b$ are known vectors in $R^{n_1}$ and $R^{m_1}$, respectively. $q$ is a known vector in $R^{n_2}$. $W$ and $T$ are known $m_2 \times n_2$ and $m_2 \times n_1$ matrices respectively. The random vector, $\bar{\omega} \in R^{m_2}$, is defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. $E[\cdot]$ represents the expectation with respect to $\bar{\omega}$.

A decision $x$ is made before the random variable $\bar{\omega}$ is realized. One desires a first-stage decision, $x$, which minimizes the sum of the current costs and the expected value of the second-stage cost. Properties of stochastic programs with recourse are examined in [1].

The evaluation of $E[Q(x, \bar{\omega})]$ for fixed $x = \bar{x}$ involves the computation of a multidimensional integral. Even in two dimensions the difficulty in evaluating the integral makes the use of standard nonlinear programming methods ([2], [3]) impractical.

Methods for the solution of stochastic programs with recourse have mainly developed along two fronts. One approach involves a reduction of the problem to a deterministic optimization problem ([4], [5]). The second approach involves statistical approximations such as in a procedure known as the stochastic quasigradient method [6], in which the gradient of the objective function is statistically approximated.

The stochastic decomposition algorithm (SD) for solving two-stage stochastic programs with recourse [7] combines features of these two approaches. Like decomposition-based algorithms, SD produces a piecewise linear approximation of the objective function. However, the function approximation is statistical in nature and only one subproblem is solved in each iteration as in stochastic quasigradient methods. The SD algorithm was modified and extended to be more adaptive and computationally efficient in [8] with the introduction of a quadratic term in the objective function (hence the acronym QSD) and a cut dropping scheme that bounds the size of the master program in each iteration.
In this work, the SD concept is extended by developing a solution method for a model in which the recourse function appears in the constraint set rather than in the objective function.

With any decision that involves large amounts of capital, one is concerned with the risk involved. The general two-stage formulation for stochastic programs with recourse just presented does not take this into account. One way to take risk into account is to devise a risk measure and append a constraint limiting its magnitude. Risk can be limited, for example, by imposing an upper bound on the expected second-stage cost. This is the approach presented in this paper. Measures of risk that appear in the literature include model variance (see [15] and [9]) and downside risk [10]. The former approach results in quadratic constraints, thus making the problem quite difficult to solve. Additionally, in [10] it is pointed out that with a variance-based measure, points on the "mean-variance efficient frontier" may be stochastically dominated.

This paper is organized as follows. The model and solution method is revealed in Section 2 with the algorithm steps listed in Section 3. Convergence results are presented in Section 4 and stopping criteria are suggested in Section 5. Results of applying the algorithm to several problems in the literature are presented in Section 6.

2. DESCRIPTION OF THE METHOD

Suppose the decision maker wants to minimize the first-stage costs while keeping the expected second-stage costs below a target budget. For example, the decision maker accepts that there could be certain instances of the demand for the products he produces that will cause him to exceed his budget for a chosen decision \( x \) but the decision should be such that on average this is not the case. This model would yield the following formulation of a two-stage stochastic program with recourse:

\[
(P) \quad \begin{align*}
\min & \quad c x \\
\text{s.t.} & \quad x \in X, \\
& \quad E[Q(x, \omega)] \leq b,
\end{align*}
\]  

(2.1)

where \( X \) is a convex polyhedral set and \( Q(x, \omega) \geq 0 \) for all feasible \( x \) and \( \omega \) is defined as

\[
Q(x, \omega) = \min q y \\
\text{s.t.} \quad Wy = \omega - Tx, \quad y \geq 0.
\]

The matrices and vectors are defined as in Section 1.
The solution method proposed for the above problem is based on SD and the penalty function methods in nonlinear programming (see [3] and [11]). The difficult constraint, \( E[Q(x, \omega)] \leq b \), is handled by introducing an exact penalty function based on this constraint. The penalty function is then appended to the objective function weighted by a scalar, \( \sigma \), resulting in the following problem.

\[
\min_{x \in X} P(x) = \sigma c x + \max(0, E[Q(x, \omega)] - b).
\]

The algorithm proposed to solve the penalized program produces a sequence \( \{x_k\}_{k=1}^\infty \) of "incumbent" solutions and a sequence, \( \{z_k\}_{k=1}^\infty \), called "candidates." In iteration \( k \), \( z_k \) is determined through the solution of a linear master program. Beginning with an initial incumbent solution, a candidate is accepted as the next incumbent if its estimated penalized objective value is sufficiently lower than that of the current incumbent solution.

Given \( z_k \) and an observation \( \omega_k \) of \( \bar{\omega} \), we solve subproblem \( (S_k) \) defined as follows:

\[
(S_k) \quad Q(z_k, \omega_k) = \min q \ y \\
\text{s.t.} \quad Wy = \omega_k - Tz_k, \quad y \geq 0.
\]

The dual to the above problem is

\[
Q(z_k, \omega_k) = \max \pi(\omega_k - Tz_k) \\
\text{s.t.} \quad \pi \in \Pi = \{\pi: \pi W \leq q\}.
\]

The random variable \( \bar{\omega} \) is defined on a probability space \( (\Omega, \mathcal{A}, \mathcal{P}) \) with associated distribution function \( F_\bar{\omega} \). To ensure that \( (P) \) has the complete recourse property it is assumed that \( X \) and \( \Pi \) are compact convex polyhedral sets. Thus, \( |Q(x, \omega)| < \infty \) for all \( (x, \omega) \in X \times \Omega \).

The sequences \( \{x_k\} \), \( \{z_k\} \), and \( \{\omega_k\} \) are used to approximate the recourse function, \( E[Q(x, \bar{\omega})] \). Thus, at iteration \( k \), let \( x_j^k + x^k \) be an estimate of a linear support of \( E[Q(x, \bar{\omega})] \) at \( z_j \), \( 1 \leq j \leq k \) and let \( \sigma_k > 0 \) be the current penalty coefficient. The master penalty program is given by:

\[
(M^k) \quad \min \{P_k(x) = \sigma_k c x + \eta_k(x): x \in X\}
\]
where

\[ \eta_k(x) = \max\{0, \alpha^I_k + \beta^I_k x - b, 1 \leq j \leq k\}. \] (2.2)

The solution to (M^k) is denoted by \( z_{k+1} \). Let \( V \) denote the set of all extreme point solutions of the subproblem dual, and let \( V_k \subseteq V \) be the set of extreme points of the dual of (S^k) generated in the first \( k \) iterations of the algorithm. For \( t = 1, 2, \ldots, k \), choose \( \pi^k_t \) to satisfy

\[ \pi^k_t \in \text{argmax}\{ \pi(\omega_t - Tz_k) : \pi \in V_k \}. \] (2.3a)

The estimated support of \( E[Q(x, \omega)] \) generated at \( z_k \), in the \( k \)th iteration, is given by

\[ \alpha^k + \beta^k x = \frac{1}{k} \sum_{t=1}^{k} \pi^k_t (\omega_t - Tx). \] (2.3b)

The coefficients of the cuts generated in iterations \( j = 1, 2, \ldots, k - 1 \) are updated adaptively in iteration \( k \) as in the QSD algorithm in [8] as follows:

\[ \alpha^k_i = \frac{k-1}{k} \alpha^k_{i-1} + \frac{1}{k} \pi^k_i \omega_k \] (2.4a)

\[ \beta^k_i = \frac{k-1}{k} \beta^k_{i-1} - \frac{1}{k} \pi^k_i T. \] (2.4b)

Here the superscripts indicate association with the candidate \( z_j \) and the subscripts correspond to the current iteration \( k \).

An incumbent solution need not change over many iterations. Thus, let \( \gamma_j \) denote the iteration when the incumbent solution, \( x_j \), was accepted. Then \( \alpha^{k-1}_k + \beta^{k-1}_k x \) is the updated cut associated with the incumbent solution, \( x_{k-1} \). Also let \( \tau_{k-1} \) indicate the iteration at which the \( k - 1 \)st cut associated with the incumbent solution was last evaluated and \( \tau \) be a given integer. The cut associated with the incumbent solution is reevaluated whenever one of the following is satisfied:

\[ \alpha^k + \beta^k x_{k-1} - (\alpha^{\gamma_{k-1}} + \beta^{\gamma_{k-1}} x_{k-1}) > 0, \] or \[ k - \tau_{k-1} = \tau. \] (2.5a)
Reevaluation of this cut is accomplished by using the current set of subproblem dual vectors, \( V_k \). That is, for \( t = 1, 2, \ldots, k \), determine

\[
\pi^\gamma t \in \arg\max \{ \pi (x_t - Tx_{k-1}) : \pi \in V_k \},
\]

(2.6a) and

\[
\alpha_{k-1}^\gamma + \beta_k^\gamma x = \frac{1}{k} \sum_{t=1}^{k} \pi_t^\gamma (x_t - Tx).
\]

(2.6b)

With the support estimates and updating mechanism defined, we proceed with the description of the method. Suppose \((M^{k-1})\) has just been solved and now at iteration \( k \) we wish to determine the \( k \)th penalty parameter, \( \sigma_k \), and whether \( z_k \) becomes the \( k \)th incumbent. We determine \( \sigma_k \) as follows:

\[
\sigma_k = \begin{cases} \\
\frac{k}{k+1} \sigma_{k-1}, & \text{if } \max \{ \eta_k(z), \eta_k(x_{k-1}) \} > 0; \\
\sigma_{k-1}, & \text{otherwise}.
\end{cases}
\]

(2.7)

Therefore, \( \sigma_k \) remains unchanged if feasibility to \((P)\) is suggested, and reduced, thus damping the effect of the term \( c x \), otherwise. In either case \( \{ \sigma_k \}_{k=1}^\infty \) is a non-negative, monotonic, nonincreasing sequence, and thus \( \sigma_k \to \sigma \geq 0 \). If \( \sigma > 0 \), then both \( \{ z_k \}_{k=1}^\infty \) and \( \{ x_k \}_{k=1}^\infty \) accumulate at feasible solutions, with probability 1. Feasibility in this case and of accumulation points of \( \{ x_k \}_{k=1}^\infty \) when \( \sigma = 0 \) will be addressed in Section 4.

The updated piecewise linear approximation, \( P_k(x) \), is now given by

\[
P_k(x) = \sigma_k c x + \eta_k(x),
\]

and \( x_k \) is set to \( z_k \) if

\[
P_k(z_k) - P_k(x_{k-1}) < \mu (P_{k-1}(z_k) - P_{k-1}(x_{k-1})).
\]

(2.8)

The parameter \( \mu \) is fixed such that \( 0 < \mu < 1 \). Satisfaction of (2.8) implies that a sufficient fraction of the anticipated penalized objective value reduction is attained. In such cases \( z_k \) is designated as the new incumbent \( x_k \). Otherwise, the incumbent does not change and \( x_k = x_{k-1} \).

Dropping constraints in the master program can be accomplished in the following manner. Replace (2.2) with

\[
\eta_k(x) = \max \{ 0, \alpha_k^j + \beta_k^j x - b, j \in J_k \},
\]

(2.9)
where $J^k \subseteq \{1, 2, \ldots, k\}$ is determined as follows: Let

$$\hat{J}^{k-1} = \{ j \in J^{k-1} : \lambda_{k-1}^j > 0 \},$$

(2.10)

where $\lambda_{k-1}^j$ are the dual multipliers associated with the constraints $\eta_k(x) \geq \alpha_{k-1}^j + \beta_{k-1}^j x - b$, $j \in J^{k-1}$ present in master program $(M^{k-1})$. Owing to the definition of $\eta_k(x_k)$ in (2.9), it is possible that none of the cuts added to the master program has a positive dual multiplier. Therefore, it is advisable to retain as many as $n_1 + 1$ of the most recent cuts including the cut indexed by $\gamma_k$, such that $\alpha_k^j + \beta_k^j x_k = \alpha_k^\gamma + \beta_k^\gamma x_k$, $j \in J^{k-1}$. If this set is denoted by $I$ then

$$J^k = \hat{J}^{k-1} \cup I \cup \{ k \}.$$  

(2.11)

Thus, if $n_1$ is the dimension of $x$, at most $2n_1 + 3$ are retained in each iteration.

3. THE SD EXACT PENALTY ALGORITHM

The SD Exact Penalty Algorithm, as introduced in Section 2, will now be summarized. A brief discussion of the major steps follows the algorithmic statement. Adapting the algorithm to include cut dropping is then presented.

**Algorithm.** Exact Penalty Stochastic Decomposition

**Step 0.** (Initialize)

$k \leftarrow 0$, $\sigma_0 \geq 1.0$ is given, $\omega_0 \leftarrow E[\hat{\omega}]$, $x_0 \in \text{argmin}\{ P_0(x) = \sigma_0 c x + \max\{0, Q(x, \omega_0) - b\} : x \in X\}$, $z_1 \leftarrow x_0$, $J^0 = \{ \emptyset \}$, $\eta_0(x_0) = \max\{0, Q(x_0, \omega_0) - b\}$, $\gamma_0 \leftarrow 0$, $V_0 = \{ \emptyset \}$, $\tau_0 = 0$, $0 < \mu < 1$, and $\tau$ is given.

**Step 1.** (Generate a random vector)

$k \leftarrow k + 1$. Randomly generate an observation, $\omega_k$, according to its distribution, $F_\omega$ ($\omega_t$, $t = 1, 2, \ldots, k$, are presumed generated independently).

**Step 2.** (Solve subproblem)

Solve $S^k$ and obtain the dual vector $\pi(z_k, \omega_k)$. $V_k \leftarrow V_{k-1} \cup \{ \pi(a_k, \omega_k) \}$. 


Step 3. (Update master program)
Evaluate cut $k$ according to (2.3) and for $j \in \mathcal{J}^{k-1}$ determine the updated cut coefficients according to (2.4).

Step 4. (Update penalty)
Determine $\sigma_k$ by (2.7).

Step 5. (Reevaluate cut associated with incumbent)
Check reevaluation conditions (2.5). If satisfied, redetermine $\alpha_k^{T_k-1}$, and $\beta_k^{T_k-1}$ according to (2.6).

Step 6. (Check for new incumbent)
(a) If (2.8) is satisfied then $x_k \leftarrow z_k$, $\gamma_k \leftarrow k$. Otherwise,
(b) $x_k \leftarrow x_{k-1}$, $\gamma(k) \leftarrow \gamma_{k-1}$.
In either case, determine $\mathcal{J}^k$ according to (2.11).

Step 7. (Solve master program)
Solve $(M^k)$ to obtain $z_{k+1}$ and $\eta_k(z_{k+1})$ and the dual variables $\lambda^j_k$, $j \in \mathcal{J}^k$.
Return to Step 1.

The algorithm proceeds as follows: Initialize in Step 0 by finding an initial first-stage decision $x_0$. The vector $\omega_k$ is generated in Step 1 according to the distribution $F_\omega$ and the recourse problem is solved at the candidate point, $z_k$, in Step 2. This yields a dual vector which is then stored in $V_k$. A dual variable in $V_k$ is then associated with each $\{\omega_i\}_{i=1}^k$. An estimate of a support of $E[Q(x, \omega)]$ at the candidate point, $z_k$, is then determined and the past cuts remaining are updated in Step 3.

The penalty, $\sigma_k$, is determined in Step 4 by (2.7). If infeasibility of the incumbent or candidate point is detected, then the penalty parameter is reduced.

In Step 5, (2.5) is used to ascertain whether a better estimate of the cut associated with the incumbent solution should be made using the current set of dual solutions to the recourse problem, $V_k$. The incumbent is updated in Step 6. When (2.8) is satisfied, $z_k$ is accepted as the new incumbent. If (2.8) is not satisfied, the incumbent remains the same. Cut dropping is then performed and set $\mathcal{J}^k$ is determined according to (2.11). Problem $(M^k)$ is solved in Step 7 to produce a new candidate point, and the process repeats.

4. CONVERGENCE ANALYSIS

The cuts generated by an SD-type algorithm underestimate, asymptotically, the expected value of the recourse function by [7]. Thus, by replacing $E[Q(x, \omega)] - b$ with $\max_{1 \leq j \leq k} (\alpha^j_k + \beta^j_k x) - b$ we obtain,
asymptotically, a relaxation of $(\mathcal{P})$ as $k \to \infty$. This fact is exploited in the proposed method.

In this section we will show that if $(\mathcal{P})$ is feasible, then under certain circumstances to be specified, every accumulation point of the sequence of incumbent solutions, $\{x_k\}$, is an optimal solution of $(\mathcal{P})$.

Since each of the cuts added to the master program in the limit underestimates the recourse constraint, they form an asymptotic relaxation of problem $(\mathcal{P})$. Just as in deterministic optimization, a solution to a relaxed optimization program is a solution to the original if it is feasible to the original problem.

Let $\overline{X}$ be the set of accumulation points of the sequence of incumbent solutions, $\{x^*_k\}$, and let $\bar{x}$ be the unique limit of $\{x_k\}$ for $k \to \infty$. Let $\{k_n\}_{n \in N}$ represent the sequence of iterations at which the incumbent is changed. That is, for every $n \in N$, (2.8) is satisfied and $x_{k_n} = z_{k_n}$. We also assume that cuts are dropped and thus $\eta_k(x)$ is defined by (2.9) and $J_k$ is defined by (2.11).

The updating and reestimation mechanisms of this penalty algorithm are the same as those in QSD. The development in [7] (Theorem 2, Corollary 5) yields the following.

**Lemma 4.1.** Let $\{x_{k_n}\}_{n=1}^\infty$ be a subsequence of $\{x_k\}_{k=1}^\infty$ such that $\{x_k\}_{n=1}^\infty \to x_\infty$. With probability 1,

$$\alpha_{k_n}^\gamma + \beta_{k_n}^\gamma x_{k_n} \to E[Q(x_\infty, \bar{\omega})].$$

Furthermore, with probability 1, any accumulation point of $\{(\alpha_{k_n}^\gamma, \beta_{k_n}^\gamma)\}_{k=1}^\infty$ defines a support of $E[Q(x, \bar{\omega})]$ at $x_\infty$ and

$$\lim_{n \to \infty} \eta_{k_n}(x_{k_n}) = \lim_{n \to \infty} \eta_{k_n+1}(x_{k_n}) = \max \{0, E[Q(x_\infty, \bar{\omega})] - b\}. $$

With regard to the asymptotic behavior of $\{x_k\}_{k=1}^\infty$, it has been established that $\bar{\sigma} > 0$ and $\bar{\sigma} = 0$ form two distinct cases of interest. In addition, it will be shown that finiteness of the set $N$ also provides two distinct cases of interest. In this section, with cut dropping in force, it is shown that if

C1. $N$ is an infinite set, or
C2. $N$ is a finite set and $\sigma_k \to \bar{\sigma} > 0$,

then every accumulation point of $\{x_k\}$ is an optimal solution to $(\mathcal{P})$, with probability 1. The remaining case, namely when $N$ is a finite set and
\( \bar{\sigma} = 0 \), remains an open theoretical question if cuts are dropped. In Section 6, an empirical investigation of this case is made.

We begin by establishing feasibility of the accumulation points of \( \{ x_k \} \) under cases C1 and C2 as defined above.

**Lemma 4.2.** Let \( x_{\infty} \in \bar{X} \). If there exists an \( \bar{x} \in X \) such that \( E[Q(\bar{x}, \bar{\omega})] \leq b \) and either C1 or C2 hold, then \( E[Q(x_{\infty}, \bar{\omega})] \leq b \) with probability 1.

**Proof.** Let \( \bar{x} \in X \) be such that \( E[Q(\bar{x}, \bar{\omega})] \leq b \). Note that \( \bar{X} \) is closed. Therefore, let \( x_{\infty} \in \bar{X} \) such that

\[
E[Q(\bar{x}, \bar{\omega})] = E[Q(x_{\infty}, \bar{\omega})] = b.
\]

Suppose \( E[Q(x_{\infty}, \omega)] = b + \epsilon > b \). That is, \( x_{\infty} \) is infeasible. Suppose first that \( N \) is an infinite set. Let \( N' \subseteq N \) such that \( \lim_{n \in N'} x_k = x_{\infty} \) and \( \lim_{n \in N'} x_{k-1} = \hat{x} \). By Lemma 4.1,

\[
\eta_{k_n}(x_{k_n}) \to E[Q(x_{\infty}, \bar{\omega})] = b = \epsilon > 0.
\]

Then, with probability 1, there exists a \( K \) such that for all \( n \in N' \) such that \( k_n \geq K \), \( \eta_{k_n}(x_{k_n}) > 0 \), and therefore, by (2.7), \( \sigma_k \to 0 \). Also, by Lemma 4.1 and (4.1),

\[
\lim_{n \to \infty} \eta_{k_n}(x_{k_n-1}) = \lim_{n \to \infty} \eta_{k_n-1}(x_{k_n-1}) \leq \epsilon.
\]

Additionally, since \( \bar{x} \) is feasible to (P), and all cuts in the limit underestimate the value of \( E[Q(\bar{x}, \bar{\omega})] - b \), \( \lim_{k \to \infty} \eta_{k-1}(\bar{x}) = 0 \). By the definition of the set \( N \)

\[
P_{k_n}(x_{k_n}) - P_{k_n}(x_{k_n-1}) < \mu \left( P_{k_n-1}(x_{k_n}) - P_{k_n-1}(x_{k_n-1}) \right)
\]

\[
\leq \mu \left( P_{k_n-1}(\bar{x}) - P_{k_n-1}(x_{k_n-1}) \right).
\]
Thus, with probability 1
\[ \epsilon = \lim_{n \to \infty} P_k(x_k) \]
\[ < \mu \lim_{n \to \infty} P_{k-1}(\bar{x}) - \mu \lim_{n \to \infty} P_{k-1}(x_{k-1}) + \lim_{n \to \infty} P_k(x_{k-1}) \]
\[ \leq \mu \lim_{n \to \infty} P_{k-1}(\bar{x}) + (1 - \mu) \lim_{n \to \infty} P_{k-1}(x_{k-1}) \]
\[ \leq (1 - \mu) \epsilon < \epsilon, \]
which is a contradiction since \( 0 < \mu < 1 \) and \( \epsilon > 0 \). Therefore \( x_\infty \) must be feasible (wp1). Thus by (4.1) all \( x \in X \) are feasible to (P).

If \( N \) is a finite set and \( \bar{\sigma} > 0 \), then there exists a \( K \) such that for all \( k \geq K, x_k = x_K, \) and \( \bar{X} = \{x_K\}, \) and feasibility of \( x_K \) follows immediately as discussed.

The next lemma examines the limiting behavior of \( P_{k-1}(z_k) - P_{k-1}(x_{k-1}) \), the descent anticipated in moving from \( x_{k-1} \) to \( z_k \), when C2 holds.

**Lemma 4.3.** Let \( N \) be the set of indices at which the incumbent changes. If \( N \) is a finite set and \( \bar{\sigma} > 0 \) (i.e., if C2 holds), then with probability 1
\[ \lim_{k \to \infty} \{P_{k-1}(z_k) - P_{k-1}(x_{k-1})\} \to 0, \]
\[ \lim_{k \to \infty} c(z_k - x_{k-1}) = 0. \]

**Proof.** Let \( N \) be a finite set. Then, there exists a \( K \) such that for all \( k > K \) we have \( x_k = x_K \) and
\[ P_k(z_k) - P_k(x_K) \geq \mu(P_{k-1}(z_k) - P_{k-1}(x_K)). \] (4.4)

Let \( \{z_k\}_{j=1}^\infty \) be a subsequence of \( \{z_k\}_{k=1}^\infty \) such that \( z_{k_j} \to \bar{z} \). It has already been noted that \( \bar{\sigma} > 0 \) implies \( x_K \) and \( \bar{z} \) are feasible to (P) and thus
\[ \lim_{j \to \infty} P_{k_j}(z_{k_j}) = \lim_{j \to \infty} P_{k_j-1}(z_{k_j}) = \bar{\sigma} c \bar{z}. \] (4.5a)
\[ \lim_{j \to \infty} P_{k_j}(x_K) = \lim_{j \to \infty} P_{k_j-1}(x_K) = \bar{\sigma} c x_K. \] (4.5b)
Combining (4.4) and (4.5) yields

$$\lim_{j \to \infty} P_k(\mathbf{z}_{kj}) - P_k(x_K) \geq \mu \lim_{j \to \infty} P_{k_{j-1}}(\mathbf{z}_{kj}) - P_{k_{j-1}}(x_K)$$

$$= c \alpha \mathbf{z} - \alpha c x_K \geq \mu (\alpha \mathbf{z} - \alpha c x_K).$$

Since both sides of the inequality are nonpositive and $0 < \mu < 1$, $c \alpha \mathbf{z} = c x_K$ and thus the result.

Suppose that there exists an $x \in \mathbf{X}$ such that $E[Q(x, \tilde{\omega})] < b$. Since all limiting cuts underestimate the value of $E[Q(x, \tilde{\omega})] - b$, there exists a $K'$ such that for $k \geq K'$ the following problem is feasible.

$$(\text{MA}^k) \quad \min c x \quad \text{s.t.} \eta_k(x) \leq 0.$$ 

For $k \geq K'$, let $\hat{x}_{k+1}$ solve $(\text{MA}^k)$. For all $k \geq K$ we have

$$\sigma_k c \mathbf{z}_{k+1} \leq P_k(\mathbf{z}_{k+1}) \leq P_k(\hat{x}_{k+1}) = \sigma_k c \hat{x}_{k+1}.$$ 

Therefore,

$$c \mathbf{z}_{k+1} \leq c \hat{x}_{k+1}. \quad (4.6)$$

If $\mathbf{z}_{k+1}$ is feasible to $(\text{MA}^k)$, then we have $c \mathbf{z}_{k+1} = c \hat{x}_{k+1}$. The following can now be proved.

**Theorem 4.1.** Let $x_\infty \in \overline{\mathbf{X}}$ and suppose there exists an $x \in \mathbf{X}$ such that $E[Q(x, \tilde{\omega})] < b$. If $N$ is in an infinite set then $x_\infty$ is an optimal solution of $(\mathbf{P})$ with probability 1. If $N$ is a finite set and $\tilde{\sigma} > 0$ then the unique limit point, $x_K$, is an optimal solution of $(\mathbf{P})$ with probability 1.

**Proof.** Let $u^*$ be the optimal objective value of problem $(\mathbf{P})$. By the cut dropping scheme in Section 2 we have $|J^k| \leq 2n_1 + 3$ for all $k = 1, 2, \ldots$. Let $k_j$ denote the $j$th index in set $J^k$. Let $\hat{K} \subseteq \{1, 2, \ldots\}$ be such
that \( \{u_k\}_{k \in \hat{K}} \to u_\infty, \) \(|J^k| = p\) for all \( k \in \hat{K} \) and for \( j = 1, \ldots, p \)
\[
\{ \alpha_k^j \}_{k \in \hat{K}} \to \bar{\alpha}^j, \quad \{ \beta_k^j \}_{k \in \hat{K}} \to \bar{\beta}^j.
\]

For \( k \in \hat{K} \)
\[
\eta_k(x) = \max_{j = 1, \ldots, p} \{ 0, \alpha_k^j + \beta_k^j x - b \}.
\]

Therefore,
\[
\{ \eta_k(x) \}_{k \in \hat{K}} \to \bar{\eta}(x) = \max_{j = 1, \ldots, p} \{ 0, \bar{\alpha}^j + \bar{\beta}^j x - b \}.
\]

Note that for \( j = 1, \ldots, p, \) \( \bar{\alpha}^j + \bar{\beta}^j x \leq E[Q(x, \bar{\omega})] \). Therefore, \( \bar{\eta}(x) \leq 0 \) is a relaxation of the constraint \( E[Q(x, \bar{\omega})] \leq b \) and therefore \( u_\infty \leq u^* \). Since the above can be constructed for any accumulation point of \( \{ u_k \}_{k = 1}^\infty \) we have
\[
\lim_{k \to \infty} u_k \leq u^*. \tag{4.7}
\]

Since any \( x_\infty \in \overline{X} \) is feasible to \((P)\) with probability 1 under the given hypothesis, we must have for all \( x_\infty \in \overline{X} \)
\[
u^* \leq c x_\infty. \tag{4.8}
\]

Suppose \( N \) is an infinite set. Let \( \{ x_{k_n} \}_{k_n \in N} \to x_\infty \in \overline{X} \) and \( \{ u_{k_n-1} \}_{k_n \in N} \to u_\infty \). Then, by (4.6) and the definition of \( N, c x_{k_n} \leq u_{k_n-1} \) for all \( k_n \in N, \)
\( k_n \geq K' \). Therefore,
\[
c x_\infty \leq u_\infty. \tag{4.9}
\]

Thus (4.7), (4.8), and (4.9) imply that \( c x_\infty = u^* \) almost surely and \( x_\infty \) solves \((P)\) with probability 1.

Suppose \( N \) is a finite set. Let \( \{ z_{k_j} \}_{j=1}^\infty \) be a subsequence of \( \{ z_k \}_{k=1}^\infty \) and \( \{ u_{k_j-1} \}_{j=1}^\infty \) be a subsequence of \( \{ u_k \}_{k=1}^\infty \) such that \( z_{k_j} \to z_\infty \) and \( u_{k_j-1} \to u_\infty \). Let \( x_K \) be the unique limit point of \( \{ x_k \}_{k=1}^\infty \). By (4.6) we have \( c z_{k_j} \leq u_{k_j-1} \)
for all \( k_j \geq K' \). Therefore, combining (4.6) and Lemma, 4.3 we have

\[
c x_k = c z_\infty \leq u_\infty \tag{4.10}
\]

almost surely. Therefore, (4.7), (4.8), and (4.10) yield \( x_k = u_\infty \) with probability 1 and thus \( x_k \) is almost surely optimal to (P).

As a consequence of the previous analysis, the following corollary applies regardless of whether \( \sigma_k \to 0 \) or not.

**Corollary 5.** Any accumulation point of \( \{ z_k \}_{k=1}^\infty \) is optimal to problem (P) if it is feasible to (P).

Feasibility of the accumulation points of the sequence of candidates, \( \{ z_k \}_{k=1}^\infty \), when \( N \) is a finite set and \( \sigma_k \to 0 \) is guaranteed if cuts are not dropped. This can be easily shown by noting that the set of constraints in \( M_k \) that have a positive dual multiplier at a candidate solution can be bounded by \( n_1 + 1 \) for all \( k \). Thus, for any accumulation point, \( z_\infty \) of \( \{ z_k \}_{k=1}^\infty \), a subsequence of iterations, \( \hat{K} \subseteq \{1, 2, \ldots, \} \), exists such that for \( k \in \hat{K} \) we have \( z_k \to z_\infty \), only \( p \) constraints in \( M_k \) have positive dual multipliers, and \( \{ \eta_k \}_{k=\hat{K}} \) epi-converges to a function \( \bar{\eta} \) where

\[
\bar{\eta}(x) \leq \max \{0, E[Q(x, \tilde{\omega})] - b\}, \quad \text{and}
\]

\[
\bar{\eta}(z_\infty) = \max \{0, E[Q(z_k, \tilde{\omega})] - b\}. \tag{4.11}
\]

It follows from [12] that \( z_\infty \) solves

\[
\min \{ \bar{\eta}(x) : x \in X \},
\]

which is a relaxation of the problem

\[
\min \{ \max \{0, E[Q(x, \tilde{\omega})] - b\} : x \in X \}. \]

Therefore, since there exists a feasible solution to (P) by assumption, (4.11) implies that \( z_\infty \) is feasible to (P).

Since one goal of this research is to design an algorithm that is not computationally burdensome, the computational tests presented in Section 6 all include cut dropping.
5. TERMINATION CRITERIA

Step 7 of the Penalty Stochastic Decomposition Algorithm given in Section 3 can be altered to check if certain heuristic termination criteria are satisfied.

**Step 7.** Solve \((M^k)\) to obtain \(z_k+1\) and \(\eta^*_k(z_k+1)\) and the dual variables \(\lambda^j_k, j \in J^k\). If termination criteria are met, then stop. Otherwise return to Step 1.

As noted in [7], the stability of the set \(V_k\) must be considered before deciding to terminate an SD algorithm. Thus termination is considered if

\[
V_k = V_{k-K'},
\]

where \(K'\) is a sufficiently larger integer. This is easily verified by comparing the cardinality of the sets since \(V_{k-K} \subseteq V_k\) for all \(k\).

As in SD, the objective function at the incumbent solution should exhibit a degree of stability also. One way to detect that the objective has stabilized is to keep a summary, \(v_k\), of its value at the current incumbent solution over the first \(k\) iterations. This summary could be an exponentially smoothed average, in which case \(v_k = \lambda c \cdot x_k + (1 - \lambda) v_{k-1}\), \(\lambda \epsilon (0,1)\), and \(v_0 = c \cdot x_0\). Then for \(\epsilon_1 > 0\) and small, satisfaction of

\[
\frac{|c \cdot x_k - v_{k-1}|}{|c \cdot x_k|} \leq \epsilon_1
\]

at iteration \(k\) would indicate that the objective has stabilized.

It is also important not to stop while one has some indication that constraint (2.1) is violated. Let \(g(x_k) = \alpha_k x_k^t + \beta_k x_k\). Since any point \(x \in X\) such that \(E[Q(x, \omega)] = b\) is feasible to \((P)\), the sample variance associated with \(g(x_k)\) can be used to test the hypothesis \(H_0: E[Q(x_k, \omega)] \geq b + \epsilon_2\) versus \(H_1: E[Q(x_k, \omega)] < b + \epsilon_2\), where \(\epsilon_2\) is small compared with \(b\) and is necessary if the algorithm is to terminate when the recourse constraint (2.1) is satisfied as an equality. Only an approximation of the theoretical variance of \(g(x_k)\) is available. Let \(s^2\) denote this approximation. That is,

\[
s^2 = \frac{1}{k-1} \sum_{t=1}^{\tilde{k}} \left[ \pi^t_k (\omega_t - T x_k) - g(x_k) \right]^2
\]

\[
+ \frac{1}{k-1} \sum_{t=\tilde{k}+1}^k \left[ \pi^t_k (\omega_t - T x_k) - g(x_k) \right]^2.
\]
Then assuming a normal approximation, the most powerful test available is one that rejects $H_0$ at a given level of significance $\alpha$, assuming large $k$, only if

$$\sqrt{k} \left( g(x_k) - (b + \varepsilon_2) \right) s < -t_{\alpha, k-1}. \quad (5.1)$$

Here for a given level of significance $\alpha$, $t_{\alpha, k-1}$ is the critical value for the $t$ distribution based on $k - 1$ degrees of freedom and a one-sided critical region (see [13] or [14]). Thus one could consider terminating at iteration $k$ if (5.1) is satisfied.

Meeting all of the criteria just described is suggested.

6. COMPUTATIONAL TESTING

The Penalty Stochastic Decomposition Algorithm presented in Section 3 has been applied to variations of three problems found in the literature after they were put in the form of (P). The problem name followed by the letters $rc$, which stand for recourse constrained, will designate this form.

Problem CEP1rc is a capacity expansion problem for a manufacturing plant with the objective of minimizing the cost of new capacity while keeping the expected cost of labor plus tooling within a fixed budget. This problem (due to S. Sen) is described in the Appendix.

SCAGR7 is a large dairy farm expansion planning model, [16]. The original model was used to maintain an optimal livestock mix as well as projected growth rates and profits by determining the acreage of crops to plant, the amount of grain and hay to purchase, and the disposition of newborn cattle.

SCRS8 is a dynamic energy model for the transition from fossil fuels to renewable energy resources. This problem was derived from a model of U.S. options for a minimum cost transition from oil and gas to synthetic fuels while meeting future energy demands, [17]. The future depends on estimates of the remaining quantities of domestic oil and gas resources and the technical and environmental feasibility of new methods for synthetic fuel production.

The last two problems are two-stage stochastic versions of deterministic multistage problems due to [18]. The origins and descriptions of these two appear in [19].

To verify the method, for each problem the right-hand side, $b$, was set equal to the value of the recourse function at the optimal solution of the original two-stage formulation given at the beginning of this paper. This
solution was determined using the L-shaped method, [20], which was coded by Mousa Mitwasi at the University of Arizona. This solution is also an optimal solution of each recourse-constrained problem for that value of $b$. The terminating incumbent solution was checked for optimality and feasibility to $\mathcal{P}$.

The method has been implemented on a Vax 8650 at the University of Arizona. The FORTRAN code utilizes the XMP subroutines [21] for solving the subproblems ($S^k$) and the penalty master program ($M^k$). All of the criteria for termination discussed in Section 5 were implemented. Additionally, a minimum of 1500 iterations was performed. A high iteration count was necessary in an attempt to prevent premature termination at an infeasible point. Storage restrictions required termination within 3000 iterations.

The following parameters were used in all computer runs.

\[
\begin{align*}
\sigma_0 &= 10.0 \text{ (initial penalty parameter)}, \\
\mu &= 0.25 \text{ (new incumbent parameter)}, \\
\tau &= 20 \text{ (cut reestimation parameter)}, \\
\epsilon_1 &= 0.0005 \text{ (termination tolerance)}, \\
\lambda &= 0.25 \text{ (exponential smoothing parameter)}, \\
\alpha &= 0.05 \text{ (level of significance)}. 
\end{align*}
\]

The value of $\epsilon_2$, which is used in the hypothesis test for each problem, was as follows:

\[
\begin{align*}
\epsilon_2 &= 0.04b \text{ for CEP2}, \\
\epsilon_2 &= 0.02b \text{ for SCRS8rc}, \\
\epsilon_2 &= |0.0005b| \text{ for SCAGR7rc}.
\end{align*}
\]

The parameter $\epsilon_2$ will unavoidably be problem dependent since scaling the constraint set, for example, would imply that $\epsilon_2$ would have to be scaled similarly for the same hypothesis test level. Ideally, the value of $\epsilon_2$ should be as small as possible.

For each problem, the program was run using 10 different starting seeds for the random number generator. Table 1 summarizes the averages.
TABLE 1
SUMMARY OF COMPUTATION

<table>
<thead>
<tr>
<th></th>
<th>CEP1rc</th>
<th>SCRS8rc</th>
<th>SCAGR7rc</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg. # iter.</td>
<td>1970.0</td>
<td>1501</td>
<td>1501.4</td>
</tr>
<tr>
<td></td>
<td>(595.0)</td>
<td>(0)</td>
<td>(0.5)</td>
</tr>
<tr>
<td>avg. # cuts</td>
<td>16.0</td>
<td>46.3</td>
<td>31.6</td>
</tr>
<tr>
<td></td>
<td>(1.0)</td>
<td>(0.3)</td>
<td>(0.3)</td>
</tr>
<tr>
<td>max. dev. from opt. (^1)</td>
<td>0.072</td>
<td>0</td>
<td>0.001</td>
</tr>
<tr>
<td>avg. dev. from opt. (^2)</td>
<td>0.01</td>
<td>0</td>
<td>0.0004</td>
</tr>
<tr>
<td></td>
<td>(0.02)</td>
<td>(0)</td>
<td>(0.0004)</td>
</tr>
<tr>
<td>max. dev. from feas. (^3)</td>
<td>0.12</td>
<td>0</td>
<td>0.000024</td>
</tr>
<tr>
<td>avg. dev. from feas. (^4)</td>
<td>0.001</td>
<td>0</td>
<td>0.00001</td>
</tr>
<tr>
<td></td>
<td>(0.004)</td>
<td>(0)</td>
<td>(0.00001)</td>
</tr>
</tbody>
</table>

\(^1\)The maximum deviation from the optimal objective value of the objective value at the terminal incumbent.
\(^2\)The average deviation from the optimal objective value of the objective value at the terminal incumbent.
\(^3\)The maximum deviation from feasibility of the recourse constraint at the terminal incumbent.
\(^4\)The average deviation from feasibility of the recourse constraint at the terminal incumbent.

over the 10 runs for the listed quantities: the average number of iterations (avg. # iter.), the average number of cuts added to the master programs (avg. # cuts), the maximum deviation from feasibility of the recourse function at the terminal incumbent (max. dev. from feas.), the average deviation from feasibility of the recourse function at the terminal incumbent (avg. dev. from feas.), the maximum deviation from optimality of the objective function at the terminal incumbent (max. dev. from opt.), and the average deviation from optimality of the objective function at the terminal incumbent (avg. dev. from opt.). The standard deviations appear in parentheses.

Nine of the runs of problem CEP1rc produced optimal solutions at termination. Of these nine, two did not terminate before the maximum number of iterations, 3000. Only a single run produced a nonoptimal solution. This particular terminal incumbent deviated from feasibility by 1.2% and had an objective value 7.2% lower than the optimal solution.
Estimates of the recourse function play a critical role in determining the next candidate. For example, low estimates of the recourse function increase the size of the region that the algorithm detects as feasible. Thus an infeasible point that is detected as feasible when the estimates are low may be preferred over an optimal point. This is apparently the case with the single run that produced a nonoptimal terminal incumbent. A subsequent rerun of this particular run with a smaller value for the level of significance $\alpha$ performed no better.

Each run of the algorithm with problem SCRS8rc terminated at the optimal solution. This was expected since only one dual basis was produced for the given data set.

The method applied to problem SCAGR7rc terminated at incumbent points that exhibited an average deviation from optimality of only 0.04%. Only four of the terminal incumbent solutions were actually feasible but the average deviation from feasibility was only 0.001%.

Only the output of problem CEP1rc indicated that if these runs were not terminated the incumbent might have changed only finitely often and $\sigma_k \rightarrow 0$. Only one run of this problem terminated at a nonoptimal solution. The fact that all of the other runs terminated at an optimal solution is an indication that this unresolved case may not be an obstacle.

7. CONCLUSIONS AND EXTENSIONS

In this paper a stochastic decomposition algorithm for solving a stochastic programming problem in which a recourse function is present in the constraint set was introduced. This algorithm uses an exact penalty function in the master program. The convergence results, when cuts are dropped, were obtained for all cases except the situation in which the incumbent changes only finitely often and the penalty parameter converges to zero.

Results of the computational testing indicate that the algorithm shows promise and that the unresolved case may not be problematic. Further testing of the algorithm with various values of $b$ and sensitivity analysis is warranted.

It is likely that the case in which the incumbent does not change but $\sigma_k \rightarrow 0$ can not be resolved with the current sequence of functions, $\{P_k(x)\}_{k=1}^\infty$, since they are not constructed in a manner that ensures their epi-convergence to $P(x) = \max\{0, E[Q(x, \bar{\omega}) - b]\}$. The addition of a quadratic term in the master program may aid in resolving this case.

The algorithm can be adapted to accommodate the case in which the matrix $T$ is also a function of the random variable $\bar{\omega}, T(\bar{\omega})$. Thus, in generating an observation of $\bar{\omega}$, one simultaneously obtains observations of
the matrix \( T \) as well as the right-hand side of the subproblem. Except for this addition, the resulting algorithm is identical to that presented in this paper.

Variance reduction techniques such as importance sampling (as in [22]) or stratified sampling (see [23]) may be combined with the algorithm and may prove especially useful since detection of feasibility is of primary importance in terminating the algorithm.

The algorithm presented here and the SD or QSD algorithms can be combined to solve stochastic programming problems that include several constraints of type (1.1) as well as a recourse problem in the objective. Consider the following formulation:

\[
\begin{align*}
\min & \; c x + E[Q_0(x, \tilde{\omega}_0)] \\
\text{s.t.} & \; x \in X \\
& \; E[Q_i(x, \tilde{\omega}_i)] \leq b_i, \; i = 1, \ldots, L,
\end{align*}
\]

where \( Q_i(x, \omega_i), \; i = 0, \ldots, L \) is defined as

\[
Q_i(x, \omega_i) = \min q_i y \\
\text{s.t.} \; W_i y + T_i x = \omega_i, \quad y \geq 0.
\]

Conceptionally, there is no problem with combining the QSD algorithm presented in [8] and the one here. In this case, penalty weights on the exact penalty functions for each of the constraints, rather than the original objective, would provide the individual control needed to enforce feasibility. Since the subproblems are independent of one another, the resulting algorithm would benefit from parallel processors that could handle the numerous subproblems and the increased storage requirements.

APPENDIX. DESCRIPTION OF PROBLEM CEP1rc

Suppose a manufacturing plant produces several different part types on any one of several machines. A particular part can be produced on one or more of the machines at a fixed rate (parts per hour) at a fixed cost (labor plus tooling cost). Each machine has a fixed number of hours currently available per week and hours of new capacity of each machine type can be acquired at a fixed cost per hour.

The maximum time that any machine can be used in a week is
restricted to a fixed number of hours and each machine requires a fixed number of hours of maintenance for every hour of operation. The total scheduled maintenance for all machines is also restricted each week.

Each week the demand for each part type must be met. If the total demand exceeds the total capacity, the excess parts are obtained from a subcontractor at a premium price that is assumed to be much greater than the manufacturer’s own cost of producing the part on any one of the machines. The weekly demands are treated as independent, identically distributed random variables with known distribution. Let

\[ m \equiv \text{number of different part types}, \]
\[ n \equiv \text{number of different machines}, \]
\[ T \equiv \text{total maintenance hours for all machines}, \]
\[ b \equiv \text{the total budget for labor plus tooling on all machines}. \]

Then, for \( j = 1, \ldots, n \) and \( i = 1, \ldots, m \) let

\[ x_j \equiv \text{the number of hours of new capacity of type } j, \]
\[ c_j \equiv \text{the cost of new capacity of type } j, \]
\[ a_{ij} \equiv \text{rate (parts per hours) of producing part } i \text{ on machine } j, \]
\[ q_{ij} \equiv \text{cost (labor plus tooling) of producing part } i \text{ on machine } j, \]
\[ h_j \equiv \text{the number of hours machine } j \text{ is currently available each week}, \]
\[ U_j \equiv \text{total usage of machine } j \text{ allowed each week}, \]
\[ z_j \equiv \text{hours of operation of machine } j \text{ each week}, \]
\[ t_j \equiv \text{hours of maintenance for every hours of operation of machine } j, \]
\[ \omega_i \equiv \text{weekly demand for part } i, \]
\[ s_i \equiv \text{number excess parts of type } i \text{ obtained from subcontractor}, \]
\[ p_i \equiv \text{price of subcontracting for part } i, p_i \geq q_{ij}. \]

With the objective of minimizing the cost of new capacity while keeping the expected cost of weekly labor plus tooling below a fixed budget \( b \), the recourse-constrained stochastic linear program can be formulated as follows:

\[
\text{CEP1rc}
\]

\[
\min \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad -x_j + z_j \leq h_j, \quad j = 1, \ldots, n, \\
\sum_{j=1}^{n} t_j z_j \leq T
\]
\[ E[Q(z, \omega)] \leq b \]
\[ 0 \leq z_j \leq U_j \]
\[ 0 \leq x_j, \quad j = 1, \ldots, n, \]

where

\[ Q(z, \omega) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} q_{ij} y_{ij} + \sum_{i=1}^{n} p_i s_i \]

s.t. \[ \sum_{j=1}^{n} a_{ij} y_{ij} + s_i \geq \omega_i, \quad i = 1, \ldots, m \]
\[ \sum_{i=1}^{m} y_{ij} \leq z_j, \quad j = 1, \ldots, n \quad y_{ij} \geq 0, \quad s_i \geq 0. \]

CEP1rc Data

- \( n = 4 \)
- \( m = 3 \)
- \( T = 100 \)
- \( b = 338494.0 \)
- \( c_j = (2.5, 3.75, 5.0, 3.0) \)
- \( t_j = (0.08, 0.04, 0.02, 0.01) \)
- \( h_j = (500, 500, 500, 500) \)
- \( u_j = (2000, 2000, 3000, 3000) \)
- \( p_i = (400, 400, 400) \)

\[
[a_{ij}] = \begin{bmatrix}
0.6 & 0.6 & 0.9 & 0.8 \\
0.1 & 0.9 & 0.6 & 0.8 \\
0.05 & 0.2 & 0.5 & 0.8
\end{bmatrix}
\]

\[
[g_{ij}] = \begin{bmatrix}
2.6 & 3.4 & 3.4 & 2.5 \\
1.5 & 2.3 & 2.0 & 3.6 \\
4.0 & 3.8 & 3.5 & 3.2
\end{bmatrix}
\]
\( \omega_i, \ i = 1, 2, 3 \) are independent, discrete random variables with all outcomes equally likely,

\[ \omega_i \in \{0, 600, 1200, 1800, 2400, 3000\}. \]

REFERENCES


