A REGULARIZED ALGORITHM FOR SOLVING TWO-STAGE STOCHASTIC LINEAR PROGRAMMING PROBLEMS : A WATER RESOURCES EXAMPLE

Diana S. Yakowitz
USDA-Agricultural Research Service
Southwest Watershed Research Center
2000 E. Allen Rd. Tucson, AZ 85719
USA

Stochastic linear programming problems are linear programming problems in which one or more data elements are random variables. Two-stage stochastic linear programming problems are problems in which a first stage decision is made before the random variables are observed. A second stage, or recourse decision, which varies with these observations, compensates for any deficiencies that result from the earlier decision. In this paper, an algorithm for solving stochastic linear programming problems with recourse is presented. Referred to as Regularized Stochastic Decomposition, the algorithm is a major improvement over the original Stochastic Decomposition algorithm. It was developed to be computationally more efficient than the original by introducing a quadratic proximal term in the master program objective function and altering the manner in which the recourse function approximations are updated. The addition of the quadratic regularizing term in the master program objective function justifies a cut dropping scheme that allows one to bound the size of the master programs. The algorithm is applied to a water resources problem assuming continuous random variables.

INTRODUCTION

The original Stochastic Decomposition (SD) algorithm (Higle and Sen, 1991) for solving two-stage stochastic programming problems combines the use of sampling procedures to estimate the objective function with a decomposition method similar to the L-Shaped method of Van Slyke and Wets (1969). A major handicap of the SD algorithm is that the size of the master programs solved in each iteration of the method increases progressively. In order to alleviate this problem, Yakowitz (1991) introduced a quadratic proximal (regularizing) term to the otherwise linear objective function of the SD master program making possible a cut dropping scheme similar to that given in Mifflin (1977) and Kiwiel (1985). Convergence results have been strengthened by including such a quadratic term in mathematical programming

K. W. Hipel (ed.),
Stochastic and Statistical Methods in Hydrology and Environmental Engineering, Vol. 2, 271–284.

algorithms which are otherwise linear (Kiwiel, 1985; Ruszczynski, 1986,1987).

In addition to eliminating unnecessary constraints from the master program, a new updating mechanism for the retained past cuts that is statistically motivated and takes advantage of information obtained in each iteration of the algorithm is proposed. An adaptive method to determine when to make additional re-evaluation of certain cuts is also presented (Yakowitz, 1991,1994). The resultant algorithm is referred to as Regularized Stochastic Decomposition (RSD).

The algorithm described in this paper is applicable to a wide class of water resource problems such as water reuse planning problems. An example which requires determining the capacity of a reservoir and canal system that is to supply water for irrigation of crop land is considered. Yeh (1985) and Reznicek and Cheng (1991) review modeling efforts in reservoir management and operations including stochastic efforts with two-stage programs. Yeh points out that these efforts usually assume discrete random variables or discrete approximations leading to large size deterministic equivalent programs which are expensive to solve since the consequences of the 1st stage action requires that the second stage be solved for every possible realization of the random variables. SD algorithms can be applied even with continuous random variables.

This presentation is organized as follows: A preliminary discussion of the general problem formulation and a summary of the RSD algorithm steps are given in the first section. This is followed by a detailed description of some of the algorithm operations. Convergence results and algorithmic stopping rules are summarized in the third section. A water resources example is presented in the forthe section and the RSD algorithm is applied to it assuming continuously distributed random variables.

PRELIMINARIES AND ALGORITHM SUMMARY

The two-stage stochastic linear program with recourse can be stated as follows:

(P)

$$\begin{aligned} &\text{Min} \;\; f(x) = cx + E_{\tilde{\omega}}[Q(x,\tilde{\omega})] \\ &\text{s.t.} \;\; x \in \mathbf{X} \subseteq \mathbf{R}^{n_1} \end{aligned}$$

where

$$\begin{aligned} Q(x,\omega) = \;\; &\text{Min} \;\; qy \\ &\text{s.t.} \;\; Wy = \omega - Tx \\ &\qquad y \geq 0, \end{aligned}$$

The set $\mathrm{X} = \{x | Ax \leq b\}$ is a compact convex polyhedral set. A is a known $m_1 \times n_1$ matrix, and c, q and b are known vectors in $\mathbf{R}^{n_1}$, $\mathbf{R}^{n_2}$, and $\mathbf{R}^{m_1}$, respectively. The random vector, $\tilde{\omega}$, is defined on a probability space $(\Omega, \mathcal{A}, P)$, with associated distribution function $F_{\tilde{\omega}}$. Ω is a compact set, and $E_{\tilde{\omega}}[\cdot]$ represents the mathematical expectation with respect to $\tilde{\omega}$. The specified matrix W is $m_2 \times n_2$, and T, which can be stochastic or deterministic, is $m_2 \times n_1$.

The RSD algorithm produces a sequence of points $\{x_k\}_{k=1}^{\infty}$, referred to as "incumbent solutions", a sequence of directions, $\{d_k\}_{k=1}^{\infty}$, and a sequence of "candidate" solutions, $\{z_k\}_{k=2}^{\infty}$. These sequences are related by $z_{k+1} = x_k + d_k$ for $k = 1, 2, \ldots$. In iteration k, the direction d_k is determined through the solution of a quadratic master program. Beginning with an initial incumbent solution, a candidate is accepted as the next incumbent if its estimated objective value is sufficiently lower than that of the current incumbent solution.

Given a candidate solution z_k and an observation ω_k of $\tilde{\omega}$, a subproblem that is the dual to the recourse problem given above is solved:

(S^k)

$$\begin{aligned} Q(z_k, \omega_k) = &\text{Max} \ \ \pi(\omega_k - T z_k) \\ &\text{s.t. } \pi \in \Pi = \{\pi \ : \ \pi W \leq q\}. \end{aligned}$$

where π is an m_2 dimensional row vector. We assume that Π is a non-empty compact convex polyhedral set. Therefore, $|Q(x, \omega)| < \infty$ for all $(x, \omega) \in \mathbf{X} \times \mathbf{\Omega}$, and this implies that (P) has the complete recourse property (Wets, 1982).

The quadratic master program (M^k) of the RSD algorithm is given by:

(M^k)

$$\text{Min}\{ \ \frac{1}{2}\|d\|^2 + \nu_k(d) \ : \ x_k + d \in \mathrm{X}\},$$

where

$$\nu_k(d) = \max_{j \in J^k}\{f_k^j(x_k + d)\}. \tag{1}$$

The function $\nu_k(d)$ is used to approximate the objective function of (P) at $x_k + d$ using linear approximations (cuts) of f denoted by $f_k^j(\cdot)$. The superscript indicates that the cut is associated with the j^{th} candidate solution, z_j, while the subscript indicates the iteration in which the cut was last updated (the current iteration in this case). The size of the index set $J^k \subseteq \{1, 2, \ldots, k\}$ is constrained and thereby acts to limit the size of the master program. The set J^k is redefined in each iteration of the algorithm. The precise definition of this set and that of the cuts are given in the next section. The solution of (M^k) is denoted by d_k and the $k + 1^{st}$ candidate solution is given by $z_{k+1} = x_k + d_k$.

A summary of the steps of the algorithm is now presented. Suggested initialization and other details of the steps (including how the cuts are formed) will follow in the next section.

Summary of the RSD algorithm

Step 0. $k \leftarrow 0$. Initialize: ω_0, x_0, d_0, $f_0^0(x_0) = cx_0 + Q(x_0, \omega_0)$, $z_1 = x_0 + d_0$, $J^0 = \emptyset$.

Step 1. $k \leftarrow k + 1$. Generate random vector.
Randomly generate an observation, ω_k, according to its distribution.

Step 2. Solve subproblem (S^k) at z_k and save it's solution.

Step 3. Determine the cut at z_k. Evaluate $f_k^k(z_k)$ using the current and past solutions to (S_k).

Step 4. Update past cuts and re-evaluate the cut associated with the current incumbent solution if necessary.

Step 5. Determine the k^{th} incumbent solution.
If the estimate of f at z_k is significantly lower than the estimate of f at x_{k-1} then $x_k \leftarrow z_k$. Otherwise, $x_k \leftarrow x_{k-1}$.

Step 6. Drop cuts and solve master program (M^k).
Determine $J^k \subseteq J^{k-1} \cup k$ and solve (M^k) to obtain d_k and $\nu_k(d_k)$. Set $z_{k+1} = x_k + d_k$.

Step 7. Determine if stopping criteria are met. If so, stop. Otherwise, return to Step 1.

ALGORITHMIC DETAILS

Initialization of the algorithm, Step 0, can be accomplished in many ways. One obvious choice would be to let $\omega_0 = E[\tilde{\omega}]$, and choose $x_0 \in$ argmin $\{cx + Q(x, \omega_0) : x \in \mathrm{X}\}$. $d_0 \leftarrow 0$, $z_1 \leftarrow x_0$. In Step 1 of each iteration, ω_k is generated independently of previous samples.

Function approximation

RSD utilizes estimates of the objective function $f(x)$. These estimates are produced as in SD as follows:

Let V denote the set of all extreme point solutions of the recourse problem, and let $\mathrm{V}_k \subseteq \mathrm{V}$ be the set of extreme points of (S^k) identified in the first k iterations of the algorithm (Step 2). For $t = 1, 2, \ldots, k$, let π_t^k satisfy

$$\pi_t^k \in \operatorname{argmax}[\pi(\omega_t - Tz_k) \ : \ \pi \in \mathrm{V}_k]. \tag{2a}$$

At Step 3 of the k^{th} iteration, an estimate of a support of f at x_k is given by

$$f_k^k(x) = cx + \frac{1}{k}\sum_{t=1}^{k} \pi_t^k(\omega_t - Tx). \tag{2b}$$

As iterations progress, the cuts will be updated and the subscript on f incremented to indicate that the updating has been performed. Representing this function in terms of the variable d we obtain:

$$f_k^k(x_k + d) = \alpha_k^k + (c + \beta_k^k)(x_k + d),$$

where

$$\alpha_k^k = \frac{1}{k}\sum_{t=1}^{k} \pi_t^k \omega_t, \tag{3a}$$

and

$$\beta_k^k = -\frac{1}{k}\sum_{t=1}^{k} \pi_t^k T. \tag{3b}$$

Updating

With each iteration, previously generated constraints lack information gained from subsequent sampling of the random variable $\tilde{\omega}$. Note that for any x and ω_t

$$Q(x, \omega_t) \geq \pi(\omega_t - Tx) \quad \forall \pi \in \mathbf{V}.$$

In particular, since $\mathbf{V_k} \subseteq \mathbf{V}$, for all k

$$Q(x, \omega_t) \geq \pi_k^k(\omega_t - Tx).$$

Yakowitz (1991,1994) proposes the following update of the coefficients of past cuts (those with superscripts $j < k$) in iteration k at Step 4:

$$\alpha_k^j = \frac{k-1}{k}\alpha_{k-1}^j + \frac{1}{k}\pi_k^k\omega_k$$
$$\beta_k^j = \frac{k-1}{k}\beta_{k-1}^j - \frac{1}{k}\pi_k^k T,$$

where α_k^k and β_k^k are defined as in (3). With this updating scheme the piecewise linear approximation $\nu_k(d)$ defines a statistically valid lower bound of the objective function in (P).

Since a particular solution may remain as the incumbent over many iterations and the functions $\{f_k^j\}_{j=1}^k$ represent a statistically motivated, piecewise linear approximation of the convex function f, the cut associated with the incumbent solution should be re-estimated whenever

$$f_k^k(x_{k-1}) > f_k^{\gamma_{k-1}}(x_{k-1}),$$

where γ_{k-1} denotes the iteration that the $k-1^{st}$ incumbent solution, x_{k-1}, was accepted. If the number of iterations between re-estimations of the cut at the incumbent is bounded then the function estimates at the incumbent solution converge to the actual value (with probability 1). The re-estimation can be accomplished as in (2) replacing the superscripts with γ_{k-1} and using the current set of subproblem vectors, V_k, and x_{k-1} in place of z_k.

Determining the next incumbent solution

In Step 5, the k^{th} incumbent solution is determined. Since $z_k = x_{k-1} + d_{k-1}$, the quantity $\nu_{k-1}(d_{k-1}) - f_{k-1}^{\gamma_{k-1}}(x_{k-1})$ represents the amount of descent anticipated in the $k-1^{st}$ iteration in moving from x_{k-1} to z_k, while $f_k^k(z_k) - f_k^{\gamma_{k-1}}(x_{k-1})$ is the descent the function estimates actually exhibit in the k^{th} iteration. Thus, z_k becomes the k^{th} incumbent, x_k, if

$$f_k^k(z_k) - f_k^{\gamma_{k-1}}(x_{k-1}) < \mu\big(\nu_{k-1}(d_{k-1}) - f_{k-1}^{\gamma_{k-1}}(x_{k-1})\big), \tag{4}$$

where μ is a fixed parameter such that $0 < \mu < 1$. Satisfaction of (4) implies that a sufficient fraction of the anticipated objective value reduction is attained. In such cases $x_k = z_k$. Otherwise, the incumbent does not change and $x_k = x_{k-1}$.

Dropping master program constraints

At each iteration of the original SD algorithm, one additional linear inequality is added to the master program. After a large number of iterations, the number of constraints can become burdensome. Many of the constraints do not play a role in defining the optimal solution to the master program.

When the incumbent changes, descent is indicated and elimination of constraints that do not define the piecewise linear approximation near the new incumbent is desired. In iterations that the incumbent does not change, one should retain the cut that is associated with the current incumbent solution as well as those constraints that define the piecewise linear approximation near the current candidate.

Let $J^{k-1} \subseteq \{1, 2, \ldots, k-1\}$ be the set of indices that define $\nu_{k-1}(d)$ in (1) in iteration $k-1$. Let n_1 be the dimension of x, the first stage decision variable. By Caratheodory's Theorem (see Bazaraa and Shetty, 1979), at most $n_1 + 1$ constraints are needed to define a solution to (M^{k-1}). The active constraints are identified as those with non-zero Lagrange multipliers. Most quadratic programming subroutines automatically provide the associated Lagrange multipliers with not more than n_1+1 of them non-zero (Kiwiel, 1985). Since the constraints associated with X, the feasible region of the first stage variable, are fixed, we are concerned only with the multipliers λ^j_{k-1}, $j \in J^{k-1}$, that are associated with the constraints indicated by (1) at d_{k-1}. Let $\hat{J}^{k-1} = \{j \in J^{k-1} \; : \; \lambda^j_{k-1} > 0\}$. The set J^k is defined as follows in Step 6.

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

Thus, a finite master program size of at most $n_1 + 3$ constraints can be maintained in any iteration.

Since the termination criteria are related to the convergence results of the algorithm, discussion of these will follow those results in the next section.

CONVERGENCE RESULTS AND TERMINATION CRITERIA

Convergence results

In this section a way to easily identify a subsequence of the incumbent solutions, $\{x_k\}$, $k = 1, 2, \ldots$ whose accumulation points are almost surely optimal solutions of P is described. Lemmas, theorems and proofs that establish that there exists such a sequence appear in Yakowitz (1991, 1994). The following theorem (stated without proof here) is a result of the limiting behavior of the sequence of incumbent cuts (Theorem 2, Higle and Sen, 1991) and a consequence of the quadratic term in the master program (M^k), which bounds the anticipated descent in moving from x_k to z_k, in each iteration (Lemma 2, Yakowitz 1991,1994).

Theorem 1. *(Lemma 3 and Theorem 1, Yakowitz, 1994)*
Let $\{d_k\}_{k=1}^{\infty}$ be the sequence of master program solutions. Then there exists a subset of indices, K', such that $\{d_k\}_{k\in K'} \to 0$ almost surely and if K is any index set such that

$$\{x_k\}_{k\in K} \to x_\infty, \quad \{d_k\}_{k\in K} \to 0,$$

then x_∞ is an optimal solution of (P) almost surely.

A description of how to identify such a subsequence is now given. Since with probability 1 there exists an infinite subset K' such that $\{d_k\}_{k\in K'} \to 0$, any accumulation point of $\{x_k\}_{k\in K'}$ is an optimal solution of (P) with probability 1, by Theorem 1. When the incumbent changes only finitely often, the unique accumulation point of the incumbent solutions is an optimal solution with probability 1.

When the incumbent changes infinitely often a method to identify a subsequence of the incumbent solutions that accumulates at optimal solutions is needed.

Let δ_0 be sufficiently large and define constants $\mu_2 \leq \mu_1 < 1$ (used to prevent δ from decreasing too rapidly). Then, δ_k is defined as follows:

$$\delta_k = \begin{cases} \mu_1 \delta_{k-1}, & \text{if } \|d_k\| < \mu_2 \delta_{k-1}; \\ \min[\, \delta_{k-1}, \|d_k\|], & \text{otherwise.} \end{cases}$$

The monotonic sequence $\{\delta_k\}_{k=1}^{\infty}$ converges to zero, with probability 1, since there exists, by Theorem 1, a subsequence of indices, K', such that $\{d_k\}_{k \in K'} \to 0$. The set of indices K' can be defined as follows:

$$K' = \{k : \|d_k\| \leq \delta_{k-1}\}. \tag{5}$$

Clearly, K' is an infinite set since either $\delta_k = \mu_1 \delta_{k-1}$ infinitely often or $\delta_k = \|d_k\|$ infinitely often. Since $\delta_k \to 0$, we then have $\{d_k\}_{k \in K'} \to 0$ and obtain the following.

Corollary 2. *Let $\{x_k\}_{k=1}^{\infty}$ be the sequence of incumbent solutions identified by the algorithm, and let K' be the index set defined in (5), then every accumulation point of $\{x_k\}_{k \in K'}$ is an optimal solution of P, with probability 1.*

Termination criteria

Since the cuts used to constrain $\mathbf{M_k}$ are derived from set $\mathbf{V_k}$, termination of the algorithm at Step 7 should be considered only after a sufficiently large number of iterations have passed in which no new vector in V has been found.

A statistical summary, η_k, of the incumbent objective values, $f_k^{\gamma_k}(x_k)$, can be used to monitor their progress. In particular, for those iterations corresponding to the subsequence defined in (5) (i.e. $k \in K'$) one can test whether the following is satisfied.

$$\frac{|f_k^{\gamma_k}(x_k) - \eta_k|}{|f_k^{\gamma_k}(x_k)|} < \epsilon \tag{6}$$

We use an exponentially smoothed average defined as

$$\eta_k = \begin{cases} \lambda f_k^{\gamma_k}(x_k) + (1-\lambda)\eta_{k-1}, & \text{if } k \in K'; \\ \bar{\eta}_{k-1}, & \text{otherwise} \end{cases}$$

where $\lambda \in (0,1)$, and η_0 is appropriately chosen.

Termination should not be considered unless $\|d_k\|$ is small. One can use the sequence $\{\|d_k\|\}_{k \in K'}$ to compute a statistic, ρ_k, similar to that above. The algorithm may be terminated if for $k \in K'$ we have

$$\rho_k < c.$$

Meeting all of the criteria described above is suggested in order to avoid premature termination.

A WATER RESOURCES EXAMPLE

While the convergence results given above require that the probability space be compact, most real world examples are not so obliging. The following example illustrates the use of the algorithm for random variables which violate the compactness assumption. In particular gamma distributions are assumed for the annual precipitation and inflow to a reservoir.

Problem description

To illustrate the RSD algorithm the following hypothetical situation is considered. A small dam is to be constructed across a river in Arizona providing the facility for the storage of water delivered by means of a canal for agricultural use, or for downstream use by direct releases of the water. Water for agriculture can also be purchased from an outside source, such as the Central Arizona Project (CAP), or pumped from existing groundwater wells (the amount pumped restricted by recharge estimates). The first stage variables are the capacity R of the reservoir and the capacity C of the canal. The second stage variables include the amount of water x_a released from the reservoir for agriculture, the amount of water x_d released downstream, the amount of water x_g pumped from groundwater, and x_e, the amount of water obtained from the external source (i.e. CAP) for agricultural use. These are determined after the stochastic rainfall, inflow and downstream demand are realized. The first stage objective is to minimize the maintenance cost of the dam and canal system given by $c_r R + c_c C$. We assume that the initial cost of building the dam and canal system is to be amortized over an extended period and is reflected in the maintenance costs. The second stage objective is to minimize the cost of purchasing or pumping water, $c_e x_e + c_g x_g$, minus the crop yield revenues, which are directly proportional to the water used for irrigation, $r(x_a + x_e + x_g)$.

First stage constraints impose upper and lower bounds on the reservoir and canal capacities, which are given by R_{max}, C_{max} and R_{min}, C_{min} respectively. The initial storage level of the water in the reservoir at the end of stage 1, $\tilde{s}_1$, is assumed to be a fraction (random) of the capacity R. The following second stage constraints are imposed:

$$C - x_a - x_e - x_g \geq 0 \tag{7a}$$

$$\frac{1}{5}R \leq s_2 \leq R \tag{7b}$$

$$R - \frac{1}{3}(x_a + x_d) \geq 0 \tag{7c}$$

$$s_2 + x_a + x_d = \tilde{Y} + \tilde{s}_1 \tag{7d}$$
$$x_d \geq \tilde{M} \tag{7e}$$
$$w_{min} - \tilde{P} \leq x_a + x_e + x_g \leq w_{max} - \tilde{P} \tag{7f}$$
$$x_g \leq \frac{1}{10}(x_e + x_a + \tilde{P}) \tag{7g}$$

Constraint (7a) ensures that the capacity of the canal is adequate to handle the flow. The constraints of (7b) guarantee that the storage s_2 at the end of the second stage does not exceed the reservoir capacity or drop below $\frac{1}{5}$ of the reservoir capacity. Constraint (7c) requires that the reservoir capacity be at least $\frac{1}{3}$ of the total water released for agriculture and downstream use. This constraint is a surrogate for a more complicated constraint system that would insure that peak demand could be met. If $\tilde{Y}$ represents the stochastic inflow to the reservoir, (7d) is the water balance equation, that is, the change in storage, $s_2 - \tilde{s}_1$ must equal the inflow minus the outflow from the reservoir. The next constraint, (7e), requires that the water released downstream must satisfy a minimum stochastic demand, $\tilde{M}$. The set of constraints (7f) bounds the amount of water for agriculture, including the stochastic precipitation, $\tilde{P}$, between a minimum, w_{min}, and maximum, w_{max}, crop requirement. The last inequality, (7g), restricts the water pumped from groundwater for agricultural to amounts less than the recharge of the aquifer, which is estimated in this example to be 10% of the precipitation and water from other sources applied to the fields.

Thus, if we let $\tilde{\omega} = (\tilde{Y},\ \tilde{M},\ \tilde{P},\ \tilde{s}_1)$, the stochastic program takes the form:

$$\begin{aligned} &\text{Min} \quad c_R R + c_C C + E[Q(R, C; \tilde{\omega})] \\ &\text{s.t.} \\ &\qquad R_{min} \leq R \leq R_{max} \\ &\qquad C_{min} \leq C \leq C_{max} \end{aligned}$$

where

$$\begin{aligned} Q(R, C; \omega) = \quad &\text{Min} \quad c_e x_e + c_g x_g - r(x_a + x_e + x_g) \\ &\text{s.t.} \quad (7\text{a}, \text{b}, \text{c}, \text{d}, \text{e}, \text{f}, \text{g}) \\ &\qquad x_a,\ x_d,\ x_e,\ x_g,\ s_2 \geq 0. \end{aligned}$$

The RSD algorithm will be used to solve the above problem. An alternative stochastic programming formulation to the one above is one that bounds the expected value of the recourse problem, which would therefore appear in the first stage constraint set instead of the objective function of (P). Minimizing the 1st stage costs of the reservoir and canal while ensuring that second stage decisions are such that a certain profit level is achieved is an example of this type of formulation. An algorithm for this variation of the standard two-stage stochastic program with recourse is proposed in Yakowitz (1992). It involves the introduction of an exact penalty term into the objective function of the master program.

Cost coefficients, distributions of the random variables and other parameter values used for this example problem and the algorithm implementation appear in the appendix to this paper.

Results

The RSD algorithm was applied to the problem above using 5 independently generated streams of the continuous random variables (see Appendix for distribution information).

Figure 1 is a plot of the incumbent objective value estimates from one of the five replications showing iterations 300 through 1519 when the algorithm terminated. Notice that the estimates lack monotonicity from iteration to iteration but exhibit an increasing trend. From iteration 1000 till termination the change in objective function was less than 0.02% of the termination value.

Figure 2 is a plot of the norm of d in moving from one incumbent solution to the next for the same replication as above (iteration 600 through 1519). In iterations when the incumbent does not change, the norm of d is 0. Note that between iteration 600 and 1400 the incumbent changed every iteration.

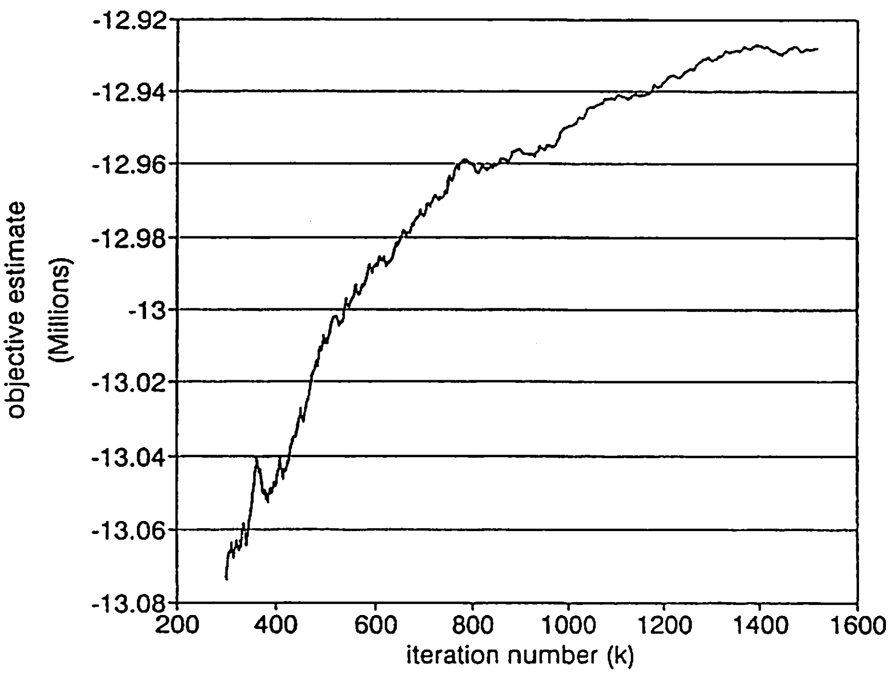

Figure 1. Plot of incumbent objective estimates: iterations 300 - 1519.

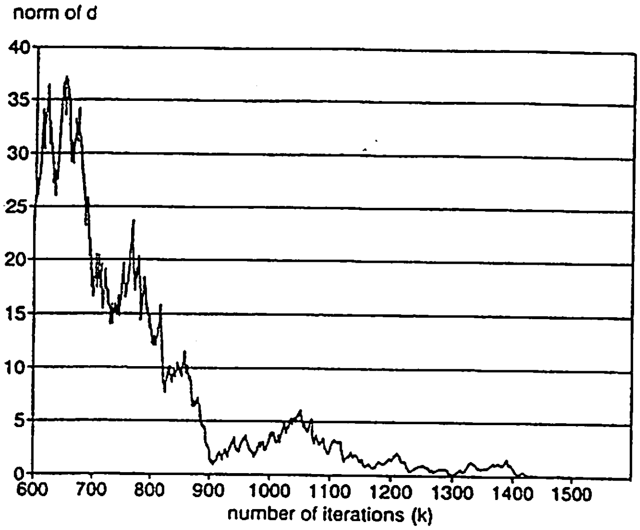

Figure 2. Plot of $\|d_k\|$: iterations 600 - 1519.

	Average	(std. dev.)
# of iterations	1793	(165)
avg. # of cuts	3.39	(0.01)
# re-estimations	381	(36)
cardinality of V_T	7.6	(0.6)
relative error in F_T	0.0035	(0.0004)

TABLE 1. Summary of RSD replications at termination

Table 1 summarizes the results with averages over the 5 replications of the indicated quantities: average number of iterations, average number of cuts in the master programs, average number of times the incumbent cut was re- estimated and the average cardinallity of the the set V_k at termination ($k = T$). With continuous random variables the optimal value of the objective function is unknown. Therefore reported in Table. 1 is the relative error in the objective value estimates at the terminal incumbent. We use the average deviation from the sample mean of the terminal incumbent objective value, based on an independent sample of size 3000, as a fraction of the sample mean objective value. Standard deviations associated with the replications appear in parentheses. All five replications satisfied the stopping conditions at termination.

The apparent convergence of the incumbent sequence and the stability of the objective function estimates as indicated by the strict stopping rules for all five trials suggests that the algorithm performs quite well for this example. The RSD algorithm appears to be a computationally viable alternative to other stochastic programming methods that require discrete random variables, or the discretization of continuous ones, before those methods can be used. Other computational tests of the algorithm appear in Yakowitz (1994).

APPENDIX

Parameter values for the example problem are not based on an actual example. We have assumed that a total of 22,500 acres (9105.5 hectares (ha)) are to be supplied with adequate water for cotton. Some information such as water requirements and the cost of CAP water was obtained from Wilson (1992). Initial solution was set at $R = R_{max}$, $C = C_{max}$.

$$\begin{aligned}
c_R &= \$60.00\ /acre\ foot\ (\$486.40/ha \cdot m) \\
c_C &= \$25.00\ /acre\ foot\ (\$202.68/ha \cdot m) \\
R_{min} &= 10,000\ acre\ feet\ (1,233.5\ ha \cdot m) \\
R_{max} &= 250,000\ acre\ feet\ (30,837.5\ ha \cdot m) \\
C_{min} &= 90,000\ acre\ feet\ (11,101.5\ ha \cdot m) \\
C_{max} &= 135,000\ acre\ feet\ (16,652.25\ ha \cdot m) \\
r &= \$175.00\ /acre\ foot\ (\$1,418.73\ /ha \cdot m) \\
c_e &= \$52.00\ /acre\ foot\ (\$421.56\ /ha \cdot m) \\
c_g &= \$35.00\ /acre\ foot\ (\$283.75\ /ha \cdot m) \\
w_{min} &= 90,000\ acre\ feet\ (11,101.5\ ha \cdot m) \\
w_{max} &= 135,000\ acre\ feet\ (16,652.25\ ha \cdot m)
\end{aligned}$$

The annual inflow, $\tilde{Y}$, is assumed to be gamma(a,b) distributed with a=180,000 acre feet ($22,203.0\ ha \cdot m$), b=1/2. The annual downstream demand is $\tilde{M} = \frac{1}{3}\tilde{Y}$.Annual precipitation, $\tilde{P}$ is assumed for this example to be gamma(a) distributed with parameter a=22,000 acre feet ($2,713.7\ ha \cdot m$).The initial storage of the reservoir, s_1, is assumed to be uniformly distributed between $\frac{1}{5}R$ and R.

The following parameters were used in all replications: $\mu = 0.25$ (the new incumbent parameter); $\epsilon = 0.005$ (termination tolerance); $\lambda = 0.25$ (exponential smoothing parameter).

The convergence analysis of RSD requires that only those cuts with indices in J^k need be retained. However, the current implementation also retains those cuts which are tight at the current incumbent solution in iterations when the incumbent does not change. That is, at most n_1+1 cuts such that $\alpha_k^j + \beta_k^j x_k = \alpha_k^{\gamma_k} + \beta_k^{\gamma_k} x_k$, $j \in J^{k-1}$ are also retained. Thus, we retain at most $2n_1 + 3$ cuts in each iteration.

An indication of the stability of the objective function before termination was determined by satisfying (6) with $\eta_k = \lambda f_k^{\gamma_k}(x_k) + (1-\lambda)\eta_{k-1}$. We also require that in the iterations that the incumbent changes, we have $\rho_k < \epsilon$ where $\rho_k = \lambda \|d_k\| + (1-\lambda)\rho_{k-1}$. If the incumbent has not changed, we require $\|d_k\| < \epsilon$.

Termination of the algorithm was considered only if the cardinality of the set V_k remained the same for at least 50 iterations.

Linear programs were solved using the XMP algorithm of Marsten (1987). Quadratic programs were solved with ZQPCVX of Powell (1986). Gamma distributions were generated using Best's rejection algorithm XG in Devroye (1986).

REFERENCES

Bazaraa, M.S., Shetty, C.M. (1979) Nonlinear Programming, Theory and Algorithms, John Wiley & Sons, Inc., New York.

Devroye, L. (1986) Non Uniform Random Variate Generation, Springer-Verlag, New York.

Higle, J.L. and Sen, S. (1991) "Stochastic decomposition: an algorithm for two stage linear programs with recourse," Math. of Oper. Res. 16, 650-669.

Kiwiel, K.C. (1985) Methods of Descent for Nondifferentiable Optimization, Lecture notes in Mathematics no. 1133, Spring-Verlag, Berlin.

Marsten, R.E. (1987) XMP Technical Reference Manual, Department of Management Information Systems, College of Business and Public Administration, University of Arizona, Tucson, Az.

Mifflin, R. (1977) "An algorithm for constrained optimization with semismooth functions," Math. of Oper. Res. 2, 191-207.

Powell, M.J.D. (1986) ZQPCVX (MODIFIED) Quadratic Programming Algorithm, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England.

Reznicek, K. and Cheng, T.C.E. (1991) "Stochastic modelling of reservoir operations," European Journal of Operational Research 50, 235-248.

Ruszczynski, A. (1986) "A regularized decomposition method for minimizing a sum of polyhedral functions," Math. Prog. 35, 309-333.

Ruszczynski, A. (1987) "A linearization method for nonsmooth stochastic programming problems," Math. Oper. Res. 12, 32-49.

Van Slyke, R., and Wets, R.J-B (1969) "L-shaped linear programs with applications to optimal control and stochastic programming," SIAM J. Appl. Math. 17, 638-663.

Wets, R.J-B. (1982) "Stochastic programming: solution techniques and approximation schemes," in Mathematical Programming: The State of the Art, A. Bachem, M. Groetschel, B. Korte, eds., Springer- Verlag, Berlin, 506-603.

Wilson, P.N. (1992) "An economic assissment of central Arizona project agriculture," report, Department of Agricultural and Resource Economics, College of Agriculture, University of Arizona, Tucson, AZ.

Yakowitz, D.S. (1991) Two-Stage Stochastic Linear Programming: Stochastic Decomposition Approaches, Ph.D. Dissertation, University of Arizona.

Yakowitz, D.S. (1992) "An exact penalty algorithm for recourse-constrained stochastic linear programs," Appl. Math. and Comp. 49, pp. 39-62.

Yakowitz, D.S. (1994) "Regularlized stochastic decomposition with finite master program size for two-stage stochastic linear programs with recourse," Computational Optimization and Appl. 3, 59-81.

Yeh, W. W-G. (1985) "Reservoir management and operations models: a state-of-the-art review," Water Resources Research 21(12), 1797-1818.