Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

A COMBINED DETERMINISTIC AND SAMPLING-BASED SEQUENTIAL BOUNDING METHOD FOR STOCHASTIC PROGRAMMING

Péguy Pierre-Louis
Güzin Bayraksan

Department of Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721, USA

David P. Morton

Graduate Program in Operations Research
University of Texas at Austin
Austin, TX 78712, USA

ABSTRACT

We develop an algorithm for two-stage stochastic programming with a convex second stage program and with uncertainty in the right-hand side. The algorithm draws on techniques from bounding and approximation methods as well as sampling-based approaches. In particular, we sequentially refine a partition of the support of the random vector and, through Jensen's inequality, generate deterministically valid lower bounds on the optimal objective function value. An upper bound estimator is formed through a stratified Monte Carlo sampling procedure that includes the use of a control variate variance reduction scheme. The algorithm lends itself to a stopping rule theory that ensures an asymptotically valid confidence interval for the quality of the proposed solution. Computational results illustrate our approach.

1 INTRODUCTION

We develop an algorithm for approximately solving a class of two-stage stochastic programs by combining Jensen's inequality with a Monte Carlo estimator. Our approach is related to a classic approximation scheme in stochastic programming, called the *sequential approximation method* (SAM) (Edirisinghe and Ziemba 1992, Frauendorfer 1992, Huang, Ziemba, and Ben-Tal 1977, Kall, Ruszczyński, and Frauendorfer 1988). A SAM uses upper and lower bounds on the expected value of the future-cost function. These bounds exploit convexity and, in this classic approach, are iteratively refined. Our approach is similar except that we replace the more computationally expensive of these two bounds with a Monte Carlo estimator. Valid termination of an optimization algorithm with noisy bounds on the optimality gap demands attention to accompanying sequential issues. We describe how to deal with those issues, and we further propose a variance-reducing estimator that exploits special structures associated with the iterative refinement scheme.

For a two-stage stochastic program, under certain convexity properties, lower bounds are typically based on Jensen's inequality. While there are a number of exceptions (see, e.g., Birge and Dulá 1991, Birge and Teboulle 1989, Birge and Wets 1989, Kall 1991, Morton and Wood 1999), many of the upper bounding schemes are rooted in the Edmundson-Madansky inequality (Edmundson 1957, Madansky 1959, Madansky 1960). In contrast to Jensen's lower bound, which requires only one function evaluation for each element of the partition, the number of function evaluations required to apply the Edmundson-Madansky inequality to a random vector with independent components can grow exponentially with the dimension of the random vector. In this paper, we develop a method that combines deterministically valid lower bounds on the objective function (those used by a SAM through Jensen's inequality) with Monte Carlo sampling-based upper bound estimates. We use this approach within the sequential framework of Bayraksan and Morton (2011) to devise an algorithm that generates high-quality solutions with high probability. We call the resulting algorithm the *sampling-based sequential approximation method* (SSAM).

SSAM integrates the two algorithmic frameworks as follows. First, it uses SAM to generate the sequence of candidate solutions. That is, it sequentially refines a partition of the random vector's support, and employs Jensen's inequality conditionally on each cell of the partition, to generate a candidate solution. Next, SSAM uses Monte Carlo sampling-based upper bounds, alleviating the computational bottleneck caused by calculating Edmundson-Madansky upper bounds in SAM. Calculating the upper bound via Monte Carlo sampling corresponds to estimating the objective function value for the current candidate solution, which is the usual sample mean estimator. We reduce the variance of this estimator within SSAM using stratified sampling and control variates. Finally, the procedure iteratively increases the sample size and terminates according to the sequential sampling framework of Bayraksan and Morton (2011).

SSAM uses Jensen's lower bound applied to each cell of the partition and then weighted by the associated probability mass and summed over all cells. While Jensen's lower bound requires modest effort to compute, this approach to lower bounding leads to three potential drawbacks. First, the convexity assumption required to use Jensen's inequality restricts the class of problems to which SSAM applies (see Section 2). Second, for some problems, Jensen's lower bound can be loose. Third, because SSAM forms a partition of the random vector's support, just as in SAM, it can be less effective as the dimension of the random vector grows. The use of tighter lower bounds (Dokov and Morton 2005, Topaloglu 2009), through means other than refining the partition, may improve SSAM's performance in these cases, although we do not pursue such alternatives here.

2 PROBLEM CLASS

We consider two-stage stochastic programs of the form:

$$z^* = \min_{x \in X} \; \{f(x) \equiv c(x) + \mathbb{E}[Q(x,\xi)]\}, \tag{SP}$$

where

$$Q(x,\xi) = \begin{array}{ll} \min\limits_{y \geq 0} & q(y) \\ \text{s.t.} & g(y) \leq h(\xi) - T(x,\xi). \end{array} \tag{1}$$

We assume the random vector ξ has finite dimension which we denote by d_ξ, and known distribution, which we assume independent of decision x. We denote the set of feasible first stage decisions by $X \subseteq \mathbb{R}^{d_x}$. Here, $c : X \to \mathbb{R}$, $Q : X \times \text{co}(\Xi) \to \mathbb{R}$, $q : \mathbb{R}^{d_y} \to \mathbb{R}$, $g : \mathbb{R}^{d_y} \to \mathbb{R}^{m_y}$, $h : \text{co}(\Xi) \to \mathbb{R}^{m_y}$, and $T : X \times \text{co}(\Xi) \to \mathbb{R}^{m_y}$, where Ξ is ξ's support and $\text{co}(\cdot)$ denotes the convex hull operator. We assume that: the expectation in (SP) is finite for all $x \in X$; model (1) has a finite optimal solution achieved by a feasible y for every element of $X \times \text{co}(\Xi)$; and, model (SP) has a finite optimal solution achieved on X. We further assume:

(A1) $Q(x,\cdot)$ is convex on $\text{co}(\Xi)$ for all $x \in X$; and,
(A2) ξ has independent components, and $h(\cdot)$ and $T(x,\cdot)$ are affine on $\mathbb{R}^{d_\xi}$ for all $x \in X$.

Assumption (A1) allows us to use Jensen's inequality to form lower bounds on the objective function. Under assumption (A2), a sufficient condition to ensure that assumption (A1) holds is that $q(\cdot)$ and $g(\cdot)$ are convex functions; see, e.g., Fiacco and Kyparisis (1986). Given these conditions, convexity of $Q(\cdot,\xi)$ on $\text{co}(X)$ holds, provided T is convex on $\text{co}(X)$ for each fixed element of $\text{co}(\Xi)$. From the perspective of the underlying probability model, assumption (A2), with the affine dependence of $h(\cdot)$ and $T(x,\cdot)$ on ξ, permits capturing first order notions of dependency of the random parameters in (SP), such as those that arise in commonly-used linear factor models. With these assumptions in place, model (SP) can take a variety of forms. For instance, it can be a stochastic linear program or a stochastic convex program or X can have integrality restrictions, leading to a stochastic integer program. This changes the requisite tools for solving the lower bounding problems (see Sections 3 and 4 for details) but the algorithmic framework of SSAM that we put forward remains the same.

3 SEQUENTIAL APPROXIMATION METHOD

Significant parts of the algorithmic machinery from the literature on sequential approximation methods are inherited by our SSAM, and so we briefly review the key ideas behind a SAM here. For more details on

SAMs, see, e.g., Kall, Ruszczyński, and Frauendorfer (1988). Let $a_i < b_i$, $i = 1,2,\dots,d_\xi$, be such that $\Xi \subseteq \prod_{i=1}^{d_\xi}[a_i,b_i]$, where we allow for the possibility that $a_i = -\infty$ and/or $b_i = \infty$.

3.1 Bounding the Objective Function

We now review the lower bounding function on $f(x)$ that arises from Jensen's inequality applied to a partition of Ξ. Let $\mathscr{S} = \{\Xi_\ell : \ell = 1,2,\dots,\nu\}$ denote a partition of Ξ; that is, $\Xi_i \bigcap \Xi_j = \emptyset, i \neq j$ and $\bigcup_{\ell=1}^{\nu} \Xi_\ell = \Xi$. Let $p^\ell = \mathbb{P}(\xi \in \Xi_\ell)$ denote the probability mass of cell ℓ and let $\bar{\xi}^\ell = \mathbb{E}[\xi | \xi \in \Xi_\ell]$ denote ξ's mean, conditional on being in cell ℓ. Then, $\mathbb{E}[Q(x,\xi)] = \sum_{\ell=1}^{\nu} p^\ell \mathbb{E}[Q(x,\xi)|\xi \in \Xi_\ell]$. Since $Q(x,\cdot)$ is convex on $\operatorname{co}(\Xi_\ell)$, application of Jensen's inequality on each cell results in a lower bounding function induced by the partition $\mathscr{S}$,

$$L_{\mathscr{S}}(x) \equiv c(x) + \sum_{\ell=1}^{\nu} p^\ell Q(x,\bar{\xi}^\ell) \leq f(x). \tag{2}$$

In what follows, we assume that each cell of the partition has form $\Xi_\ell = \prod_{i=1}^{d_\xi}[a_i^\ell, b_i^\ell]$, where $a_i \leq a_i^\ell < b_i^\ell \leq b_i$, $\ell = 1,2,\dots,\nu$. Under assumption (A2), this means that forming p^ℓ and $\bar{\xi}^\ell$ only requires computing univariate expectations.

Let $U_{\mathscr{S}}(x)$ denote a deterministically-valid upper bound that, like the Jensen bound, is adapted to the partition $\mathscr{S}$, and has the following form:

$$U_{\mathscr{S}}(x) \equiv c(x) + \sum_{\ell=1}^{\nu} p^\ell \mathbb{E}[Q(x,\xi_{U_\ell})] \geq f(x). \tag{3}$$

In the Edmundson-Madansky bound, ξ_{U_ℓ} is a random variable whose support is the set of extreme points of Ξ_ℓ, and so when $\Xi_\ell = \prod_{i=1}^{d_\xi}[a_i^\ell, b_i^\ell]$ this means evaluating $\mathbb{E}[Q(x,\xi_{U_\ell})]$ requires 2^{d_ξ} function evaluations. See, for example, Frauendorfer (1992), Kall, Ruszczyński, and Frauendorfer (1988) for further details on this conditional version of the Edmundson-Madansky inequality.

3.2 Generating Solutions

Given a partition $\mathscr{S}$, we generate a candidate solution, $\hat{x}$, in SAM by optimizing $L_{\mathscr{S}}(x)$ over X:

$$\hat{x} \in \operatorname*{argmin}_{x \in X} L_{\mathscr{S}}(x). \tag{4}$$

For a given $x \in X$, let μ_x denote its optimality gap, $\mu_x = f(x) - z^*$. For a candidate solution found by (4), we can bound $\mu_{\hat{x}}$ via $\mu_{\hat{x}} \leq U_{\mathscr{S}}(\hat{x}) - L_{\mathscr{S}}(\hat{x})$. This follows immediately from (2) and (3) and the fact that $\hat{x}$ minimizes the lower bound. SAM sequentially refines the partition $\mathscr{S}$ to improve the candidate solution as well as the bound on the optimality gap. The procedure stops when the upper bound on $\mu_{\hat{x}}$ is sufficiently small (see Section 3.4 for details).

3.3 Refining Partitions

Three issues arise when refining a partition $\mathscr{S} = \{\Xi_\ell : \ell = 1,2,\dots,\nu\}$. First, which cell(s) of the partition should we refine? Second, given a cell, perpendicular to which axis should we construct a hyperplane to form two new cells? Third, at which point on the designated axis should the splitting hyperplane be positioned? In SAM, and in SSAM of the next section, we answer these questions for a fixed $\hat{x}$ found via (4). We only sketch basic ideas here and leave it to Section 4.4 to detail specifics for SSAM.

To answer the first question, note that the error bound $U_{\mathscr{S}}(\hat{x}) - L_{\mathscr{S}}(\hat{x})$ on $\mu_{\hat{x}}$ may be expressed as a sum of contributions from each cell, where that from cell ℓ is given by:

$$p^\ell \left[\mathbb{E}[Q(\hat{x},\xi_{U_\ell})] - Q(\hat{x},\bar{\xi}^\ell)\right]. \tag{5}$$

We refine those cells with the largest errors, as defined by (5).

Assume that cell ℓ is to be split. Jensen's inequality arises from the best linear approximation of $Q(\hat{x},\cdot)$ across a cell, which is the first-order Taylor approximation formed at the conditional mean of the cell, $\bar{\xi}^{\ell}$. Assume for the moment that $Q(\hat{x},\cdot)$ is twice differentiable (as is our computational example in Section 5). Forming a second-order Taylor approximation of $Q(\hat{x},\cdot)$ at $\bar{\xi}^{\ell}$ along the ith coordinate axis, computing the expected value of the difference with the first-order approximation, and selecting the axis with the largest such expected error means that we choose axis:

$$i^* \in \operatorname*{argmax}_{i=1,\ldots,d_\xi} \left[\left. \frac{\partial^2 Q(\hat{x},u)}{\partial u_i^2} \right|_{u=\bar{\xi}^{\ell}} \operatorname{var}(\xi_i | \xi \in \Xi_\ell) \right]. \tag{6}$$

In the case of a two-stage stochastic program with recourse, $Q(\hat{x},\cdot)$ is not even once differentiable and other approaches are required. Birge and Wets (1986) suggest using differences of subgradients at the endpoints of each edge of a bounded cell to make this determination. Frauendorfer and Kall (1988) found that examining differences between first order Taylor approximations and actual recourse function values along each edge often further improves the refinement procedure.

Finally, we turn to the third question. Once the axis to split, i^*, has been determined, the corresponding component of the conditional mean may be used as the point to position the splitting hyperplane (Birge and Wets 1986, Kall, Ruszczyński, and Frauendorfer 1988).

3.4 SAM and Convergence Properties

Algorithm 1 summarizes the discussion in Sections 3.1-3.3, providing a brief algorithmic statement of SAM. Going forward, we use index k to denote an iteration of an algorithm. So, $\mathscr{S}_k$ denotes the partition at iteration k of SAM, which contains ν_k cells, and we denote the corresponding candidate solution, obtained in (4) with $\mathscr{S} = \mathscr{S}_k$, by $\hat{x}_k$.

We now turn to the limiting behavior of SAM as $k \to \infty$. Let

$$f_k(x,\xi) = c(x) + \sum_{\ell=1}^{\nu_k} \mathbb{I}(\xi \in \Xi_\ell) \left[Q(x,\bar{\xi}^{\ell}) + \nabla_\xi Q(x,\bar{\xi}^{\ell})(\xi - \bar{\xi}^{\ell}) \right],$$

where we use $\nabla_\xi Q(x,\bar{\xi}^{\ell})$ to denote a (sub)gradient of $Q(x,\cdot)$ at $\bar{\xi}^{\ell}$ and $\mathbb{I}(\cdot)$ denotes the indicator function. Note that $L_{\mathscr{S}_k}(x) = \mathbb{E} f_k(x,\xi)$. In this way, $f_1(x,\xi)$ is integrable with $\mathbb{E} f_1(x,\xi) = c(x) + Q(x,\mathbb{E}\xi)$. And, because the partition for $k+1$ is a refinement of that at k, we have $f_k(x,\xi) \le f_{k+1}(x,\xi) \le c(x) + Q(x,\xi)$. Thus, if we refine the partitions in such a way to ensure pointwise convergence,

$$\lim_{k\to\infty} f_k(x,\xi) = c(x) + Q(x,\xi), \tag{7}$$

then we can employ the monotone convergence theorem to infer

$$\lim_{k\to\infty} \mathbb{E} f_k(x,\xi) = c(x) + \mathbb{E} Q(x,\xi). \tag{8}$$

Moreover, $\mathbb{E} f_k(x,\xi) \le \mathbb{E} f_{k+1}(x,\xi)$ coupled with (8), lower semicontinuity of $\mathbb{E} f_k(x,\xi)$ and continuity of $c(x) + \mathbb{E} Q(x,\xi)$ ensures that $L_{\mathscr{S}_k}(x) = \mathbb{E} f_k(x,\xi)$ converges uniformly to $c(x) + \mathbb{E} Q(x,\xi)$. Under these assumptions $\lim_{k\to\infty} L_{\mathscr{S}_k}(\hat{x}_k) = z^*$ and every accumulation point of $\{\hat{x}_k\}_{k=1}^{\infty}$ solves (SP). For these types of results under weaker assumptions via the notion of epi-convergence, see, e.g., Birge and Wets (1986). Frauendorfer (1992) discusses conditions on the refinement schemes to ensure (7) holds.

Algorithm 1: Sequential Approximation Method (SAM)

```
step 0    (initialization)
    select $\varepsilon > 0$ and let $k = 1$, $\mathscr{S}_k = \{\Xi\}$ and $\nu_k = |\mathscr{S}_k|$ ;
step 1    (lower bounding problem)
    let $\hat{x}_k \in \text{argmin}_{x \in X} L_{\mathscr{S}_k}(x)$;
step 2    (stopping criterion)
    if $U_{\mathscr{S}_k}(\hat{x}_k) - L_{\mathscr{S}_k}(\hat{x}_k) \le \varepsilon \cdot \max\{|U_{\mathscr{S}_k}(\hat{x}_k)|, |L_{\mathscr{S}_k}(\hat{x}_k)|\}$ then stop;
step 3    (partition refinement)
    refine partition $\mathscr{S}_k$, let $k = k+1$, update $\nu_k$; let $\bar{\xi}^\ell = \mathbb{E}[\xi | \xi \in \Xi_\ell]$ and $p^\ell = \mathbb{P}(\xi \in \Xi_\ell)$, $\forall \ell$;
    goto step 1;
```

4 SAMPLING-BASED SEQUENTIAL APPROXIMATION METHOD

This section describes an algorithm similar to SAM of the previous section except that we replace the computationally taxing Edmundson-Madansky upper bound by a sampling-based estimate of $\mathbb{E}[Q(x,\xi)]$. The error bounds provided by our new sampling-based SAM (SSAM) have statistical error. This leads us to characterize solution quality in SSAM via a confidence interval rather than the deterministically-valid bounds on the optimality gap of SAM. Moreover, in our sampling-based adaptation of SAM, we iteratively test whether an estimator of the optimality gap satisfies a termination criterion, an inescapably *sequential* procedure. Such sequential procedures can exhibit premature termination with associated poor coverage results for their interval estimators; see, e.g., Glynn and Whitt (1992). To avoid premature termination and ensure an asymptotically valid confidence interval on the optimality gap of the solution proposed by SSAM, we adapt the sequential sampling framework in Bayraksan and Morton (2011) to SSAM (see Section 4.3).

SSAM generates candidate solutions and calculates a lower bound in the same manner as SAM. SSAM also improves this lower bound and hence, the solutions generated, by refining the partition of the random vector's support, again, much like SAM. When forming a sampling-based estimator of $\mathbb{E}[Q(x,\xi)]$, the partitioning scheme inherent in SSAM suggests that we employ stratified sampling to reduce the variance of the associated estimator (see Section 4.1). We attempt to further reduce variance via a control variate, where the control on each cell of the partition is a linear approximation of $Q(x,\cdot)$ that use subgradient information, which is already available (see Section 4.2). The motivation for using a control variate estimator lies in the notion that the quality of these linear approximations should improve with finer partitions of Ξ. Indeed, the partitioning scheme already chooses and splits cells for the purpose of reducing nonlinearity of $Q(x,\cdot)$ in order to tighten the Jensen bound (see Section 3.3).

4.1 Stratified Monte Carlo Sampling

Given a partition $\mathscr{S}$ of Ξ with ν cells, and a solution $x \in X$, we estimate $f(x)$ by stratified Monte Carlo sampling. Suppose we have a sample size of N to be allocated among the ν cells; i.e., $N = \sum_{\ell=1}^{\nu} n_\ell$, where n_ℓ denotes the number of observations drawn from the conditional distribution of ξ given that $\xi \in \Xi_\ell$. We propose a stratified estimator of the form

$$\bar{U}_{\mathscr{S},N}(x) = c(x) + \sum_{\ell=1}^{\nu} p^\ell \bar{U}^\ell_{n_\ell}, \tag{9}$$

where $\bar{U}^\ell_{n_\ell}$ estimates $\mathbb{E}[Q(x,\xi)\,|\,\xi \in \Xi_\ell]$ with sample size n_ℓ. For now, we may view $\bar{U}^\ell_{n_\ell}$ as the standard sample mean estimator, but below we modify it to be a control variate estimator. We prefer a stratified estimator to a crude Monte Carlo estimator because the stratified approach reduces variance, guides selection of cells for refinement (see Section 4.4), and facilitates a control variate to further reduce variance (see Section 4.2).

Ignoring integrality, we allocate the sample size N across the cells of the partition according to

$$n_\ell = p^\ell N, \quad \ell = 1,2,\ldots,\nu, \tag{10}$$

which ensures variance reduction. Substituting the sample size formula (10) into $\mathrm{var}[\bar{U}_{\mathscr{S},N}(x)]$ yields

$$\mathrm{var}[\bar{U}_{\mathscr{S},N}(x)] = \frac{1}{N}\sum_{\ell=1}^{\nu} p^{\ell}\sigma_{\ell}^{2}(x) \equiv \frac{1}{N}\sigma^{2}(x), \tag{11}$$

and we have the central limit theorem

$$\sqrt{N}\left(\bar{U}_{\mathscr{S},N}(x) - f(x)\right) \Rightarrow N(0,\sigma^{2}(x)), \quad \text{as} \quad N \to \infty \quad \forall x \in X,$$

where $\Rightarrow$ denotes convergence in distribution. We estimate $\sigma^2(x)$ using conditional sample variance estimators from each cell, $s_{\ell}^{2}(x)$, via

$$s^{2}(x) = \sum_{\ell=1}^{\nu} p^{\ell} s_{\ell}^{2}(x). \tag{12}$$

In implementation, we round sample sizes up to the next largest integer.

4.2 A Control Variate Estimator

We develop a control variate scheme that we employ separately in estimating $\mathbb{E}[Q(x,\xi)\,|\,\xi \in \Xi_{\ell}]$ at each cell ℓ of the partition. Recall $\bar{\xi}^{\ell} = \mathbb{E}[\xi\,|\,\xi \in \Xi_{\ell}]$ and consider the following first-order Taylor approximation of $Q(x,\cdot)$ as the control random variable

$$\Gamma_{\ell}(x,\xi) = Q(x,\bar{\xi}^{\ell}) + \nabla_{\xi}Q(x,\bar{\xi}^{\ell})(\xi - \bar{\xi}^{\ell}), \qquad \ell = 1,2,\ldots,\nu,$$

where, again, $\nabla_{\xi}Q(x,\bar{\xi}^{\ell})$ denotes a (sub)gradient of $Q(x,\cdot)$ at $\bar{\xi}^{\ell}$. When we have $Q(x,\xi)$ in analytic form (see, e.g., Section 5), we can obtain $\nabla_{\xi}Q(x,\bar{\xi}^{\ell})$ analytically. Or, if $T(x,\xi) = T(x)$ then the vector of Lagrange multipliers on the constraints of (1), say $\pi(x,\bar{\xi}^{\ell})$, is a subgradient of $Q(x,\cdot)$ at $\bar{\xi}^{\ell}$. More generally, when $T(x,\cdot)$ depends on the random parameter, we can use the chain rule to obtain the desired subgradient.

Note that $\mathbb{E}[\Gamma_{\ell}(x,\xi)\,|\,\xi \in \Xi_{\ell}] = Q(x,\bar{\xi}^{\ell})$ and hence $\Gamma_{\ell}(x,\xi)$ has known conditional mean. SSAM obtains value $Q(x,\bar{\xi}^{\ell})$ when solving the lower bounding problem, and the subgradient value may also be obtained immediately (e.g., if it has value $\pi(x,\bar{\xi}^{\ell})$), or with minimal additional effort (e.g., via the chain rule with an affine $T(x,\xi)$). Therefore, the control, $\Gamma_{\ell}(x,\xi)$, may be formed with essentially no additional computation.

Let

$$W_{\ell}(x,\xi,\lambda) = Q(x,\xi) - \lambda[\Gamma_{\ell}(x,\xi) - Q(x,\bar{\xi}^{\ell})].$$

We have subtracted from $Q(x,\xi)$ a random variable with conditional mean zero. Choosing λ to minimize $\mathrm{var}[W_{\ell}(x,\xi,\lambda)\,|\,\xi \in \Xi_{\ell}]$ yields

$$\lambda^{*} = \frac{\mathrm{cov}[Q(x,\xi),\,\Gamma_{\ell}(x,\xi)\,|\,\xi \in \Xi_{\ell}]}{\mathrm{var}[\Gamma_{\ell}(x,\xi)\,|\,\xi \in \Xi_{\ell}]}.$$

We can estimate λ^{*} by

$$\hat{\lambda}_{n_{\ell}} = \frac{\widehat{\mathrm{cov}}[Q(x,\xi),\,\Gamma_{\ell}(x,\xi)\,|\,\xi \in \Xi_{\ell}]}{\widehat{\mathrm{var}}[\Gamma_{\ell}(x,\xi)\,|\,\xi \in \Xi_{\ell}]}, \tag{13}$$

using the sample covariance and sample variance based on the n_{ℓ} observations for cell ℓ. We then form the following estimate of $\mathbb{E}[Q(x,\xi)\,|\,\xi \in \Xi_{\ell}]$:

$$\bar{W}_{n_{\ell}}^{\ell}(\hat{\lambda}_{n_{\ell}}) = \frac{1}{n_{\ell}}\sum_{j=1}^{n_{\ell}} W_{\ell}(x,\xi^{j},\hat{\lambda}_{n_{\ell}}).$$

Under a joint normality assumption, Lavenberg and Welch (1981) obtain a confidence interval associated with the point estimate, $\bar{W}^{\ell}_{n_\ell}(\hat{\lambda}_{n_\ell})$. Nelson (1990) shows that asymptotically, the normality assumption is unnecessary, obtaining in our notation that

$$\sqrt{n_\ell}\left(\bar{W}^{\ell}_{n_\ell}(\hat{\lambda}_{n_\ell}) - \mathbb{E}[Q(x,\xi)\,|\,\xi \in \Xi_\ell]\right) \Rightarrow N(0, \bar{\sigma}^2_\ell(x)), \quad \text{as } n_\ell \to \infty \;\; \forall x \in X, \tag{14}$$

where $\bar{\sigma}^2_\ell(x) = (1-R^2)\,\sigma^2_\ell(x)$ and R^2 is the squared conditional correlation coefficient between $Q(x,\xi)$ and $\Gamma_\ell(x,\xi)$.

Instead of estimating λ^*, we can simply use a deterministic value, say, $\lambda = 1$. Then, $\bar{W}^{\ell}_{n_\ell}(1)$ is a sample mean of i.i.d. random variables it again satisfies a central limit theorem, albeit with a larger variance term than that of (14). In forming our sampling-based estimate of $f(x)$ can replace $\bar{U}^{\ell}_{n_\ell}$ in (9) by $\bar{W}^{\ell}_{n_\ell}(\hat{\lambda}_{n_\ell})$ or by $\bar{W}^{\ell}_{n_\ell}(1)$. In these cases, we would similarly replace $\sigma^2_\ell(x)$ in (11) with $\bar{\sigma}^2_\ell(x)$ or its analog for $\bar{W}^{\ell}_{n_\ell}(1)$, with corresponding conditional sample variance estimators in (12). In Section 5 we report computational results for both $\bar{W}^{\ell}_{n_\ell}(\hat{\lambda}_{n_\ell})$ and $\bar{W}^{\ell}_{n_\ell}(1)$.

4.3 Rules to Increase the Sample Size and Stop

At the kth iteration of SSAM we use G_k to estimate the current optimality gap $\mu_{\hat{x}_k}$, where

$$G_k = \bar{U}_{\mathscr{S}_k,N_k}(\hat{x}_k) - L_{\mathscr{S}_k}(\hat{x}_k).$$

Extending the notation as in SAM, we use N_k to denote the total sample size at iteration k. To simplify the discussion, we take $\bar{U}_{\mathscr{S}_k,N_k}(\hat{x}_k)$, the point estimate of $f(\hat{x}_k)$, to be the stratified-sampling estimator of equation (9) but it could instead be that estimator with $\bar{U}^{\ell}_{n_\ell}$ replaced by $\bar{W}^{\ell}_{n_\ell}(\hat{\lambda}_{n_\ell})$ or $\bar{W}^{\ell}_{n_\ell}(1)$. Associated with G_k is a sample variance estimator, s^2_k from equation (12).

Following the sequential sampling framework of Bayraksan and Morton (2011), SSAM stops when G_k falls below a pre-specified factor of the sample standard deviation, s_k; i.e., SSAM terminates at iteration

$$T = \inf_{k\geq 1}\{k : G_k \leq h' s_k + \varepsilon'\}. \tag{15}$$

When the procedure stops, a statement on the quality of the candidate solution, $\hat{x}_T$, is made as a $100(1-\alpha)\%$ approximate confidence interval on its optimality gap of the form

$$[0, hs_T + \varepsilon].$$

Here, $h > h' > 0$ and $\varepsilon > \varepsilon' > 0$ are prespecified constants. The terms ε and ε' are typically small, relative to h and h', and ensure finite stopping. (We discuss how to select h and h' in Section 5.) Using a sample size schedule N_k as prescribed in Bayraksan and Morton (2011)

$$N_k \geq \left(h - h'\right)^{-2}\left(c_{p,q} + 2pk^{2q/r}\right), \tag{16}$$

where $q > 1$, $p > 0$ and $c_{p,q} = \max\{2\ln\left(\sum_{j=1}^{\infty}\exp[-pj^{2q/r}]/\sqrt{2\pi}\alpha\right), 1\}$. Under a finite variance assumption,

$$\lim_{h \downarrow h'} \mathbb{P}(\mu_{\hat{x}_T} \leq hs_T + \varepsilon) \geq 1 - \alpha.$$

In (16), $q > 1$ and $p > 0$ are parameters that can be chosen to minimize the computational effort of the sequential sampling procedure. For details on the set of conditions to ensure asymptotic validity and how to select p and q, see Bayraksan and Morton (2011).

4.4 Refining Partitions

We describe in Section 3.3 how to refine partitions in SAM. That discussion carries over to SSAM directly, except that the error contributed by each cell ℓ to the current estimate of the optimality gap would revise from (5) to $p^\ell[\bar{U}^\ell_{n_\ell} - Q(\hat{x}_k, \bar{\xi}^\ell)]$. However, we can further revise this by taking into account the stopping criteria (15). In particular, we can choose to split cell ℓ if

$$p^\ell \left[\bar{U}^\ell_{n_\ell} - Q(\hat{x}_k, \bar{\xi}^\ell) - h' s_{k,\ell}(\hat{x}_k)\right] > 0. \tag{17}$$

If (17) fails to hold for any cell then the stopping criteria is satisfied because $\sum_{\ell=1}^{\nu} p^\ell s_\ell \leq \left(\sum_{\ell=1}^{\nu} p^\ell s_\ell^2\right)^{1/2}$. If we find this rule to be too aggressive in splitting cells we can restrict attention to those cells with the largest left-hand side of (17). In our implementation, we do this in a matter depicted in Algorithm 2. Our implementation further uses (6) to select the axis to split, and we split at the conditional mean.

4.5 Algorithm Statement

Algorithm 2 details SSAM, beginning with initialization of the parameters, $h, h', \varepsilon, \varepsilon'$, α, q and p. The initial partition $\mathscr{S}_1$ is the entire sample space and so $\nu_1 = 1$. The list of cells to be considered for splitting, `List`, is initially empty. In step 1 of the algorithm, the current partition is refined by selecting cells from `List` to be partitioned. For each cell Ξ_ℓ with $\ell \in$ `List` we split Ξ_ℓ by selecting the axis according to (6) and splitting at the conditional mean as indicated in Section 3.3. In the first iteration the `List` is empty, and so we just solve the lower-bounding problem using Jensen's bound with the single "scenario," $\bar{\xi} = \mathbb{E}\xi$.

Algorithm 2: Sampling-based Sequential Approximation Method (SSAM)

```
step 0  (initialization)
    select $h > h' > 0$, $\varepsilon > \varepsilon' > 0$, $0 < \alpha < 1$, $q > 1$ and $p > 0$ ;
    let $k = 1$, $\Xi_k = \Xi$, $\mathscr{S}_k = \{\Xi_k\}$, List=$\emptyset, \nu_k = 1$, $LB_k = -\infty$;
step 1  (partitioning and lower bounding problem)
    for $\ell \in$ List do
        split $\Xi_\ell$ and update $\mathscr{S}_k$; compute $p^{\ell'}$ and $\bar{\xi}^{\ell'}$ for two new cells;
    end
    $\hat{x}_k \in \text{argmin}_{x \in X} L_{\mathscr{S}_k}(x)$,  $\nu_k = |\mathscr{S}_k|$,  $LB_k = L_{\mathscr{S}_k}(\hat{x})$;
step 2  (upper bound estimation)
    $N_k = \left\lceil (h - h')^{-2} \left(c_{p,q} + 2pk^{2q/r}\right) \right\rceil$;
    for $\ell$=1 to$\nu_k$ do
        allocate $n_\ell = \lceil p^\ell N_k \rceil$ samples to cell $\ell$; form $\bar{U}^\ell_{n_\ell}$ and $s^2_{k,\ell}(\hat{x}_k)$;
    end
    $\bar{U}_{\mathscr{S}_k,N_k}(\hat{x}_k) = c(\hat{x}_k) + \sum_{\ell=1}^{\nu_k} p^\ell \bar{U}^\ell_{n_\ell}$,   $s_k^2 = \sum_{\ell=1}^{\nu_k} p^\ell s^2_{k,\ell}(\hat{x}_k)$,   $G_k = \bar{U}_{\mathscr{S}_k,N_k}(\hat{x}_k) - L_{\mathscr{S}_k}(\hat{x}_k)$;
step 3  (stopping criterion)
    if $G_k \leq h' s_k + \varepsilon'$ then
        output $\hat{x}_k$ and confidence interval $[0, hs_k + \varepsilon]$ on $\mu_{\hat{x}_k}$ and stop;
    end
step 4  (update List)
    Let $0 \leq r < 1$,  $\Delta_k = G_k - h' s_k$,  $\delta_k^\ell = p^\ell \left(\bar{U}^\ell_{n_\ell} - Q(\hat{x}_k, \bar{\xi}^\ell) - h' s_k^\ell\right)$;
    List = $\left\{\ell : \sum_{\ell \in \texttt{List}} \delta_k^\ell > r\Delta_k, \text{ and } \delta_k^\ell \geq \delta_k^{\ell_1} \;\; \forall \ell_1 \notin \texttt{List}\right\}$ and |List| is minimal;
    Set $k = k + 1$ and goto step 1;
```

Step 2 chooses a sample size according to (16), allocates samples to each cell proportional to the probability mass of that cell, and forms the conditional sample mean and conditional sample variance. In each cell of the partition, we form these by generating i.i.d. observations from the distribution of ξ, conditional on ξ being in that cell. These samples are also independent of samples generated in previous iterations. We then form a point estimate of the objective function value for the current candidate solution under stratified sampling according to (9) and the associated sample variance estimator according to (12). After forming the point estimate of the optimality gap, G_k, we check the stopping criterion in step 3. While we describe the algorithm in step 2 for the simplest case of stratified sampling, it is straightforward to modify step 2 in an attempt to further reduce variance using the control variate scheme described in Section 4.2. Step 4 of the algorithm selects cells of the current partition to consider for refinement. Instead of selecting all cells according to condition (17), it selects the cells that contribute at least $100r\%$ to the current violation. In our numerical experiments, a value of $r = 0.5$ is found to give a good balance between the number of cells in the partition and the convergence of the algorithm; so, we use this value in our experiments in Section 5.

5 NUMERICAL RESULTS

In this section, we experiment with SSAM on an asset allocation problem with m assets defined as

$$z^* = \max_{x \in X} \mathbb{E}[u(x,\xi)], \tag{18}$$

where $X = \{x : \sum_{j=1}^m x_j = 1, x_j \geq 0, j = 1,\ldots,m\}$, $u(x,\xi) = 1 - e^{-\rho(\xi x)}$ and $\xi x = \sum_{j=1}^m \xi_j x_j$. We use problem data for model (18) from Partani, Morton, and Popova (2006), which specifies the mean vector and covariance matrix for a 14-asset model in which the random return vector ξ follows a multivariate normal distribution. We depart from Partani, Morton, and Popova (2006) in that we use an exponential, rather than a power, utility function, but we use a risk parameter of $\rho = 0.812211$ in an attempt to match their optimal solution. Our optimal solution, x^*, has three non-zero components ($x_4 = 0.0786$, $x_9 = 0.7317$, and $x_{10} = 0.1897$, using the same ordering as in Partani, Morton, and Popova 2006) and $z^* = 0.5881$.

All problem instances were solved using CVX version 1.21 (Grant and Boyd 2008, Grant and Boyd 2011) in MATLAB on a 2.93GHz Dell Xeon multi-processor computer with 9 GB of memory, using only one processor. We used the pre-defined MATLAB utilities to generate the normal random variates. We set $q = 1.5$, $p = 4.67 \times 10^{-3}$ and $c_{p,q} = 9.689$ according to the guidelines in Bayraksan and Morton (2009) with $\alpha = 0.1$, $\varepsilon' = 1 \times 10^{-8}$, $\varepsilon = 2 \times 10^{-8}$, and we determine $\Delta h = h - h'$ so that the sample size at the first iteration $k = 1$ is at least as large as some initial target sample size, N_0. To find suitable values for h', we carryout pilot runs with a moderate sample size n in order to get a sense of the range of values of the ratio G_k/s_k, which dictates termination in SSAM. We can then select h' to be slightly smaller than the average of values of the G/s ratio observed during the pilot runs.

5.1 Variance Reduction

We study variance reduction in SSAM for the different estimators we have proposed. Our simplest estimator, a stratified estimator, allocates samples proportional to each cell's probability mass according to equation (10). This stratified estimator with proportional sampling (STP) already reduces variance, and to assess this reduction we also employ a crude Monte Carlo (CMC) estimator. We can use the control variate estimator of Section 4.2 within the stratified sampling scheme in an attempt to further reduce variance. This estimator has a parameter λ, and we can either fix $\lambda = 1$ (SCV1) or we can estimate λ ($\widehat{\text{SCV}}$) by the $\hat{\lambda}_{n_\ell}$ of equation (13).

We compare the sample variances of these four estimators as follows. We use $h' = 0.4275$, $N_0 = 100$, and we made independent 100 runs of SSAM in which the sampling-based upper bound estimator uses the $\widehat{\text{SCV}}$ procedure. For each run, we compute the other estimators (CMC, STP and SCV1) in the background using the same sample sizes, as the algorithm progresses. For each estimator, we take the average of the variance estimates at iteration k, s_k^2, over the 100 runs. Table 1 compares the estimated variances between each pair of these sampling schemes by computing the ratio of the average of these variances as the procedure progresses for the first 10 iterations.

Table 1: Ratio of variance estimators for each pair of sampling scheme.

k	CMC/$\widehat{\text{SCV}}$	STP/$\widehat{\text{SCV}}$	SCV1/$\widehat{\text{SCV}}$	CMC/SCV1	STP/SCV1	CMC/STP
1	11.60	12.26	1.15	10.11	10.69	0.95
2	83.50	35.64	1.82	45.75	19.53	2.34
3	277.11	70.83	1.77	156.33	39.96	3.91
4	394.64	89.41	1.70	231.85	52.53	4.41
5	531.26	114.21	2.74	193.84	41.67	4.65
6	954.93	160.30	2.52	379.20	63.66	5.96
7	1387.37	199.13	2.54	547.25	78.55	6.97
8	1461.46	204.69	3.00	486.51	68.14	7.14
9	2180.77	256.79	3.71	587.52	69.18	8.49
10	3925.98	393.51	3.25	1207.85	121.06	9.98

According to Table 1, $\widehat{\text{SCV}}$ dominates the other estimators. The second column indicates that $\widehat{\text{SCV}}$ reduces variance by a factor of 10-3900 over crude Monte Carlo sampling, the third column indicates $\widehat{\text{SCV}}$ improves over the simple stratified estimator by factors of 12-390, and the fourth column indicates that estimating λ improves variance ratios by 1-3. Variance reduction of the stratified estimators ($\widehat{\text{SCV}}$, SCV1 and STP) grows over CMC as the algorithm proceeds and the partition refines, but this is particularly true for the control variate estimators because their linear controls provide better approximations as k grows. Table 1 suggests the following ordering in terms of variance reduction over CMC: $\widehat{\text{SCV}} >$ SCV1 $>$ STP $>$ CMC. Similar results are obtained for larger initial sample size values $N_0 = 200$, 300, 400, and 500. Since the sampling scheme $\widehat{\text{SCV}}$ results in the largest variance reduction, we use that estimator to form the sampling-based upper bound in SSAM in the remainder of the results we report in this section.

5.2 Results Summary

Table 2 summarizes empirical performance measures of SSAM for initial sample size values of $N_0 = 100$, 200, 300, 400 and 500. For each value of the initial sample size, we report 90% confidence intervals on the following: stopping iteration T; the sample size used for upper-bounding at termination N_T; the number of scenarios (number of cells in the partition) used for the lower-bounding problem at termination; $hs_T + \varepsilon$, the width of the confidence interval on the optimality gap; and, the coverage probability p based on the results of 100 independent runs. The last column of Table 2 reports the proportion of z^* represented by the width of the 90% CI on the optimality gap, $[0, hs_T + \varepsilon]$, obtained by the procedure. Overall, the algorithm performs well on our asset-allocation model: the empirical coverage probability is 100%, the procedure requires a modest number of iterations to stop, and the number of cells in the partition is relatively small. However, the confidence intervals on the optimality gap appear to be somewhat conservative. For instance, even though the CI width on the optimality gap ranges from 0.72% to 0.97% of the optimal value, more than 90% of the solutions, $\hat{x}_T$, that the procedure produced are actually within 0.05% of optimality.

Table 2: Empirical results.

N_0	(h, h')	T	N_T	# Cells	$hs_T + \varepsilon$	$\hat{p}$	% Opt
100	(0.7389, 0.4275)	13.94 ± 1.25	107.54 ± 1.02	21.89 ± 2.29	$4.1 \times 10^{-3} \pm 1.3 \times 10^{-3}$	1.00 ± 0.00	0.92
200	(0.6477, 0.4275)	20.48 ± 2.34	226.60 ± 4.39	47.14 ± 6.25	$3.4 \times 10^{-3} \pm 0.9 \times 10^{-3}$	1.00 ± 0.00	0.74
300	(0.6073, 0.4275)	22.02 ± 2.76	347.53 ± 8.20	64.59 ± 9.67	$4.2 \times 10^{-3} \pm 1.5 \times 10^{-3}$	1.00 ± 0.00	0.97
400	(0.5832, 0.4275)	34.13 ± 3.94	514.97 ± 17.56	129.54 ± 17.02	$3.6 \times 10^{-3} \pm 1.4 \times 10^{-3}$	1.00 ± 0.00	0.86
500	(0.5668, 0.4275)	34.30 ± 4.30	650.18 ± 24.80	148.51 ± 20.79	$3.0 \times 10^{-3} \pm 1.2 \times 10^{-3}$	1.00 ± 0.00	0.72

6 CONCLUSIONS

We have developed a sequential procedure, denoted SSAM, that at each iteration, generates a candidate solution by solving a lower-bounding problem derived from Jensen's inequality applied on a partition of the random vector's support. This lower bound, coupled with a stratified sampling-based estimate of the objective function value at the candidate solution and a control variate estimator, provides a point estimate of the optimality gap. The procedure stops when this estimate falls below a pre-specified fraction of the sample standard deviation. Otherwise, we refine the partition, increase the sample size and repeat the procedure. We applied SSAM to an asset allocation problem. Our results indicate that the control variate estimator on a stratified sample significantly reduces variance and SSAM performs well on this test problem.

ACKNOWLEDGMENTS

This work has been supported by the National Science Foundation through grants CMMI-0653916, CMMI-0800676, and EFRI-0835930 and the Defense Threat Reduction Agency through grant HDTRA1-08-1-0029.

REFERENCES

Bayraksan, G., and D. P. Morton. 2009. “Assessing solution quality in stochastic programs via sampling”. In *Tutorials in Operations Research*, edited by M. R. Oskoorouchi, 102–122. INFORMS.

Bayraksan, G., and D. P. Morton. 2011. “A Sequential Sampling Procedure for Stochastic Programming”. *To appear in Operations Research*.

Birge, J. R., and J. H. Dulá. 1991. “Bounding separable recourse functions with limited distribution information”. *Annals of Operations Research* 30:277–298.

Birge, J. R., and M. Teboulle. 1989. “Upper Bounds on the Expected Value of a Convex Function using Gradient and Conjugate Function Information”. *Mathematics of Operations Research* 14:745–759.

Birge, J. R., and R. Wets. 1986. “Designing approximation schemes for stochastic optimization problems, in particular, for stochastic programs with recourse”. *Mathematical Programming Study* 27:54–102.

Birge, J. R., and R. Wets. 1989. “Sublinear upper bounds for stochastic programs with recourse”. *Mathematical Programming* 43:131–149.

Dokov, S. P., and D. P. Morton. 2005. “Second-order lower bounds on the expectation of a convex function”. *Mathematics of Operations Research* 30:662–677.

Edirisinghe, N. C. P., and W. T. Ziemba. 1992. “Tight bounds for stochastic convex programs”. *Operations Research* 40:660–677.

Edmundson, H. P. 1957. “Bounds on the expectation of a convex function of a random variable”. Technical report, The Rand Corporation Paper 982, Santa Monica, California.

Fiacco, A. V., and J. Kyparisis. 1986. “Convexity and Concavity Properties of Optimal Value Function in Parametric Nonlinear Programming”. *Journal of Optimization Theory and Applications* 48:95–126.

Frauendorfer, K. 1992. *Stochastic Two-Stage Programming*. Lecture Notes in Economics and Mathematical Systems 392. Springer-Verlag, Berlin.

Frauendorfer, K., and P. Kall. 1988. “A Solution Method for SLP Recourse Problems with Arbitrary Multivariate Distributions—The Independent Case”. *Problems of Control and Information Theory* 17:177–205.

Glynn, P. W., and W. Whitt. 1992. “The Asymptotic Validity of Sequential Stopping Rules for Stochastic Simulations”. *The Annals of Applied Probability* 2 (1): 180–198.

Grant, M., and S. Boyd. 2008. “Graph implementations for nonsmooth convex programs”. In *Recent Advances in Learning and Control*, edited by V. Blondel, S. Boyd, and H. Kimura, Lecture Notes in Control and Information Sciences, 95–110. Springer-Verlag Limited. http://stanford.edu/~boyd/graph_dcp.html. Accessed 17 October 2011.

M. Grant and S. Boyd 2011, April. “CVX: Matlab Software for Disciplined Convex Programming, version 1.21”. http://cvxr.com/cvx. Accessed 17 October 2011.

Huang, C. C., W. T. Ziemba, and A. Ben-Tal. 1977. “Bounds on the expectation of a convex function of a random variable: with applications to stochastic programming”. *Operations Research* 25:315–325.

Kall, P. 1991. “An upper bound for SLP using first and total second moments”. *Annals of Operations Research* 30:267–276.

Kall, P., A. Ruszczyński, and K. Frauendorfer. 1988. "Approximation techniques in stochastic programming". In *Numerical Techniques for Stochastic Optimization*, edited by Y. Ermoliev and R. Wets, 33–64. Springer-Verlag, Berlin.

Lavenberg, S. S., and P. D. Welch. 1981. "A perspective on the use of control variates to increase the efficiency of Monte Carlo simulations". *Management Science* 27:322–335.

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

Madansky, A. 1960. "Inequalities for stochastic linear programming problems". *Management Science* 6:197–204.

Morton, D. P., and R. K. Wood. 1999. "Restricted-recourse bounds for stochastic linear programming". *Operations Research* 47:943–956.

Nelson, B. L. 1990. "Control Variate Remedies". *Operations Research* 38:359–375.

Partani, A., D. Morton, and I. Popova. 2006. "Jackknife estimators for reducing bias in asset allocation". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. Perrone, F. Wieland, J. Liu, B. Lawson, D. Nicol, and R. Fujimoto, 783–791. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Topaloglu, H. 2009. "A Tighter Variant of Jensen's Lower Bound for Stochastic Programs and Separable Approximations to Recourse Functions". *European Journal of Operations Research* 199:315–322.

AUTHOR BIOGRAPHIES

PÉGUY PIERRE-LOUIS is a Ph.D. candidate in the Department of Systems and Industrial Engineering at the University of Arizona. His research focuses on algorithmic developments in Monte Carlo sampling-based methods for stochastic programming. His email address is pierrelp@email.arizona.edu.

DAVID P. MORTON is an Engineering Foundation Professor in the Graduate Program in Operations Research at The University of Texas at Austin. His research interests include computational stochastic programming, including simulation-based approximations in stochastic programming, and interdiction modeling. His email address is morton@mail.utexas.edu and his web page is www.me.utexas.edu/∼orie/Morton.html.

GÜZİN BAYRAKSAN is an Assistant Professor in the Department of Systems and Industrial Engineering at the University of Arizona. Her research interests include stochastic optimization, particularly Monte Carlo sampling-based methods for stochastic programming, with applications to water resources management. Her email address is guzinb@sie.arizona.edu and her web page is www.sie.arizona.edu/faculty/guzinb.