Computational Optimization and Applications, 3, (1994), 59–81
© 1994 Kluwer Academic Publishers. Manufactured in The Netherlands.

A Regularized Stochastic Decomposition Algorithm for Two-Stage Stochastic Linear Programs*

DIANA S. YAKOWITZ
USDA-Agricultural Research Service, Southwest Watershed Research Center, 2000 E. Allen Rd., Tucson, AZ 85719

Received January 27, 1993; Revised July 30, 1993

Abstract. In this paper a regularized stochastic decomposition algorithm with master programs of finite size is described for solving two-stage stochastic linear programming problems with recourse. In a deterministic setting cut dropping schemes in decomposition based algorithms have been used routinely. However, when only estimates of the objective function are available such schemes can only be properly justified if convergence results are not sacrificed. It is shown that almost surely every accumulation point in an identified subsequence of iterates produced by the algorithm, which includes a cut dropping scheme, is an optimal solution. The results are obtained by including a quadratic proximal term in the master program. In addition to the cut dropping scheme, other enhancements to the existing methodology are described. These include (i) a new updating rule for the retained cuts and (ii) an adaptive rule to determine when additional reestimation of the cut associated with the current solution is needed. The algorithm is tested on problems from the literature assuming both descrete and continuous random variables.

Keywords: stochastic programming, decomposition, cutting plane methods

Introduction

The algorithm presented here is a stochastic decomposition (SD) based method [5] for solving two-stage stochastic linear programs. The purpose of this work is to address a major handicap of a SD algorithm: the progressively increasing size of the master program. In deterministic cutting plane algorithms, theoretical justification for elimination of old cuts has been addressed in the pioneering work of Eaves and Zangwill [3] and by others [7, 10]. In these works, dropping cuts while preserving the convergence results relies on the monotonic descent of the objective function. In stochastic programming, the L-shaped method of Van Slyke and Wets [16], for example, exhibits monotonic descent of the objective function since the cuts are produced by solving the subproblem for every possible realization of the discrete or discretized random variables. Consequently,

*A majority of this work is part of the author's Ph.D. dissertation prepared at the University of Arizona in 1990.

implementations based on this method [2, 4] that include cut dropping schemes have been able to do so without jeopardizing the convergence to optimal solutions.

Monotonic descent of the objective function is not exhibited, however, by the SD algorithm. In SD, cuts are estimated based on the solutions of only one subproblem, solved at the point of concern, as well as information available from past subproblem solutions. Consequently, the objective function changes nonmonotonically with each iteration. In a deterministic setting, simple examples can be constructed in which the elimination of active constraints causes the algorithm to oscillate between nonoptimal solutions. With SD, due to the stochastic nature of the cuts, even cuts that are inactive at a given point in one iteration may become active at the same point in a subsequent iteration (a situation which does not arise in a deterministic setting). Therefore, justification for dropping cuts from the SD master program is not straightforward.

In order to alleviate the problem of the ever-increasing master program size, while preserving the convergence results of the original SD algorithm, a quadratic proximal (regularizing) term is introduced to the objective function. The introduction of this term makes possible a cut dropping scheme similar to that given in [7] and [10]. Introduction of this term was motivated by its use in several deterministic algorithms, especially those dealing with nondifferentiable optimization problems, that exhibited stronger convergence results than could otherwise be obtained when such a quadratic term is absent (see [7, 10, 11, 13, 14, 15]).

In addition to eliminating unnecessary constraints from the master program, a new updating mechanism for the retained past cuts is proposed. It is statistically motivated and takes advantage of information obtained in each iteration of the algorithm. An adaptive method to determine when to make additional reestimations of certain cuts is also presented. The resultant algorithm will be referred to as regularized stochastic decomposition (RSD).

This presentation is organized as follows: Components of the RSD algorithm are described in Section 1 and 2 concluding with a formal statement of the RSD algorithm. Convergence results for RSD are presented in Section 3 and termination criteria are discussed in Section 4. Section 5 contains computational results for several test problems.

1. The RSD Method

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

(P)

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

Where

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

The set $\mathbf{X} = \{x | Ax \leq b\}$ is a convex polyhedral set assumed to be compact. 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 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 an observation ω_k of $\tilde{\omega}$, a subproblem $(\mathbf{S}^k)$ defined as follows is solved:

$(\mathbf{S}^k)$

$$\begin{aligned} Q(z_k, \omega_k) = \text{Min } & qy \\ \text{s.t. } & Wy = \omega_k - Tz_k \\ & y \geq 0. \end{aligned}$$

The dual to the above problem is

$(\mathbf{DS}^k)$

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

where π is an m_2-dimensional row vector. We assume that Π is a nonempty compact convex polyhedral set. Therefore, $|Q(x, \omega)| < \infty$ for all $(x, \omega) \in \mathbf{X} \times \Omega$, which implies that $(\mathbf{P})$ has the complete recourse property [17].

Note that the constraint set Π is independent of the realization ω_k. This fact is exploited by the method by which the estimates of the objective function $f(x)$ are constructed by SD algorithms. How estimates are produced follows.

Let $\mathbf{V}$ denote the set of all extreme point solutions of the subproblem dual, and let $\mathbf{V}^k \subseteq \mathbf{V}$ be the set of extreme points of $(\mathbf{DS}^k)$ identified in the first k iterations of the algorithm. For $t = 1, 2, \ldots, k$, let π_t^k satisfy

$$\pi_t^k \in \text{argmax}[\pi(\omega_t - Tz_k) : \pi \in \mathbf{V}^k]. \tag{1a}$$

In the kth iteration, an estimate of a support of f at z_k is given by

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

The superscript of f is used to indicate the iteration at which the cut was first derived. As iterations progress, these cuts will be updated and the subscript on f will be used to indicate the iteration in which the cut was last updated. Thus, $f_t^j(x)$ is the cut that was first derived in the jth iteration and was last updated in the tth iteration. Representing (1b) in terms of the variable d, we have

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

where

$$\alpha_k^k = \frac{1}{k}\sum_{t=1}^{k} \pi_t^k \omega_t \tag{2a}$$

and

$$\beta_k^k = -\frac{1}{k}\sum_{t=1}^{k} \pi_t^k T. \tag{2b}$$

With each iteration, previously generated constraints lack information, gained from subsequent sampling of the random variable $\tilde{\omega}$. If we write $Q^k(x)$ for the sample mean of $Q(x,\tilde{\omega})$ using the information obtained so far, we have

$$\begin{aligned} Q^k(x) &= \frac{1}{k}\sum_{t=1}^{k} Q(x, \omega_t) \\ &= \frac{k-1}{k} Q^{k-1}(x) + \frac{1}{k} Q(x, \omega_k) \\ &\geq \frac{k-1}{k}\left[\alpha_{k-1}^t + \beta_{k-1}^t x\right] + \frac{1}{k} L(x, \omega_k), \qquad t = 1, \ldots, k-1, \end{aligned}$$

where $L(x, \omega_k)$ is a function that is at most linear in x and satisfies $L(x,\omega_k) \leq Q(x,\omega_k)$. One choice, if $Q(x,\omega) \geq 0$ for all $(x,\omega) \in \mathbf{X} \times \boldsymbol{\Omega}$, is to let $L(x,\omega_k) = 0$. This update is used in the original SD algorithm. Note that for any x and ω_t even if Q is unrestricted in sign we have

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

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

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

Therefore, if we let $L(x, \omega_k) = \pi_k^k(\omega_t - Tx)$, the following more flexible update of the coefficients of past cuts (those with superscripts $j = 1, 2, \ldots, k-1$) is proposed: $f_k^j(x) = \alpha_k^j + (c + \beta_k^j)x$ where

$$\alpha_k^j = \frac{k-1}{k}\alpha_{k-1}^j + \frac{1}{k}\pi_k^k\omega_k \tag{3a}$$

$$\beta_k^j = \frac{k-1}{k}\beta_{k-1}^j - \frac{1}{k}\pi_k^k T, \tag{3b}$$

and α_k^k and β_k^k defined as in (2).

Let $\|\cdot\|$ indicate the Euclidean norm. The quadratic master program $(\mathbf{M}^k)$ of the RSD algorithm is given by:

$$(\mathbf{M}^k) \quad \text{Min}\left\{\frac{1}{2}\|d\|^2 + \nu_k(d) : x_k + d \in \mathbf{X}\right\},$$

where

$$\nu_k(d) = \max_{j \in \mathbf{J}^k}\{f_k^j(x_k + d)\}. \tag{4}$$

The function $\nu_k(d)$ approximates the objective function at $x_k + d$. The size of the index set $\mathbf{J}^k \subseteq \{1, 2, \ldots, k\}$ will be constrained and thereby acts to limit the size of the master program. $\mathbf{J}^k$ is redefined in each iteration. The cut dropping scheme and the precise definition of this set is described in Section 2. The solution of $(\mathbf{M}^k)$ is denoted by d_k and the $k+1$st candidate solution is given by $z_{k+1} = x_k + d_k$.

Before describing how an incumbent solution in SD algorithms is selected we note that a particular solution may remain as the incumbent solution over many iterations. In order to guarantee that the function estimate at an incumbent solution converges to the actual value (with probability 1), Higle and Sen [5] suggest that the cut associated with a current incumbent solution be reestimated at each iteration. They do, however, point out that reestimation of the incumbent cut at each iteration is not necessary if the number of iterations between reestimations is bounded. Let γ_{k-1} denote the iteration in which the $k-1$st incumbent solution, x_{k-1}, was accepted. Then in iteration k, $f_k^{\gamma_{k-1}}(x_{k-1})$ represents the current estimate of the objective in $(\mathbf{P})$ at x_{k-1}. Since the functions $\{f_k^j\}_{j=1}^k$ represent a statistically motivated, piecewise linear, approximation of the convex function f, one way to adaptively determine when reestimation should be performed is to check if

$$f_k^k(x_{k-1}) > f_k^{\gamma_{k-1}}(x_{k-1}). \tag{5}$$

This inequality can occur since π_t^t for $t = \gamma_{k-1} + 1$ to $t = k$ may not be the element from $\mathbf{V}^k$ that maximizes $\pi(\omega_t - Tx_{\gamma_{k-1}})$. Thus, the estimates of the cut at the incumbent are "looser" than should be. If (5) is satisfied, the value of $f_k^{\gamma_{k-1}}(x_{k-1})$ is too low and can be raised. The reestimation can be accomplished as in (1) using the current set of subproblem dual vectors, $\mathbf{V}^k$.

Checking (5) above does not guarantee that the iterations between reestimation will be bounded. Therefore, to complete the reestimation instructions, let τ be a positive integer and in iteration k let $\overline{\tau}_{k-1}$ represent the iteration at which the cut associated with the $k-1$st incumbent solution was last evaluated. Then, reestimate $f_k^{\gamma_{k-1}}(x)$ whenever one of the following is satisfied:

$$f_k^k(x_{k-1}) - f_k^{\gamma_{k-1}}(x_{k-1}) > 0, \quad \text{or} \tag{6a}$$

$$k - \overline{\tau}_{k-1} = \tau. \tag{6b}$$

If one of the conditions in (6) is satisfied then $\overline{\tau}_{k-1}$ is reset to k. Analogous to (1), reestimation requires that for $t = 1, 2, \ldots, k$,

$$\overline{\pi}_t^{\gamma_{k-1}} \in \text{ argmax}[\pi(\omega_t - Tx_{k-1}) : \pi \in \mathbf{V}^k], \tag{7a}$$

is determined and then,

$$f_k^{\gamma_{k-1}}(x) = cx + \frac{1}{k}\sum_{t=1}^{k} \overline{\pi}_t^{\gamma_{k-1}}(\omega_t - Tx) = \alpha_k^{\gamma_{k-1}} + (c + \beta_k^{\gamma_{k-1}})x. \tag{7b}$$

Suppose $(\mathbf{M}^{k-1})$ has just been solved and now in iteration k the kth incumbent is to be determined. If $z_k = x_{k-1} + d_{k-1}$, then $\nu_{k-1}(d_{k-1}) - f_{k-1}^{\gamma_{k-1}}(x_{k-1})$ represents the amount of descent anticipated 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 (after updating) in the kth iteration. Then, z_k becomes the kth incumbent, x_k, if

$$f_k^k(z_k) - f_k^{\gamma_{k-1}}(x_{k-1}) < \mu(\nu_{k-1}(d_{k-1}) - f_{k-1}^{\gamma_{k-1}}(x_{k-1})), \tag{8}$$

where μ is a fixed parameter such that $0 < \mu < 1$. Satisfaction of (8) 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 $x_k = x_{k-1}$.

2. Finite master program size and the RSD algorithm

At each iteration of the 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. In a deterministic setting, cut dropping schemes such as those of Eaves and Zangwill [3], Miffin [10] and Kiwiel [7] have been proposed. These schemes use a line search procedure that guarantees descent in the objective function, thus justifying the elimination of cuts. The next function approximation is obtained by defining a piecewise linear function whose pieces include the most recent cut together with a select subset of previously obtained pieces. Unlike these methods, satisfaction of (8), the condition for accepting a candidate as the new incumbent solution, is only a statistically based indication of descent.

When the incumbent changes, descent is indicated by (8) and one would like to retain information in a neighborhood of the new incumbent and eliminate those constraints that do not define the piecewise linear approximation near the new incumbent. In iterations in which 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 $\mathbf{J}^{k-1} \subseteq \{1, 2, \ldots, k-1\}$ be the set of indices in (4) that define $\nu_{k-1}(d)$ in iteration $k-1$. Let n_1 be the dimension of x, the first-stage decision variable. By Caratheodory's Theorem (see [1], for example), at most $n_1 + 1$ constraints are needed to define a solution to $(\mathbf{M}^{k-1})$. The active constraints are identified as those with nonzero Lagrange multipliers at the solution. Most quadratic programming subroutines automatically provided the associated Lagrange multipliers with not more than $n_1 + 1$ of them nonzero [7]. Since the constraints associated with $\mathbf{X}$, the feasible region of the first-stage variable, are fixed, we are concerned only with the multipliers λ_{k-1}^j, $J \in \mathbf{J}^{k-1}$, that are associated with the constraints indicated by (4) at d_{k-1}. Let $\hat{\mathbf{J}}^{k-1} = \{j \in \mathbf{J}^{k-1} : \lambda_{k-1}^j > 0\}$. The set $\mathbf{J}^k$ is defined as follows.

$$\mathbf{J}^k = \hat{\mathbf{J}}^{k-1} \cup \{\gamma_k, k\}. \tag{9}$$

Thus, $\mathbf{J}^k$ is the set of the constraint indices from $\mathbf{J}^{k-1}$ that are active (have positive multipliers) at the current candidate solution, plus the cut associated with the current incumbent and the new cut associated with the current candidate.

With this cut dropping scheme, RSD will maintain a finite master program size with at most $n_1 + 3$ constraints at any iteration.

The preceding developments are summarized as follows.

Algorithm: Regularized stochastic decomposition (RSD)

Step 0. (Initialize)
$k \leftarrow 0$, $\omega_0 \leftarrow E[\tilde{\omega}]$, $x_0 \in \operatorname{argmin}\{cx + Q(x, \omega_0) : x \in \mathbf{X}\}$,
$z_1 \leftarrow x_0$, $d_0 \leftarrow 0$, $\gamma_0 \leftarrow 0$, $f_0^0(x_0) = cx_0 + Q(X_0, \omega_0)$, $\mathbf{V}^0 = \emptyset$, $\bar{\tau}_0 \leftarrow 0$.
$0 < \mu < 1, M$ large, and τ is given. $\mathbf{J}^0 = \emptyset$.
Step 1. (Generate random vector)
$k \leftarrow k + 1$. Randomly generate an observation, ω_k, according to the distribution $F_{\tilde{\omega}}$ (ω_t, $t = 1, 2, \ldots, k$ are generated independently).
Step 2. (Solve subproblem)
Solve $(\mathbf{S}^k)$ and obtain the dual vector $\pi(z_k, \omega_k)$. $\mathbf{V}^k \leftarrow \mathbf{V}^{k-1} \cup \{\pi(z_k, \omega_k)\}$.
Step 3. (Update master program)
Evaluate $f_k^k(z_k)$ according to (1) and for $j \in \mathbf{J}^{k-1}$ determine the updated cut coefficients according to (3).
Step 4. (reestimate incumbent cut)

Check reestimation conditions (6). If satisfied, determine $f_k^{\gamma_{k-1}}(x_{k-1})$ according to (7), $\bar{\tau}_{k-1} \leftarrow k$.
Step 5. (Check new incumbent condition and drop cuts)
a) If (8) is satisfied then $x_k \leftarrow z_k$, $\gamma_k \leftarrow k. \bar{\tau}_k \leftarrow k$. Otherwise
b) $x_k, \gamma_k, \bar{\tau}_k$ are unchanged.
In either case, determine $\mathbf{J}^k$ according to (9).
Step 6. (Solve master program)
Solve $(\mathbf{M}^k)$ to obtain d_k and $\nu_k(d_k)$ and the dual variables λ_k^j, $j \in \mathbf{J}^k$. Set $z_{k+1} = x_k + d_k$, return to step 1.

3. Convergence analysis

In this section we will show that an easily identifiable subsequence of the incumbent solutions, $\{x_k\}$, accumulates at optimal solutions.

We begin with a lemma from [5] that examines the limiting behavior of the sequence of incumbent cuts generated by the SD or RSD algorithms. This lemma applies to the sequences generated by RSD since they do not depend on the type of master program, but rather on the characteristics that SD algorithms have in common (i.e., the manner in which cuts are generated, the rule by which the incumbent solutions are selected, the reestimation of the incumbent cut).

LEMMA 1. ([5], *Theorem* 2). *Let* $\{x_{k_n}\}_{n=1}^{\infty}$ *be an infinite subsequence of* $\{x_k\}_{k=1}^{\infty}$. *If* $\{x_{k_n}\} \to x_\infty$, *then with probability* 1,

$$f_{k_n}^{\gamma_{k_n}}(x_{k_n}) \to cx_\infty + E[Q(x_\infty, \tilde{\omega})] = f(x_\infty).$$

Furthermore, every accumulation point of $\{\alpha_{k_n}^{\gamma_{k_n}}, c + \beta_{k_n}^{\gamma_{k_n}}\}_{n=1}^{\infty}$ *defines a support of* $f(x)$ *at* x_∞, *with probability* 1.

Next, a lemma that establishes bounds for $(\nu_k(d_k) - f_k^{\gamma_k}(x_k))$, the anticipated descent in going from x_k to z_k, in each iteration is presented. The bound obtained is a consequence of the quadratic term in the master program $(\mathbf{M}^k)$. This bound plays an important role in establishing that subsequences of $\{d_k\}_{k=1}^{\infty}$ that converge to the zero vector, $\mathbf{0}$, exist.

LEMMA 2. *For each* k,

$$-\|c + \beta_k^{\gamma_k}\|\|d_k\| \leq (\nu_k(d_k) - f_k^{\gamma_k}(x_k)) \leq -\|d_k\|^2 \leq 0.$$

Proof. Let superscript T indicate "transpose." The necessary conditions for a solution to $(\mathbf{M}^k)$ imply that for each $k, (d_k, \nu_k)$ satisfy

$$A(x_k + d_k) \leq b,$$

$$\nu_k(d_k) \geq f_k^j(x_k + d_k) = f_k^j(x_k) + (c + \beta_k^j)d_k,\ j \in \mathbf{J}^k, \tag{10}$$

$$d_k^T + \sum_{j \in \mathbf{J}^k} \lambda_k^j (c + \beta_k^j) + \sigma_k A = \mathbf{0}^T, \tag{11}$$

$$\sum_{j \in \mathbf{J}^k} \lambda_k^j = 1,\ \lambda_k^j \geq 0,\ \sigma_k \geq \mathbf{0} \tag{12}$$

where σ_k is a m_1-dimensional row vector. Furthermore, the following complementary slackness conditions must also be met.

$$\lambda_k^j (f_k^j(x_k) + (c + \beta_k^j)d_k - \nu_k(d_k)) = 0,\ j \in \mathbf{J}^k, \tag{13}$$

$$\sigma_k(Ad_k + Ax_k - b) = 0. \tag{14}$$

Summing (13) over all $j \in \mathbf{J}^k$, adding and subtracting $f_k^{\gamma_k}(x_k)$, and using (12) yields

$$\nu_k(d_k) - f_k^{\gamma_k}(x_k) = \sum_{j \in \mathbf{J}^k} \lambda_k^j (f_k^j(x_k) - f_k^{\gamma_k}(x_k)) + \sum_{j \in \mathbf{J}^k} \lambda_k^j (c + \beta_k^j) d_k. \tag{15}$$

Hence, solving (11) for $\sum_{j \in \mathbf{J}^k} \lambda_k^j (c + \beta_k^j)$ and substituting into (15) yields

$$\nu_k(d_k) - f_k^{\gamma_k}(x_k) = \sum_{j \in \mathbf{J}^k} \lambda_k^j (f_k^j(x_k) - f_k^{\gamma_k}(x_k)) - \|d_k\|^2 - \sigma_k A d_k. \tag{16}$$

Note that the first group of terms on the right in (16) is nonpositive since $f_k^j(x_k) - f_k^{\gamma_k}(x_k)$ is nonpositive for all $j \in \mathbf{J}^k$ and k. From (14), the nonnegativity of σ_k, and the feasibility of x_k, we have $\sigma_k A d_k = \sigma_k(b - Ax_k) \geq 0$. Therefore

$$\nu_k(d_k) - f_k^{\gamma_k}(x_k) \leq -\|d_k\|^2 \leq 0. \tag{17}$$

The Cauchy-Schwarz inequality and (10) imply

$$-\|c + \beta_k^{\gamma_k}\| \|d_k\| \leq (c + \beta_k^{\gamma_k}) d_k \leq \nu_k(d_k) - f_k^{\gamma_k}(x_k) \tag{18}$$

From (17) and (18) we have the result. □

The following lemma is a result of the test for the new incumbent and Lemma 2. It establishes that there exists a subsequence of $\{d_k\}_{k=1}^{\infty}$ that converges to $\mathbf{0}$ almost surely.

LEMMA 3. *Let $\{d_k\}_{k=1}^{\infty}$ be the sequence of master program solutions. Then there exists a subset of indices, $\mathbf{K}'$, such that $\{d_k\}_{k \in \mathbf{K}'} \to \mathbf{0}$ almost surely.*

The rather lengthy proof of this lemma is included as Appendix A of this paper.

Equation (11) and the assumptions on $\mathbf{P}$, which include the compactness of $\mathbf{X}$, imply that σ_k, the dual multiplier vector associated with the set $\mathbf{X}$, is uniformly bounded. This property is used to prove the following theorem that establishes

the optimality of limiting incumbent points associated with subsequences of $\{d_k\}_{k=1}^{\infty}$ that converge to $\mathbf{0}$.

THEOREM 1. *Let index set* $\mathbf{K}$ *be such that*

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

then x_∞ *is an optimal solution of* (**P**) *with probability* 1.

Proof. Let $\hat{\mathbf{J}}^k \subseteq \mathbf{J}^k$ be the set of indices of the binding inequalities in (4) of ($\mathbf{M}^k$). By Caratheodory's Theorem (see [1]) we may assume that the cardinality of $\hat{\mathbf{J}}^k$, denoted by $p(k) = |\hat{\mathbf{J}}^k|$, satisfies $p(k) \le n_1 + 1$. Let $\{k_j\}_{j=1}^{p(k)}$, denote the indices in the set $\mathbf{J}^k$ that are also in $\hat{\mathbf{J}}^k$, and let $\lambda_k = (\lambda_k^{k_1}, \ldots, \lambda_k^{k_{p(k)}})$, be the vector of associated multipliers.

Suppose we have $\{x_k\}_{k\in\mathbf{K}} \to x_\infty$ and $\{d_k\}_{k\in\mathbf{K}} \to \mathbf{0}$. By the assumptions on (**P**) and Lemma 3, there exists a subset $\mathbf{K}' \subseteq \mathbf{K}$ and a positive integer $p \le n_1 + 1$ such that for all $k \in \mathbf{K}'$ we have $p(k) = p$, $\lambda_k \in \mathbf{R}^p$ for all $k \in \mathbf{K}'$, and from (11)

$$d_k^T + \sum_{j=1}^{p} \lambda_k^{k_j}(c + \beta_k^{k_j}) + \sigma_k A = \mathbf{0}^T, \tag{19}$$

$$\begin{aligned}
&\sigma_k(A(x_k + d_k) - b) = 0,\\
&\{c + \beta_k^{k_j}\}_{k\in\mathbf{K}'} \to (c + \beta^j),\ j = 1, \ldots, p\\
&\{\lambda_k^{k_j}\}_{k\in\mathbf{K}'} \to \lambda^j \ge 0,\ j = 1, \ldots, p,\\
&\sum_{j=1}^{p} \lambda^j = 1,\ \{\sigma_k\}_{k\in\mathbf{K}'} \to \sigma.
\end{aligned} \tag{20}$$

Therefore, the fact that $\{d_k\}_{k\in\mathbf{K}'} \to \mathbf{0}$ implies that we have

$$\sum_{j=1}^{p} \lambda^j(c + \beta^j) + \sigma A = \mathbf{0}^T, \tag{21a}$$

$$\sigma(Ax_\infty - b) = 0. \tag{21b}$$

Lemma 2, (16) and the fact that $\{d_k\}_{k\in\mathbf{K}'} \to \mathbf{0}$ imply that

$$\{f_k^{k_j}(x_k) - f_k^{\gamma_k}(x_k)\}_{k\in\mathbf{K}'} \to 0, \quad \text{if } \lambda^j > 0,\ j = 1, \ldots, p. \tag{22}$$

Let ∂ indicate "subgradient." From (22), Lemma 1, and the fact that for any $x \in \mathbf{X}$ the cuts present in master program ($\mathbf{M}^k$) accumulate at values that underestimate the actual objective (with probability 1), we have

$$(c + \beta^j) \in \partial(cx_\infty + E[Q(x_\infty, \tilde{\omega})]), \text{ almost surely for } j = 1, \ldots, p. \tag{23}$$

(20), (21), and (23) yield stationarity conditions for x_∞ with probability 1. □

A description of how to identify such a subsequence is now given. Since with probability 1 there exists an infinite subset $\mathbf{K}'$ such that $\{d_k\}_{k\in\mathbf{K}'} \to \mathbf{0}$, any accumulation point of $\{x_k\}_{k\in\mathbf{K}'}$ 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. We describe one way to identify such a subsequence. To do so, we introduce a sequence, $\{\delta_k\}_{k=1}^{\infty}$. If we let δ_0 be sufficiently large and define constants $\mu_2 \leq \mu_1 < 1$ that are used to prevent δ from decreasing too rapidly, then we let δ_k be defined as follows:

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

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

$$\mathbf{K}' = \{k : \|d_k\| \leq \delta_{k-1}\}. \tag{24}$$

Clearly, $\mathbf{K}'$ 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 $\{d_k\}_{k\in\mathbf{K}'} \to \mathbf{0}$. Note that the compactness of $\mathbf{X}$ guarantees that the sequence $\{x_k\}_{k\in\mathbf{K}'}$ has accumulation points in $\mathbf{X}$. Thus we obtain the following corollary to Theorem 1.

COROLLARY 1. *Let $\{x_k\}_{k=1}^{\infty}$ be the sequence of incumbent solutions identified by the RSD algorithm and suppose that $\mathbf{K}'$ is the index set defined by* (24) *then every accumulation point of $\{x_k\}_{k\in\mathbf{K}'}$ is optimal, with probability* 1.

4. Termination criteria

In this section several termination rules are listed. Since the piecewise linear approximations are derived from set $\mathbf{V}^k$, termination of the algorithm should be considered only after a sufficiently large number of iterations have passed in which no new dual vertex of the recourse problem has been found. Therefore, consider termination if:

1. The cardinality of $\mathbf{V}^k$ equals the cardinality of $\mathbf{V}^{k-p}$ for $p < k$ a large positive integer (stability of set $\mathbf{V}^k$).

This rule is suggested in [5].

The progress of the incumbent objective value, $f_k^{\gamma_k}(x_k)$ can be monitored. A statistical summary, η_k, of the incumbent objective values can be used to monitor those iterations corresponding to the subsequence defined in (24) (i.e., $k \in \mathbf{K}'$). Termination is considered if the following test is satisfied.

2. $\frac{|f_k^{\gamma_k}(x_k)-\eta_k|}{|f_k^{\gamma_k}(x_k)|} < \epsilon$ (stability of the objective function).

An exponentially smoothed average defined as

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

with $\lambda \in (0, 1)$, and η_0 appropriately chosen could be used.

As a result of Lemma 3, we know that the subsequence of $\{d_k\}_{k=1}^{\infty}$ as defined in (24) converges to $\mathbf{0}$. Therefore termination should not be considered unless $\|d_k\|$ is small. One can use $\{\|d_k\|\}_{k\in\mathbf{K}'}$ to compute a statistic ρ_k, similar to η_k above. The algorithm may be terminated if for $k \in \mathbf{K}'$ we have

3. $\rho_k < \epsilon$ (regularizing term small).

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

5. Computational testing

In this section we report the computational results of applying RSD (with cut dropping), SD, and SD with cut dropping and linear updating (denoted by SDw) in the solution of four test problems. All implementations reestimate the cut associated with the incumbent according to (6).

A brief description of each problem, named SCAGR7, SCRS8, PGP2, and CEP1, including references and the sizes of each stage can be found in Appendix B. Notes on the algorithm implementation are also contained in Appendix B.

Discrete random variables

In order to verify the performance of the algorithms, discrete versions of the problems introduced above were solved. Each algorithm implementation was run over five independent replications. The optimal solution of each problem was determined using the L-shaped method of Van Slyke and Wets [16]. The actual value of the objective function at the terminal incumbent solution was determined by evaluating the subproblem at this point for each possible realization of the discretized random variables.

Table 1. Discrete variable termination summary.

	SD	SDw*	RSD
SCAGR7**			
av. iterations	128.0 (11.0)	130.0 (6.0)	101.0 (0.0)
av. error in obj.	0.0003 (0.0001)	0.0003 (0.0001)	0.0002 (0.0001)
av. dev. from opt.	0.0004 (0)	0.0004 (0)	0.0004 (0)
SCRS8			
av. iterations	107.0 (7.0)	107 (8.0)	107.0 (8.0)
av. error in obj.	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)
av. dev. from opt.	0 (0)	0 (0)	0 (0)
PGP2			
av. iterations	160.0 (61.0)	163.0 (52.0)	173.0 (56.0)
av. error in obj.	0.03 (0.03)	0.02 (0.01)	0.02 (0.01)
av. dev. from opt.	0.04 (0.05)	0.01 (0.02)	0.0018 (0.0004)
CEP1			
av. iterations	315.0 (44.0)	294.0 (69.0)	327.0 (30.0)
av. error in obj.	0.07 (0.03)	0.09 (0.04)	0.07 (0.03)
av. dev. from opt.	0 (0)	0 (0)	0 (0)

*SD with cut dropping, and linear updating.

**Variable weight on $\|d\|^2$ term.

Listed in Table 1 are the average number of iterations, the average error in the terminal incumbent objective estimate as a fraction of the actual objective value at this point (*av. error in obj.*), and the average deviation of the actual objective value at the terminal incumbent from the optimal objective value as a fraction of the optimal objective value (*av. dev. from opt.*). Standard deviations associated with the replications appear in parentheses.

Very little difference between implementations is exhibited in iteration count for three of the four problems. The RSD runs of problem SCAGR7 performed roughly 20% fewer iterations than the SD implementations required to meet all of the termination conditions. However, note that a variable weight on the quadratic term (see Appendix B) was used for this problem due to the dominance of this term for many iterations.

The terminal incumbents of each of the runs for problems SCRS8 and CEP1 were optimal. In each case, the terminal incumbent for problem SCAGR7 yielded an actual objective value with a relative error of only 0.04%. Only in problem PGP2 was there any difference between the SD and RSD runs. In this case, RSD out performed SD with respect to the deviation from the optimal

objective value. With the limited number of runs, however, one cannot tell if this is significant.

As expected, the estimated objective values at the terminal incumbent exhibited deviations from the actual values by as much as 9% (CEP1). This is due to the limited number of realizations of the random variables generated and the subproblem dual bases available before termination. While this fact seems to suggest that termination may have been premature, the optimality (or near optimality) of the terminal incumbents from the RSD implementation for problem PGP2 seems to indicate that improved accuracy of the objective value estimates may not be necessary in order to determine optimality.

Information concerning the subproblem bases is contained in Table 2. Listed are the average number of subproblem dual vertices found (*av. # vertices*) and the average number of these vertices that are active at the terminal incumbent *(av. # active)*. That is, we list the average number of distinct $\pi_j \in \{\text{argmax } \pi(\omega_j - Tx_s) | \pi \in \mathbf{V}^s\}, j = 1 \ldots, s$, where s is the terminal iteration. For example, solving problem CEP1 with RSD produced an average of 33.0 bases with a standard deviation of 1.0. On average, 16 of these, with a standard deviation of 1.0, were active at the terminal incumbent. Thus, roughly half of the bases identified were used to estimate the objective value at the terminal incumbent. From Table 2 we see that a large percentage of the bases revealed during the process play a role at the terminal incumbent for all problems.

The benefits of the linear updating and cut dropping are clearly apparent from Table 2 if we examine the average number of cuts added to the master program *(av. # cuts)* and the average number of times the incumbent cut was reestimated *(av. reestimations)*. With cut dropping (SD and RSD) only 20% of the master program cuts were retained for problems SCAGR7 and SCRS8. This percentage was even smaller for problems PGP2 (<9%) and CEP1 (< 5%).

Note that since only one dual subproblem basis was produced for problem SCRS8, no reestimation of the cut associated with the incumbent solution was necessary for RSD or SDw. With respect to the other problems, SD required that the cut associated with the incumbent solution be reestimated, on average, about four times as often as RSD or SDw. Note that the increase in reestimations observed when the simple $L(x, \omega_k) = 0$ update is used in the SD implementation, is caused by imposing condition (5a). Since the reestimation of this cut requires that the operation indicated by (6) be performed each time one of the conditions in (5) is satisfied, the reduction in effort using the new update is apparent.

Continuous random variables

SD algorithms were developed to solve problems with continuous as well as discrete random variables. Since the RSD results for the discrete examples above indicated a slight advantage with respect to average deviation from optimality and

Table 2. Discrete variable termination summary (continued).

	SD	SDw*	RSD
SCAGR7**			
av. # vertices	6.0 (1.0)	6.0 (1.0)	6.0 (1.0)
av. # active	4.6 (0.5)	4.6 (0.5)	4.2 (0.4)
av. # cuts	64.0 (6.0)	13.4 (0.9)	18.0 (4.0)
av. reestimations	127.0 (11.0)	35.0 (5.0)	16.0 (7.0)
SCRS8			
av. # vertices	1 (0)	1 (0)	1 (0)
av. # active	1 (0)	1 (0)	1 (0)
av. # cuts	54.0 (4.0)	21.3 (0.9)	21.4 (0.7)
av. reestimations	106.0 (8.0)	0 (0)	0 (0)
PGP2			
av. # vertices	15.0 (2.0)	16.0 (3.0)	14.0 (2.0)
av. # active	9.0 (2.0)	10.0 (1.0)	8.0 (1.0)
av. # cuts	81.0 (31.0)	9.0 (1.0)	7.9 (0.4)
av. reestimations	158.0 (61.0)	19.0 (10.0)	24.0 (12.0)
CEP1			
av. # vertices	31.2 (0.7)	31.0 (2.0)	33.0 (1.0)
av. # active	19.4 (0.8)	20.0 (2.0)	16.0 (1.0)
av. # cuts	158.0 (22.0)	7.0 (0.1)	7.1 (0.5)
av. reestimations	313.0 (44.0)	88.0 (19.0)	125.0 (15.0)

*SD with cut dropping, and linear updating.

**Variable weight on $\|d\|^2$ term.

average error in the objective function estimates (see Table 1) over SD and SDw, the RSD algorithm has been applied to each of the four test problems assuming continuous random variables. The random variables are uniformly distributed for problems SCAGR7, SCRS8, and CEP1, and normally distributed in the case of PGP2 as in the original description of the problem [8].

Table 3 reports the average number of iterations, average number of cuts, the average number of dual vertices found, and the average number of these vertices that are active at the terminal incumbent.

With continuous random variables the optimal values of the objective functions are unknown. Therefore, the relative error in the objective value estimates at the terminal incumbent is reported in Table 3 as the average deviation from the sample mean of the terminal incumbent objective value, based on an independent sample of size 2,000, as a fraction of the sample mean objective value (*av. error*

Table 3. RSD results for problems with continuous random variables.

	SCAGR7*	SCRSD	PGP2	CEP1
av. iterations	213 (110)	119 (9.0)	200 (25.0)	376 (150.0)
av. # cuts	15.0 (2.0)	20.2 (0.6)	5.24 (0.05)	7.7 (0.4)
av. # vertices	5.0 (0)	1 (0)	14.0 (2.0)	14.0 (1.0)
av. # active	3.2 (0.04)	1 (0)	11.2 (0.7)	10.0 (0)
av. error in obj.	0.0002 (0.0003)	0.007 (0.004)	0.02 (0.01)	0.06 (0.04)

*Variable weight on $\|d\|^2$ term.

in obj.). With this sample size, under a normal assumption which does not necessarily apply, the sample mean would be within 4% of the actual objective value with 99% confidence. Standard deviations associated with the trials appear in parentheses.

Compared with the discrete versions, the continuous versions of problems SCAGR7, PGP2, and CEP1 required significantly more iterations before meeting the termination conditions. In all cases, the RSD algorithm produced good solutions in the early iterations, however, meeting termination criterion #2 in section 4 delayed termination while the objective function stabilized.

6. Summary

In this paper a stochastic decomposition algorithm with a master program containing a quadratic regularizing term was introduced. This regularizing term makes it possible to bound the size of the master program without sacrificing the SD convergence results.

Two other algorithmic enhancements have been included. A mechanism for updating past cuts was proposed that takes advantage of current information in the iterative process. With this update, the nonnegativity restriction of the recourse problem, which was necessary for the simple update of the original SD algorithm, is relaxed. A condition that adaptively determines when the incumbent cut should be reestimated was also suggested. This condition, coupled with the adaptive updating of the piecewise linear approximation for $f(x)$, should make it possible to avoid the cumbersome reestimation of the incumbent cut at each iteration as suggested in [5].

To evaluate the success of the enhancements, a computational experiment was performed. In these computational experiments we find that regardless of whether a linear or quadratic objective is used in the master program, the RSD and SD algorithms perform quite well as indicated by the deviation from optimality of the terminal incumbents. The addition of both cut dropping and

the adaptive updating scheme showed a marked improvement of RSD over SD in the number of cuts retained and the number of reestimations of the cut associated with the incumbent solution. This improvement was achieved without affecting the iteration count or the objective value error.

The increased efficiency, which the enhancements can provide, makes extension of the SD concept to other models look quite promising. One such extension is investigated in [19]. The problem considered there is one in which the recourse problem is bounded and thus appears in the constraint set rather than as part of the objective function. The solution method proposed for this variation of the standard two-stage stochastic program with recourse involves the introduction of an exact penalty term to the objective function.

Appendix A. Proof of Lemma 3

If the incumbent changes infinitely often, the proof follows from Lemma 6 in [5], and Lemma 2 in Section 3 above. The proof for the case of a finite number of incumbents uses the Langrangian dual to the master program and is in the spirit of Kiwiel [7], which is concerned with unconstrained convex programs. Presented below is a slight variation of the proof that appears in Yakowitz [18]. Credit for the proof of the finite incumbent case goes to S. Sen [20].

LEMMA 3. *Let $\{d_k\}_{k=1}^{\infty}$ be the sequence of master program solutions. Then there exists a subset of indices, $\mathbf{K}'$, such that $\{d_k\}_{k\in\mathbf{K}'} \to \mathbf{0}$ almost surely.*

Proof. Let $\{k_n\}_{n\in\mathbf{N}}$ represent the sequence of iterations at which the incumbent is changed. That is, for every $n \in \mathbf{N}$ (8) is satisfied and $x_{k_n} = z_{k_n}$, $f_{k_n}^{\gamma_{k_n}}(x_{k_n}) = f_{k_n}^{k_n}(z_{k_n})$. If $\mathbf{N}$ is an infinite set then Lemma 6 of [5] applies as long as $\{\gamma_k, k\} \in \mathbf{J}^k$. Thus for $n \in \mathbf{N}$ $\lim_{m\to\infty}\frac{1}{m}\sum_{n=1}^{m}(\nu_{k_{n-1}}(z_{k_n}) - f_{k_n-1}^{\gamma_{k_{n-1}}}(x_{k_{n-1}})) = 0$. Then by Lemma 2, $\lim_{m\to\infty}\frac{1}{m}\sum_{n=1}^{m}\|d_{k_{n-1}}\| = 0$, with probability 1. Since the summand is positive for all n, there exists an infinite subset $\mathbf{N}' \subseteq \mathbf{N}$ such that $\{d_{k_{n-1}}\}_{n\in\mathbf{N}'} \to \mathbf{0}$, almost surely.

If $\mathbf{N}$ is a finite set, there exists $\kappa < \infty$ such that $x_k = x_\infty = x_\kappa$ for all $k \geq \kappa$. The Lagrangian dual to $(\mathbf{M}^k)$ for $k > \kappa$ is given by:

$$\text{Max } \theta_k(\sigma, \lambda) = \inf_d \frac{1}{2}\|d\|^2 + \sigma(Ax_\kappa + Ad - b) + \sum_{j\in\mathbf{J}^k}\lambda^j(f_k^j(x_\kappa) + (c + \beta_k^j)d)$$

$\sigma \geq 0$, $\lambda^j \geq 0$, for $j \in \mathbf{J}^k$ and $\sum_{j\in\mathbf{J}^k}\lambda^j = 1$. For a given σ, λ, the objective is strictly convex and achieves its minimum at a point $d = d_k$ satisfying

$$d_k^T + \sigma A + \sum_{j\in\mathbf{J}^k}\lambda^j(c + \beta_k^j) = \mathbf{0}.$$

Substituting for d_k we obtain:

$(\mathbf{D}^k)$

$$\theta_k^* = \max \theta_k(\sigma, \lambda) = \sigma(Ax_\kappa - b) + \sum_{j \in \mathbf{J}^k} \lambda^j f_k^j(x_\kappa) - \frac{1}{2}\left\| \sum_{j \in \mathbf{J}^k} \lambda^j (c + \beta_k^j) + \sigma A \right\|^2$$

$$s.t. \sum_{j \in \mathbf{J}^k} \lambda^j = 1, \ \lambda^j \geq 0, \ j \in \mathbf{J}^k, \ \sigma \geq \mathbf{0}.$$

Let (λ_k, σ_k) denote an optimal solution to $(\mathbf{D}^k)$.

Recall that $\mathbf{J}^{k+1} = \{j \in \mathbf{J}^k : \lambda_k^j > 0\} \cup \{\kappa, k+1\}$ since $\gamma_k = \kappa$ for all $k \geq \kappa$. It follows that the value of $(\mathbf{D}^{k+1})$, and consequently that of $(\mathbf{M}^k)$, can be bounded using (λ_k, σ_k). Let $s \in [0, 1]$ and define $\overline{\lambda}_k^j = (1-s)\lambda_k^j$ for $j \in \mathbf{J}^{k+1} \backslash \{k+1\}$. Then $\sum_{j \in \mathbf{J}^{k+1} \backslash \{k+1\}} \overline{\lambda}_k^j + s = 1$. Then if we let $\overline{\lambda}_k$ be the vector whose elements are $\overline{\lambda}_k^j$ augmented by s, $(\sigma_k, \overline{\lambda}_k)$ is feasible to $(\mathbf{D}^{k+1})$ for all $s \in [0, 1]$. Thus we have

$$\begin{aligned} \theta_{k+1}^* \geq \theta_{k+1}(\sigma_k, \overline{\lambda}_k) = & \sum_{j \in \mathbf{J}^{k+1} \backslash \{k+1\}} \overline{\lambda}_k^j f_k^j(x_\kappa) + s f_{k+1}^{k+1}(x_\kappa) + \sigma_k(Ax_\kappa - b) \\ & - \frac{1}{2}\left\| \sum_{j \in \mathbf{J}^{k+1} \backslash \{k+1\}} \overline{\lambda}_k^j (c + \beta_k^j) + s(c + \beta_{k+1}^{k+1}) + \sigma_k A \right\|^2. \end{aligned} \tag{25}$$

Note that for $j \in \mathbf{J}^{k+1} \backslash \{\kappa, k+1\}$

$$\alpha_{k+1}^j = \frac{k}{k+1}\alpha_k^j + \frac{1}{k+1}\pi_{k+1}^{k+1}\omega_{k+1} = \alpha_k^j - \frac{1}{k+1}(\alpha_k^j - \pi_{k+1}^{k+1}\omega_{k+1}), \text{ and}$$

$$\beta_{k+1}^j = \frac{k}{k+1}\beta_k^j - \frac{1}{k+1}\pi_{k+1}^{k+1}T = \beta_k^j - \frac{1}{k+1}(\beta_k^j + \pi_{k+1}^{k+1}T).$$

And thus $f_{k+1}^j(x_\kappa) = f_k^j(x_\kappa) - \frac{1}{k+1}(f_k^j(x_\kappa) - \pi_{k+1}^{k+1}(\omega_{k+1} - Tx_\kappa))$. From the assumptions on $(\mathbf{P})$ we therefore have

$$\lim_{k \to \infty} f_{k+1}^j(x_\kappa) - f_k^j(x_\kappa) = 0, \ \lim_{k \to \infty} \|\beta_{k+1}^j - \beta_k^j\| = 0.$$

For $j = \kappa$ we have from Lemma 1

$$\lim_{k \to \infty} f_{k+1}^\kappa(x_\kappa) - f_k^\kappa(x_\kappa) = 0,$$

and assumptions on $(\mathbf{P})$, even with the reestimation, ensure that

$$\lim_{k \to \infty} \|\beta_{k+1}^\kappa - \beta_k^\kappa\| = 0.$$

Therefore, for $j \in \mathbf{J}^{k+1} \backslash \{k+1\}$, we may write

$$f_{k+1}^j(x_k) = f_k^j(x_\kappa) + \epsilon_k^j \text{ and } \beta_{k+1}^j = \beta_k^j + \bar{\epsilon}_k^j, \tag{26}$$

where $\epsilon_\kappa^j \to 0$ and $\|\bar{\epsilon}_k^j\| \to 0$. Substituting (26) into (25) and combining all diminishing terms into ϵ_k we obtain

$$\begin{aligned}
\theta_{k+1}(\sigma_k, \overline{\lambda}_k) &= (1-s) \sum_{j \in \mathbf{J}^{k+1} \setminus \{k+1\}} \lambda_k^j (f_k^j(x_\kappa) + \epsilon_k^j) + s f_{k+1}^{k+1}(x_\kappa) + \sigma_k(Ax_\kappa - b) \\
&\quad -\frac{1}{2}\Bigg\|(1-s) \sum_{j \in \mathbf{J}^{k+1} \setminus \{k+1\}} \lambda_k^j (c + \beta_k^j + \bar{\epsilon}_k^j) \\
&\qquad + s\left(c + \beta_{k+1}^{k+1}\right) + \sigma_k A\Bigg\|^2 \\
&= (1-s) \sum_{j \in \mathbf{J}^{k+1} \setminus \{k+1\}} \lambda_k^j f_k^j(x_\kappa) + s f_{k+1}^{k+1}(x_\kappa) + \sigma_k(Ax_\kappa - b) \\
&\quad -\frac{1}{2}\left\|(1-s)d_k^T + s(c + \beta_{k+1}^{k+1}) + \sigma_k A)\right\|^2 + \epsilon_k \\
&= (1-s) \sum_{j \in \mathbf{J}^{k+1} \setminus \{k+1\}} \lambda_k^j f_k^j(x_\kappa) + s f_{k+1}^{k+1}(x_\kappa) + \sigma_k(Ax_\kappa - b) \\
&\quad -\frac{1}{2}\{\|d_k\|^2 + s^2\|c + \beta_{k+1}^{k+1} + \sigma_k A - d_k^T\|^2 \\
&\quad +2s(c + \beta_{k+1}^{k+1} + \sigma_k A)d_k - 2s\|d_k\|^2\} + \epsilon_k \\
&= (1-s)\left\{\sum_{j \in \mathbf{J}^{k+1} \setminus \{k+1\}} \lambda_k^j f_k^j(x_\kappa) + \sigma_k(Ax_\kappa - b) - \frac{1}{2}\|d_k\|^2\right\} \\
&\quad +s\{f_{k+1}^{k+1}(x_\kappa) + \sigma(Ax_\kappa - b) - (c + \beta_{k+1}^{k+1} + \sigma_k A)d_k\} \\
&\quad +\frac{s}{2}\|d_k\|^2 - \frac{s^2}{2}\|c + \beta_{k+1}^{k+1} + \sigma_k A - d_k^T\|^2 + \epsilon_k.
\end{aligned}$$

Using (14), the definition of θ_k^* and the fact that $z_{k+1} = x_\kappa + d_k$ we have

$$\begin{aligned}
\theta_{k+1}(\sigma_k, \overline{\lambda}_k) &= (1-s)\theta_k^* + s f_{k+1}^{k+1}(z_{k+1}) \\
&\quad +\frac{s}{2}\|d_k\|^2 - \frac{s^2}{2}\|c + \beta_{k+1}^{k+1} + \sigma_k A - d_k^T\|^2 + \epsilon_k.
\end{aligned} \tag{27}$$

Since the incumbent does not change after iteration κ,

$$f_{k+1}^{k+1}(z_{k+1}) - f_{k+1}^\kappa(x_\kappa) \geq \mu(\nu_k(d_k) - f_k^\kappa(x_\kappa)), \text{ for all } k > \kappa. \tag{28}$$

Subtracting $f_{k+1}^\kappa(x_\kappa)$ from both sides of (27), using (28), Lemma 1, and the facts that $\theta_k^* = \nu_k(d_k) + \frac{1}{2}\|d_k\|^2$ and $\theta_{k+1}^* \geq \theta_{k+1}(\sigma_k, \overline{\lambda}_k)$ for all $s \in [0, 1]$, yields

$$\theta_{k+1}^* - f_{k+1}^\kappa(x_\kappa) \geq \theta_k^* - f_k^\kappa(x_\kappa) + \max_{s \in [0,1]}\Big\{-s(1-\mu)(\nu_k(d_k) - f_k^\kappa(x_\kappa))$$

$$-\frac{s^2}{2}\|c+\beta_{k+1}^{k+1}+\sigma_k A-d_k^T\|^2+\epsilon_k\Big\}$$

where $\epsilon_k \to 0$. Assumptions on (**P**) ensure that $\{\nu_k(d_k)-f_k^\kappa(x_\kappa)\}$ and $\{\|c+\beta_{k+1}^{k+1}+\sigma_k A-d_k^T\|^2\}$ are bounded sequences. Thus, there exists a constant $C \in [0, \infty)$ such that for all $k \geq \kappa$

$$C > \text{ Max}\left\{-\frac{1}{2}(1-\mu)[\nu_k(d_k)-f_k^\kappa(x_\kappa)],\ \frac{1}{2}\|c+\beta_{k+1}^{k+1}+\sigma_k A-d_k^T\|^2,\ 1\right\}$$

so that

$$\theta_{k+1}^*-f_{k+1}^\kappa(x_\kappa) \geq \theta_k^*-f_k^\kappa(x_\kappa)+\frac{(1-\mu)^2}{4C}\{\nu_k(d_k)-f_k^\kappa(x_\kappa)\}^2+\epsilon_k. \tag{29}$$

If $\overline{\lim}_{k\to\infty}\nu_k(d_k)-f_k^\kappa(x_\kappa)=r<0$, then since $\epsilon_k \to 0$, for κ large enough the sequence $\{\theta_k^*-f_k^\kappa(x_\kappa)\}_{k=\kappa}^\infty$ is monotone increasing. Since it is bounded from above it has a unique limit. But if follows from (29) that

$$\lim_{k\to\infty}\{(\theta_{k+1}^*-f_{k+1}^\kappa(x_\kappa))-(\theta_k^*-f_{k+1}^\kappa(x_\kappa))\} \geq \frac{(1-\mu)^2}{4C}r^2,$$

a contradiction. Hence $\overline{\lim}_{k\to\infty}\nu_k(d_k)-f_k^\kappa(x_\kappa)=0$ and the result follows from Lemma 2. □

Appendix B. Problem description and implementation notes

Problems SCAGR7 and SCRS8 are two-stage stochastic versions of deterministic multistage problems described in [6] and [2]. PGP2 can be found in [8]. CEP1 has been described in [18] and [19]. Sizes of the two stages appear after each description. The quantity $\hat{m}_2$ indicates the number of the second-stage constraints that had random right-hand sides.

SCAGR7 is a dairy farm expansion planning model 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 ($n_1 = 20$, $m_1 = 15, n_2 = 40$, $m_2 = 38$, $\hat{m}_2 = 3$).

SCRS8 is a dynamic energy model for the transition from fossil fuels to renewable energy resources. This problem models U.S. options for a minimum cost transition from oil and gas to synthetic fuels while meeting future energy demands. 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 ($n_1 = 37$, $m_1 = 28$, $n_2 = 38$, $m_2 = 28$, $\hat{m}_2 = 3$).

PGP2 is a power generation planning model in which the planner has a choice of several types of power plant equipment in order to meet the needs of the community that the plant serves. The planner wishes to determine the capacity of

each type of equipment to be installed that minimizes the sum of the installation costs and the expected operating costs. $n_1 = 4$, $m_1 = 2$, $n_2 = 12$, $m_2 = 7$, $\hat{m}_2 = 3$).

CEP1 is a capacity expansion planning problem for a manufacturing plant that produces several parts on several machines. The objective is to minimize the cost of new capacity and the expected cost of weekly labor plus tooling ($n_1 = 8$, $m_1 = 9$, $n_2 = 15$, $m_2 = 7, \hat{m}_2 = 3$).

The SD and RSD algorithms have been implemented on a VAX 8650 at the University of Arizona. The Fortran program utilizes the XMP subroutines of Marsten [9] for solving subproblems ($\mathbf{S}^k$) and the linear master program in the SD version. The quadratic master program of RSD is solved using the *ZQPCVX* algorithm of [12].

The following parameters were used in all computer replications: $\mu = 0.25$ (the new incumbent parameter); $\tau = 20$ (cut re-estimation parameter); $\epsilon = 0.0005$ (termination tolerance); and $\lambda = 0.25$ (exponential smoothing parameter).

The convergence analysis of Section 3 requires that only those cuts with indices in $\mathbf{J}^k$ need be retained when the quadratic proximal term is present in the master program. However, since no theoretical support for dropping cuts from the linear SD master program is presented here, in iterations during which the incumbent does not change cuts that are tight at the current incumbent solution, (i.e., at most $n_1 + 1$ cuts such that $\alpha_k^j + \beta_k^j x_k = \alpha_k^{\gamma_k} + \beta_k^{\gamma_k} x_k$, $j \in \mathbf{J}^{k-1}$) are also retained. Since the dimension of n_1 is not very large in the problems considered, these cuts are retained in the RSD implementation as well so that the cut dropping schemes are identical. Therefore, at most $2n_1 + 3$ cuts are present in the master programs of the implementations that drop cuts in any iteration.

The algorithmic implementation of RSD did not include an attempt to identify the subsequence indicated by the indices in $\mathbf{K}'$ defined in (24). The reader will recall that $\mathbf{K}'$ is the set of indices that corresponds to a subsequence of $\{d_k\}_{k=1}^{\infty}$ that converges to the zero vector. Nevertheless, we wanted an indication of stability of the objective function before termination. Therefore, we required that termination rule 2 be satisfied with $\eta_k = \lambda f_k^{\gamma_k}(x_k) + (1-\lambda)\eta_{k-1}$ and $\lambda \in [0, 1]$. The RSD implementation also required that for $k \in N$ (iterations in which 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 $\|d_k\| < \epsilon$. The SD implementation substituted $\Theta_k = \nu_{k-1}(d_{k-1}) - f_{k-1}^{\gamma_{k-1}}(x_{k-1})$, for $\|d_k\|$ in the above statistical summary. In addition, termination was not permitted until 100 iterations were completed and termination was considered only if the cardinality of the set $\mathbf{V}^k$ remained the same for at least 50 iterations.

In some cases it may be necessary to include a variable weight on the quadratic term in ($\mathbf{M}^k$) in order to prevent this term from dominating, and thus slowing the progress of the algorithm, for many iterations. This dominance becomes apparent when the incumbent changes for many consecutive iterations while $\|d\|^2$ remains nearly constant. Whenever this condition is detected, a weight, u, is reduced and used to dampen the term $\|d\|^2$ in ($\mathbf{M}^k$). If we initialize u as $u_0 = \frac{1}{2}$

the objective function in $(\mathbf{M}^k)$ is then replaced by:

$$\text{Min } u_k\|d\|^2 + \nu_k(d).$$

Algorithmically, a counter can keep track of the number of consecutive iterations during which the incumbent has changed. During these iterations $\|d\|$ can also be averaged. Then when the counter is larger than some fixed integer, and the current value of $\|d\|$ is comparable to the average value during these iterations, the weight, u_k, is reduced. During these iterations u_k is defined by $u_k = \max\{\frac{1}{2}u_{k-1}, \hat{u}\}$, where $\hat{u}$ is a fixed, small scalar. Otherwise, $u_k = u_{k-1}$. Thus whenever sustained progress is detected one allows the procedure to accelerate by becoming greedy. Convergence of the method using a variable weight, as defined above, can be shown with only slight modification of the proofs of Section 3 since u_k is bounded below by $\hat{u}$.

References

1. M.S. Bazaraa and C.M. Shetty, Nonlinear Programming, Theory and Algorithms, John Wiley & Sons, Inc.: New York, NY, 1979.
2. J.R. Birge, "Decomposition and partitioning methods for multi-stage stochastic linear programs," Oper. Res. vol. 33 pp. 989–1007, 1985.
3. B.C. Eaves and W.I. Zangwill, "Generalized cutting plane algorithms," SIAM J. Control vol. 9 pp. 529–542, 1971.
4. H.I. Gassmann, "MSLiP: A computer code for the multistage stochastic linear programming problem," Math. Prog. vol. 47 pp. 407–423, 1990.
5. J.L. Higle and S. Sen, "Stochastic decomposition: An algorithm for two stage linear programs with recourse," Math. Oper. Res. vol. 16 pp. 650–669, 1991.
6. J.K. Ho and E. Loute, "A set of staircase linear programming test problems," Math. Prog. vol. 20 pp. 245–250, 1981.
7. K.C Kiwiel, "Methods of descent for nondifferentiable optimization," Lecture Notes in Mathematics no. 1133, Springer-Verlag: Berlin, 1985.
8. F.V. Louveaux and Y. Smeers, "Optimal investment for electricity generation: A stochastic model and a test problem," in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R.J-B. Wets, Eds., Springer-Verlag: Berlin, 1988.
9. R.E. Marsten, XMP Technical Reference Manual, Department of Management Information Systems, College of Business and Public Administration: University of Arizona, Tucson, AZ, 1987.
10. R. Mifflin, "An algorithm for constrained optimization with semismooth functions," Math. Oper. Res. vol. 2 pp. 191–207, 1977.
11. R. Mifflin, "A modification and an extension of Lemarechal's algorithm for nonsmooth minimization," in Nondifferential and Variational Techniques in Optimization, D.C. Sorensen and R.J-B Wets, Eds., Math. Prog. Study vol. 17 pp. 77–90, 1982.
12. M.J.D. Powell, ZQPCVX Algorithm, Department of Applied Mathematics and Theoretical Physics, University of Cambridge: Cambridge, England, 1986.
13. R.T. Rockafellar and R.J-B. Wets, "A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming," Math. Prog. Study vol. 28 pp. 63–93, 1987.
14. A. Ruszczynski, "A regularized decomposition method for minimizing a sum of polyhedral functions," Math. Prog. vol. 35 pp. 309–333, 1986.
15. A. Ruszczynski, "A linearization method for nonsmooth stochastic programming problems," Math. Oper. Res. vol. 12 pp. 32–49, 1987.

16. R. Van Slyke and R.J.-B. Wets, "L-shaped linear programs with applications to optimal control and stochastic programming," SIAM J. Appl. Math. vol. 17 pp. 638–663, 1969.
17. R.J.-B. Wets, "Stochastic programming: Solution techniques and approximation schemes," in Mathematical Programming: The State of the Art, A. Bachem, M. Groetschel, B. Korte, Eds., Springer-Verlag: pp. 506–603, Berlin, 1982.
18. D.S. Yakowitz, "Two-stage stochastic linear programming: Stochastic decomposition approaches," Ph.D. Dissertation, University of Arizona, 1991.
19. D.S. Yakowitz, "An exact penalty algorithm for recourse-constrained stochastic linear programs," Appl. Math. and Comp. vol. 49 pp. 39–62, 1991.
20. J.L. Higle and S. Sen, "Finite Master Programs in Regularized Stochastic Decomposition," Math. Prog. (to appear) 1994.