A Sequential Sampling Procedure for Stochastic Programming

Güzin Bayraksan
Department of Systems and Industrial Engineering, University of Arizona, Tucson, Arizona 85721, guzinb@sie.arizona.edu

David P. Morton
Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin, Austin, Texas 78712, morton@mail.utexas.edu

We develop a sequential sampling procedure for a class of stochastic programs. We assume that a sequence of feasible solutions with an optimal limit point is given as input to our procedure. Such a sequence can be generated by solving a series of sampling problems with increasing sample size, or it can be found by any other viable method. Our procedure estimates the optimality gap of a candidate solution from this sequence. If the point estimate of the optimality gap is sufficiently small according to our termination criterion, then we stop. Otherwise, we repeat with the next candidate solution from the sequence under an increased sample size. We provide conditions under which this procedure (i) terminates with probability one and (ii) terminates with a solution that has a small optimality gap with a prespecified probability.

Subject classifications: programming: stochastic; simulation: efficiency; statistics: sampling.

Area of review: Simulation.

History: Received April 2007; revisions received March 2009, July 2009; accepted September 2009.

1. Introduction

Stochastic programming provides a means for decision making under uncertainty by incorporating random variables and probabilistic statements into optimization models. A major challenge in analyzing stochastic programs of practical size is having to consider a large, sometimes infinite, number of scenarios. This usually leads to intractable models, even when specially designed algorithms are used. Monte Carlo sampling-based methods provide an attractive approximation when the number of stochastic parameters is large. These methods replace probabilistic statements that appear in the model (e.g., expectation) with sampling-based estimators (e.g., sample mean). They are usually justified asymptotically, by providing conditions under which the approximating solutions solve the stochastic program as the sample size grows to infinity. However, practical implementation requires a finite sample size. A question of interest is, then, what this sample size should be to have a good approximate solution. One approach to this problem is to sequentially increase the sample size until we reach a good solution. Such a procedure requires reliable rules to stop and to increase the sample size. Moreover, a statement regarding the quality of the resulting solution is needed.

We develop such a sequential sampling procedure for a class of stochastic programs. In our procedure, we assume we are given as input a sequence of candidate solutions, that has at least one limit point that solves the original stochastic program. Such a sequence can be generated by solving a series of sampling problems with increasing sample size, but we allow candidate solutions to be generated by any method satisfying the above condition. Given a candidate solution, we assess its quality with increasing sample size. We measure quality via the candidate solution’s optimality gap. At each iteration when assessing the candidate solution’s quality, we can: (i) reuse previously generated sample observations and add additional, newly generated samples, or (ii) generate an entire new set of observations. Efficient warm start procedures for solving the next iteration’s stochastic program favor the former choice. The risk of persisting with a “bad” set of samples suggests we should occasionally choose the latter option. We investigate this trade-off. We terminate when a stopping criterion is satisfied, and we prove that asymptotically this procedure yields a high-quality solution with a desired probability.

Research on sequential sampling methods began in earnest in the 1940s and such methods have been successfully applied to problems in statistics, reliability, and statistical clinical testing (Ghosh and Sen 1991, Ghosh et al. 1997). In sequential sampling, the sample size is not fixed, but depends on the observations collected so far, and hence is random. Sequential estimation relates closely to the study of random walks hitting prespecified sets (Gut 1988). A classic sequential problem involves forming a fixed-width (Chow and Robbins 1965) or relative-width (Nadas 1969) confidence interval (CI) for the mean by sequentially increasing the sample size. When simulating stochastic systems, the run length takes the place of...
the sample size. For steady-state simulations, Law (2007, pp. 529–532) surveys sequential methods for constructing fixed-width and relative-width CIs for the mean performance measure of a stochastic system. Glynn and Whitt (1992) provide conditions under which asymptotic validity of sequential stopping rules can be achieved. More recent work in the area of simulation involves selecting from a number of alternative system designs using sequential sampling; see Kim and Nelson (2001, 2006).

The stochastic programs we consider represent a large class of problems found in statistics and operations research. In statistics, a generalization of maximum likelihood estimators, called M-estimators (Huber 1981), are an example of this form. Sequential sampling procedures have been developed for M-estimators; see Hlávka (2003) and references therein. These procedures apply to differentiable objective functions and focus on estimating an optimum solution. In the optimization problems we consider, the objective function is frequently nondifferentiable and we are indifferent to how close we are to the set of optimal solutions as long as the candidate solution’s objective function value is close to the optimal value, i.e., the optimality gap is small.

Even though our class of stochastic programs can be found under various names in the literature, we mainly focus on stochastic programs with recourse. Whereas our main results hold for more general models, our computational results are for two-stage stochastic linear programs with recourse (SLP-2), where the objective function is convex and typically nonsmooth. Stochastic quasigradient algorithms can be applied in such cases; they mimic steep-descent, in which gradients or subgradients are replaced by sampling-based estimates. Ermoliev (1988) surveys such methods and Pflug (1988) surveys step-size and stopping rules. One advantage of stochastic quasigradient methods is that they have the potential to handle decision-dependent stochasticity. (We return to this issue in §2.) However, when applied to SLP-2, they make limited use of special structure.

The L-shaped method (Van Slyke and Wets 1969) better exploits SLP-2’s special structure and can handle a modest number of scenarios. Dantzig and Glynn (1990) and Infanger (1992) use importance sampling to reduce variance in an L-shaped method. Higle and Sen (1996b) develop an L-shaped method with sampling-based cuts, using a single stream of observations where the cuts are updated to ensure desirable asymptotics. For stochastic global optimization, Norkin et al. (1998) sample within a branch-and-bound algorithm. There has been work on how to stop these procedures and how to assess their solution quality (Dantzig and Infanger 1995; Higle and Sen 1991, 1996a), but sequential issues that arise have received little attention.

Instead of embedding sampling in an optimization algorithm, another approach is to first sample observations and then simply solve the resulting problem. There is a significant literature on large sample size properties of this approach (e.g., Shapiro 2003). The sequence of candidate solutions our procedure requires could be generated either by these “internal” or “external” sampling methods.

For convex, piecewise-linear stochastic programs that have a unique, sharp optimum, Shapiro et al. (2002) provide insight as to the sample size needed to find the optimal solution via large deviations theory. When independent samples are drawn at each iteration, Homem-de-Mello (2003) studies rates at which the sample sizes must grow to ensure consistency of the objective function estimator, and he derives associated error statements in the spirit of the law of the iterated logarithm. For stochastic nonlinear programs, Polak and Royset (2008) propose a procedure that approximately minimizes the computational effort required to reduce an initial optimality gap by a prespecified fraction, in the context of so-called diagonalization schemes.

Morton (1998) develops stopping rules for algorithms that use asymptotically normal optimality gap estimates, formed as a difference of upper- and lower-bound estimators. For minimization problems, upper-bound estimators can be formed for a fixed feasible solution, and asymptotic normality is easy to achieve. A natural lower-bound estimator comes from minimizing a sample-mean objective function (Mak et al. 1999, Norkin et al. 1998). In general, this estimator is not asymptotically normal, and so the approach of Morton (1998) does not apply. We overcome this difficulty, allowing use of the nonnormal optimized sample-mean estimator. We improve on Morton (1998) in two other ways: Our CI on the optimality gap uses the (observable) sample variance estimator instead of the (unknown) population variance, and we develop procedures under weaker moment conditions. Importantly, our main results do not require independent and identically distributed (i.i.d.) sampling, and hence apply when using other sampling schemes that are designed, e.g., to reduce variance. Another attractive feature of our approach is its flexibility in how observations can be generated at each iteration. One option is to use a single stream of observations. At each iteration, we augment the existing set of observations with a few new samples. Alternatively, the previous observations can be discarded and we can generate an entirely new set of observations. Intermediate options also exist, and are permitted by the theory we develop.

The next section begins with the class of stochastic programs we consider, provides assumptions for our sequential procedure, and discusses how our paper relates to the simulation-optimization literature. Section 3 outlines our sequential procedure, stating the stopping rule, the rule to increase the sample sizes, and a statement regarding the quality of the solution obtained. In §4, we establish properties of the sequential procedure under a moment generating function (MGF) assumption (§4.1) and under weaker moment conditions (§4.2). Section 5 discusses how to choose free parameters of the sequential procedure to
minimize computational effort. In §6, we apply the procedure to two-stage stochastic linear programs with recourse and present and discuss computational results. We end the paper with a summary and future research directions (§7).

2. Framework

We seek a high-quality (optimal or near-optimal) solution to a stochastic program of the form

$$z^* = \min_{x \in X} Ef(x, \xi), \quad (SP)$$

where the expectation is with respect to the random vector $\xi$. We assume $\xi$ is of finite dimension $d_\xi$, the distribution does not depend on $x$, and that we can sample from it. We further assume $X \neq \emptyset$, $X \subset \mathbb{R}^d$ is compact, $E \sup_{x \in X} |f(x, \xi)| < \infty$, and $f(\cdot, \xi)$ is lower semicontinuous (lsc) on $X$, with probability one (w.p.1). This ensures $Ef(\cdot, \xi)$ is lsc, and hence (SP) has a finite optimal solution achieved on $X$. Later we impose more restrictive moment conditions on $f(\cdot, \xi)$. We also assume $f(x, \xi)$ can be evaluated given a realization of $\xi$ and $x \in X$.

The procedure we propose works as follows: At iteration $k$, we are given a candidate solution, $\hat{x}_k \in X$. We select a sample size, $n_k$, and evaluate this candidate solution. The quality of the candidate solution is defined via its optimality gap, $Ef(\hat{x}_k, \xi) - z^*$: the smaller the optimality gap, the higher the quality. The procedure stops when the optimality gap estimate falls below a certain level. Otherwise, we continue with $\hat{x}_{k+1} \in X$ and $n_{k+1} \geq n_k$. Let $X^*$ denote the set of optimal solutions to (SP). We assume the following with respect to the sequence of candidate solutions.

A1. The sequence of feasible candidate solutions $\{\hat{x}_k\}$ has at least one limit point in $X^*$, w.p.1.

Such a sequence can be found by solving a series of sampling problems

$$z^*_n = \min_{x \in X} \frac{1}{n} \sum_{i=1}^n f(x, \xi_i), \quad (SP_n)$$

with optimal solutions $x^*_n$ and with sample sizes $n = m_k$, such that $m_k \to \infty$ as $k \to \infty$. Mild conditions under which such a sequence satisfies A1 (and $z^*_n \to z^*$, w.p.1) can be found, e.g., in Attouch and Wets (1981), Dupačová and Wets (1988), and Rubinstein and Shapiro (1993). However, we note that the sequence of candidate solutions is an input to our sequential procedure and can be generated by any other method, e.g., by the stochastic decomposition algorithm of Higle and Sen (1996b). That said, we assume that the method that generates $\{\hat{x}_k\}$ does not depend on the sampled observations used in our evaluation procedures.

For any $x \in X$, let $\mu_x = Ef(x, \xi) - z^*$ and $\sigma^2(x) = \text{var}[f(x, \xi) - f(x^*_n, \xi)]$, where $x^*_n \in \arg \min_{x \in X} \text{var}[f(x, \xi) - f(y, \xi)]$. If (SP) has a unique optimum, i.e., $X^* = \{x^*\}$, then $x^*_n \to x^*$. When (SP) has multiple optimal solutions, $x^*_n$ depends on the given $x$.

Let $\xi^1, \xi^2, \ldots, \xi^n$ be a sample of size $n$. These observations could be i.i.d. from the same distribution as $\xi$ or could be drawn in some other way, e.g., to reduce variance. Suppose we have at hand an optimality gap estimator denoted $G_n(x)$ that uses this sample of size $n$ to estimate $\mu_x$, and similarly we have an estimator $\xi^*_n(x) \geq 0$ of the associated variance term $\sigma^2(x)$. As a concrete example of such estimators, let $\hat{\xi}^1, \hat{\xi}^2, \ldots, \hat{\xi}^n$ be i.i.d. from the distribution of $\xi$, let $\hat{x}^*_n$ solve $(SP_n)$ for this sample and form:

$$G_n(x) = \frac{1}{n} \sum_{i=1}^n (f(x, \hat{\xi}^i) - f(x^*_n, \hat{\xi}^i)), \quad (1a)$$

$$\hat{\xi}^*_n(x) = \frac{1}{n-1} \sum_{i=1}^n (f(x, \hat{\xi}^i) - f(x^*_n, \hat{\xi}^i)) - G_n(x)^2. \quad (1b)$$

The expression in (1a) motivates the use of the $n$ individual observations of the gap in the sample variance estimator (1b). We define

$$D_n(x) = \frac{1}{n} \sum_{i=1}^n [f(x, \hat{\xi}^i) - f(x^*_n, \hat{\xi}^i)], \quad (2)$$

where $x^*_n$ is defined as above for this $x$. The estimators $D_n(x)$, $G_n(x)$, and $\hat{\xi}^*_n(x)$ all use the same $n$ observations $\hat{\xi}^1, \hat{\xi}^2, \ldots, \hat{\xi}^n$. We make the following assumptions:

A2. Let $\{x_k\}$ be a feasible sequence (i.e., $x_k \in X$) with $x$ as one of its limit points. Let $n_k$ satisfy $n_k \to \infty$ as $k \to \infty$. Then, $\lim \inf_{k \to \infty} P(|G_{n_k}(x_k) - \mu_x| > \delta) = 0$ for any $\delta > 0$.

A3. $G_{n_k}(x) \geq D_{n_k}(x)$, w.p.1 for all $x \in X$ and $n_k \geq 1$.

A4. $\lim \inf_{n \to \infty} \hat{\xi}^*_n(x) \geq \sigma^2(x)$, w.p.1 for all $x \in X$.

A5. $\sqrt{n}(D_{n_k}(x_k) - \mu_x) \Rightarrow N(0, \sigma^2(x))$ as $n \to \infty$ for all $x \in X$, where $N(0, \sigma^2(x))$ is a normal random variable with mean zero and variance $\sigma^2(x)$. Here, $\Rightarrow$ denotes convergence in distribution.

The (SP) model encompasses a wide range of problems. For example, the model could involve discrete or continuous decision variables with the feasible region $X$ being finite, countable, or uncountably infinite. Moreover, we are primarily concerned with problems in which $\xi$ is a random vector of moderate-to-large dimension. However, our assumptions limit the scope of what we can address. As already indicated, we assume that the distribution of $\xi$ does not depend on the decision $x$. This assumption is typical in the stochastic programming literature, but is restrictive relative to the simulation-optimization literature. (SP) could represent a simulation-optimization problem where long-run average work in process in a queueing network is minimized by allocating constrained service rates to the network’s stations. Such a problem, with continuous decision variables, might be approached via stochastic approximation (SA); see, e.g., Pfug (1996, Ch. 5). If instead we allocate buffer capacity, or servers to each station, the decision variables would be discrete and one could approach the problem using random search (e.g., Andrásdóttir 2006), the stochastic nested partition method of Shi and Ölafsson.
(2000a, b), or the COMPASS method of Hong and Nelson (2006). SA requires a gradient estimator, and the latter methods require objective function estimates. Importantly, those estimates are formed for fixed x ∈ X, and this allows for the possibility that the distribution of ŝ depends on x. Such dependency naturally arises in models in which the

Assumption A3 is immediate because ŝ n(x) ≤ ŝ n(x min), w.p.1. Under i.i.d. sampling, D n(x) is a sample mean of i.i.d. observations, and hence if σ(x) < ∞, then A5 holds by the standard central limit theorem (CLT) for i.i.d. random variables. Note that if σ 2(x) = 0, then f(x, ŝ) − f(x min, ŝ) = µ, for almost all ŝ and A5 still holds, albeit in degenerate form. Sufficient conditions for A4 to hold under i.i.d. sampling are given in Bayraksan and Morton (2006). Note that when (SP) has multiple optimal solutions we cannot expect {x n} to have a single limit point, and hence we cannot expect s n(x) to converge as n → ∞. However, A4 is a form of consistency for s n(x) in the sense that it is bounded below by σ 2(x). (Recall, σ 2(x) is defined with respect to x min.) In general, a sufficient condition under which A2 holds is that ŝ n(x) converges uniformly to continuous Ef(x, ŝ) on X, w.p.1. This holds under i.i.d. sampling and compact X provided f(·, ŝ) is continuous on X, w.p.1, and E sup x∈X |f(x, ŝ)| < ∞; see, e.g., Shapiro (2003). We are now ready to present the sequential sampling procedure in more detail.

3. Sequential Sampling Procedure

At iteration k ≥ 1 of the sequential procedure we select a sample size, n k, and use n k total observations to assess the quality of the current solution, ŝ k. We can choose to generate ŝ 1, . . . , ŝ n k independently of those generated in previous iterations. Or, we can augment the observations from the previous iteration by generating n k − n k−1 additional observations, ŝ n k−1+1, . . . , ŝ n k (assume n 0 = 0). Intermediate possibilities between these two extremes are also permissible, and such choices can be made differently at each iteration. To assess ŝ k’s quality, we form an estimate of its optimality gap, G n k (ŝ k), and its variance, s 2 n k (ŝ k). If the current candidate solution does not satisfy a stopping criterion, we repeat the above steps with sample size n k+1 ≥ n k and the next candidate solution, ŝ k+1.

To simplify notation, from now on we suppress dependence on the candidate solution, ŝ k, and the sample size n k, and simply denote µ k = µ ŝ k, σ 2 k = σ 2 (ŝ k), D k = D n k (ŝ k), G k = G n k (ŝ k), and s k = s n k (ŝ k). We terminate the procedure at iteration k when the following stopping criterion is satisfied:

\[ T = \inf_{k \geq 1} \{ k \mid G_k \leq h's_k + \epsilon' \}. \tag{3} \]

The random stopping iteration T is designed to be the first iteration in which G k’s width relative to s k falls below h' > 0. Here, ϵ’ is a small positive number that ensures finite stopping, as we detail in the next section. By choosing the stopping criterion (3), we are willing to accept larger optimality gap estimates for problems with larger variability. Note that it is possible to add the condition \( s_k \leq b \) to (3) so that when we stop, G k is below a certain fixed threshold, h'b + ϵ'.
The stopping criterion (3) is with respect to \( h' \), and the statement regarding the quality of the candidate solution when we stop is with respect to a larger relative term \( h' + \epsilon \), where \( h > h' \) and \( \epsilon > \epsilon' \). Typically, we choose the epsilon terms so they are small compared to \( h' \).

Such inflation of the CI statement, relative to the stopping criterion, is fairly standard when using sampling methods with a sequential nature; see, for example, Chow and Robbins (1965) or Glynn and Whitt (1992). For the procedure we propose, we will show

\[
\lim_{h \to h'} P(\mu_T < h_T + \epsilon) \geq 1 - \alpha, \tag{4}
\]

where \( 1 - \alpha \) is the desired confidence level with \( 0 < \alpha < 1 \). In other words, (4) states that the optimality gap of \( \hat{x}_T \), the candidate solution when we stop, is a fraction of the sample standard deviation plus \( \epsilon \), with a desired probability, \( 1 - \alpha \), provided \( h \) is close enough to \( h' \). Our procedure’s sequential nature makes achieving this result a nontrivial task. Example 2 of Morton (1998) shows that when sample sizes grow to infinity as \( h \) does but do not grow with the iterate \( k \), then result (4) can fail to hold. The key is to have the sample sizes grow with the procedure’s iterates: see (5) below. As before, if we have the additional condition \( \{x_T \leq b\} \), then we asymptotically guarantee that the optimality gap of \( \hat{x}_T \) is at most \( hb + \epsilon \) with a confidence level of \( 1 - \alpha \).

At iteration \( k \), we choose the sample size according to

\[
n_k \geq \left( \frac{1}{h - h'} \right)^2 (c_p + 2p \ln^2 k), \tag{5}
\]

where \( c_p = \max\{2 \ln (\sum_{j=1}^{\infty} j^{-p} \ln j / \sqrt{2 \pi} \alpha), 1\} \). Here, \( p > 0 \) is a free parameter that affects the number of samples we generate. We discuss how to choose \( p \) in §5. The sample-size growth formula in (5) is proportional to \( (h - h')^{-2} \), it has a constant term, \( c_p \), that depends on \( p \) and \( \alpha \), and it grows as \( O(\log^2 k) \) in iterations. In the next section, we show that if the sample size satisfies (5), then (4) holds under a finite MGF assumption. We also show that the procedure stops in a finite number of iterations. Before proceeding, we summarize our procedure.

**Sequential Sampling Procedure:**

**Input:** Values for \( h > h' > 0 \), \( \epsilon > \epsilon' > 0 \), \( 0 < \alpha < 1 \), and \( p > 0 \). Method that generates candidate solutions \( \{x_k\} \) with at least one limit point in \( X^\ast \). Resampling frequency \( k_f \), a positive integer.

**Output:** Candidate solution, \( \hat{x}_f \), and a \((1 - \alpha)\)-level CI on its optimality gap, \( \mu_T \).

1. **(Initialization)** Set \( k = 1 \), calculate \( n_k \) as given in (5), and sample observations \( \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n_k} \).
2. If \( \{x_k \leq h_S + \epsilon\} \), then set \( T = k \), and go to 4.
3. Set \( k = k + 1 \) and calculate \( n_k \) according to (5). If \( k_f \) divides \( k \), then sample observations \( \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n_k} \) independently of samples generated in previous iterations. Else, sample \( n_k - n_{k-1} \) observations \( \tilde{x}_{n_{k-1}+1}, \tilde{x}_{n_{k-1}+2}, \ldots, \tilde{x}_{n_k} \) from the distribution of \( \tilde{x} \). Go to 1.

4. **Output candidate solution** \( \hat{x}_T \) and a one-sided CI on \( \mu_T \),

\[
[0, \ h_T + \epsilon]. \tag{6}
\]

If the resampling frequency \( k_f = 1 \), then at every iteration we sample observations \( \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n_k} \) independently of previously generated observations. At the other extreme, if \( k_f \) is sufficiently large, then we continually augment the existing set of observations throughout execution of the procedure. Frequent resampling decreases the likelihood that the procedure fails to terminate for a number of iterations because it is effectively stuck with a “bad” sample. Infrequent resampling increases the effectiveness of a computationally efficient warm start for solving the stochastic program required to form \( G_\epsilon \) and \( \hat{x}_T \) in step 1. We investigate the associated trade-off in §6.

### 4. Asymptotic Validity and Finite Stopping

The stopping iteration, \( T \), and the solution, \( \hat{x}_T \), provided by our sequential sampling method are random variables. Therefore, unlike a deterministic optimization algorithm, statements regarding finite stopping and the quality of the solution must be probabilistic. In §4.1 we first present a result that shows (4) holds under a finite MGF assumption. Then, we prove and discuss finite stopping of the algorithm. In §4.2 we relax the MGF assumption, replacing it with a finite \( r \)-th moment assumption.

#### 4.1. Finite Moment Generating Function

We establish the asymptotically-valid CI (4) under an assumption on the MGF of \( D_n(x) \), as defined in (2). We state the MGF assumption next, and discuss how the assumption can be shown to hold.

A6.

\[
\sup_{n \geq 1} \sup_{x \in X} E \exp \left[ \gamma \left( \frac{D_n(x) - \mu}{\sigma(x) \sqrt{n}} \right) \right] < \infty,
\]

for all \( |\gamma| \leq \gamma_0 \), for some \( \gamma_0 > 0 \).

When \( \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n \) are i.i.d., a sufficient condition to ensure that A6 holds is that the MGF of the scaled random variables \((f(x, \tilde{x}) - f(x_{\min}^\ast, \tilde{x})) / \sigma(x) \) exists, i.e.,

\[
\sup_{x \in X} E \exp \left[ \gamma \left( \frac{(f(x, \tilde{x}) - f(x_{\min}^\ast, \tilde{x})) - \mu}{\sigma(x) \sqrt{n}} \right) \right] < \infty
\]

for all \( |\gamma| \leq \gamma_0 \). \tag{7}

Condition (7) is satisfied when \( X \) is compact and \( \tilde{x} \) has bounded support. For \( \tilde{x} \) with possibly unbounded support, but \( X \) compact, condition (7), and hence A6, holds when (SP) satisfies the following Lipschitz condition

\[
|f(x_1, \tilde{x}) - f(x_2, \tilde{x})| \leq K(\tilde{x}) \|x_1 - x_2\| \ \text{w.p.1} \tag{8}
\]
for all \( x_1, x_2 \in X \), where the Lipschitz constant \( K(\tilde{x}) \) satisfies \( E \exp[\gamma K(\tilde{x})] < \infty \) for \( |\gamma| \leq \gamma_0 \). How can this Lipschitz condition be verified? We answer this for a special class of (SP), namely, for two-stage stochastic linear programs with fixed recourse in which

\[
f(x, \tilde{x}) = cx + \min_{y \geq 0} qy
es.t. \quad W y = \tilde{r} - \tilde{T} x.
\]

Suppose this linear program is dual feasible for almost all \( \tilde{q} \) and the stochastic program has relative complete recourse. Further, suppose \( \tilde{q}, \tilde{r}, \) and \( \tilde{T} \) can each be expressed as a linear combination of the underlying random vector \( \tilde{x} \), which has independent components. Note that this allows for first-order dependencies between the components of \( (\tilde{q}, \tilde{r}, \tilde{T}) \). Then, the Lipschitz condition (8), and hence condition (7), will be satisfied when the squared Euclidean norm of the random vector \( \tilde{x} \) has an MGF, i.e., \( E \exp[|\gamma| \tilde{x}] < \infty \) for \( |\gamma| \leq \gamma_0 \); see Römisch (2003, Proposition 22). This provides a natural sufficient condition under which we may be assured that the MGF Assumption A6 holds. Finally, we note that at an \( x \in X \) at which \( \sigma^2(x) = 0 \), \( D_t(x) \) is constant almost everywhere for all \( n \), and hence has an MGF. As will become clear in the proof of Theorem 1 below, this type of degeneracy does not lead to an effective violation of A6.

Below we state and prove the validity of the sequential sampling procedure under hypothesis A6. Our result, given in (4), is asymptotic as \( h \downarrow h' \), i.e., as the sample sizes grow. We note that even in the simple case of constructing a CI for a population mean under sequential sampling the validity of the resulting CI, i.e., that the CI has the desired coverage probability, is proven asymptotically (Chow and Robbins 1965). To prove (4), we make use of: (i) Fatou’s Lemma, which provides inequalities when “lim inf” and an integral (or an infinite sum) are exchanged and (ii) a bound on the tail of a normal random variable. These are given in the next two lemmas. For proofs of all lemmas that appear in the paper, please see the online appendix. An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

**Lemma 1 (Fatou’s Lemma).** Suppose \( \{f_n\} \) is a sequence of measurable functions on \( E \).

(i) If \( f_n \geq 0 \) for all \( n \), then

\[
\int_E \lim_{n \to \infty} f_n \leq \liminf_{n \to \infty} \int_E f_n.
\]

(ii) If \( L \leq f_n \leq U \) for all \( n \), such that \( \int_E L < \infty \) and \( \int_E U < \infty \), then,

\[
\int_E \lim_{n \to \infty} f_n \leq \liminf_{n \to \infty} \int_E f_n \leq \int_E \limsup_{n \to \infty} f_n.
\]

**Lemma 2 (Bound on Tail of a Standard Normal).** Let \( Z \) be a standard normal and \( t > 0 \). Then,

\[
P(Z \geq t) \leq \frac{1}{\sqrt{2\pi}} \exp[-t^2/2].
\]

The following theorem establishes asymptotic validity of the sequential procedure under A6.

**Theorem 1.** Assume A3–A6 are satisfied. Let \( \varepsilon > \varepsilon' > 0 \), \( p > 0 \) and \( 0 < \alpha < 1 \) be fixed. Consider the sequential sampling procedure where the sample size is increased according to (5). If the procedure stops at iteration \( T \) according to (3), then,

\[
\liminf_{h \downarrow h'} P(\mu_T \leq \hat{h}_T + \varepsilon) \geq 1 - \alpha.
\]

**Proof.** Let \( \Delta h = h - h' \), \( \Delta \varepsilon = \varepsilon - \varepsilon' \), and \( M^2 = \sup_{x \in X} \sigma^2(x) \). We have

\[
P(\mu_T > h_T + \varepsilon)
\leq P(\mu_T > G_T + h_T - \Delta h + \Delta \varepsilon)
\leq \sum_{k=1}^{\infty} P(G_k > h'k_1 + \varepsilon', \ldots, G_{k-1} > h'k_1 + \varepsilon', G_k \leq h'k_1 + \varepsilon, G_k - \mu_k \leq -\Delta h_k - \varepsilon_k)
\leq \sum_{k=1}^{\infty} P(D_k - \mu_k \leq -\Delta h_k - \varepsilon_k),
\]

where the first inequality follows from (3) and inequality (10) follows from A3. So, it suffices to show

\[
\limsup_{\Delta h \downarrow 0} \sum_{k=1}^{\infty} P(D_k - \mu_k \leq -\Delta h_k - \varepsilon_k) \leq \alpha.
\]

To apply part (ii) of Fatou’s lemma, we first show that the right-hand side of (10) is bounded above for all positive \( \Delta h \) that are sufficiently small:

\[
\sum_{k=1}^{\infty} P(D_k - \mu_k \leq -\Delta h_k - \varepsilon_k)
\leq \sum_{k=1}^{\infty} \int_{\hat{x}_k}^{\tilde{x}_k} E \left[ \exp \left[ -\gamma_0 \left( \frac{D_k - \mu_k}{\sigma_k} \right) \right] \right] dP_{\hat{x}_k}
\leq \sum_{k=1}^{\infty} \int_{\hat{x}_k}^{\tilde{x}_k} E \left[ \exp \left[ -\gamma_0 \left( \frac{\Delta h_k}{\sqrt{n_k}} \right) \right] \right] dP_{\hat{x}_k}
\leq \sup_{k \geq 1} \sup_{x \in X} \int_{\hat{x}_k}^{\tilde{x}_k} \left[ \exp \left[ -\gamma_0 \left( \frac{D_k - \mu_k}{\sigma_k} \right) \right] \right] \sum_{k=1}^{\infty} k^{-\gamma_0 \left( \frac{\sigma_k}{\sqrt{n_k}} \right)}.
\]
Note that for \( \hat{x} \) with \( \sigma_x^2 = 0 \), the probability in (11) is 0. The multiplier of the infinite sum in (13) is bounded by A6 whereas the sum itself is bounded for all sufficiently small \( \Delta h \), more specifically, for all \( 0 < \Delta h < \gamma_0 \sqrt{2p} \Delta \epsilon / M \). Taking limits we obtain,

\[
\limsup_{\Delta h \downarrow 0} \sum_{k=1}^{\infty} P(D_k - \mu_k \leq -\Delta h s_k - \Delta \epsilon) \\
\leq \sum_{k=1}^{\infty} \limsup_{\Delta h \downarrow 0} \int_{\hat{x}_k} \frac{D_k - \mu_k}{\sigma_k / \sqrt{m_k}} \leq -\Delta h \sqrt{\frac{m_k}{\sigma_k}} \left( \frac{s_k}{\sigma_k} \right) dP_{\hat{x}_k} \\
\leq \sum_{k=1}^{\infty} \limsup_{\Delta h \downarrow 0} \int_{\hat{x}_k} \frac{D_k - \mu_k}{\sigma_k / \sqrt{m_k}} \leq -(c_p + 2p \ln^2 k)^{1/2} \left( \frac{s_k}{\sigma_k} \right) dP_{\hat{x}_k} \\
\leq \alpha,
\]

where the first and the third inequalities follow from an application of Fatou’s lemma. With \( k \) and \( \hat{x}_k \) fixed, \( (D_k - \mu_k)/(\sigma_k / \sqrt{m_k}) \) converges to a standard normal by A5 because \( \Delta h \downarrow 0 \) ensures \( n_k \to \infty \). Similarly, \( \liminf_{\Delta h \downarrow 0} (\sigma_k / \sqrt{m_k}) \geq 1 \) by A4. The last inequality then follows by applying Lemma 2 and the definition of \( c_p \).

Theorem 1 shows that for values of \( h \) close enough to \( h' \), or when the sample sizes \( n_k \) are large enough, we have the optimality gap of the solution when we stop in \([0, hs_T + \epsilon]\) with at least the desired probability of \( 1 - \alpha \). We now turn our attention to finite stopping and show that the sequential procedure stops with probability one. We state this formally in the proposition below.

**Proposition 1.** Assume A1 and A2 are satisfied. Let \( \epsilon' > 0 \) and \( \Delta h > h' > 0 \) be fixed. Then, for the sequential sampling procedure where the sample size is increased according to (5), and the procedure stops at iteration \( T \) according to (3), we have \( P(T < \infty) = 1 \).

**Proof.** Note that

\[
P(T = \infty) \leq \liminf_{k \to \infty} P(G_k > h' s_k + \epsilon') \\
\leq \liminf_{k \to \infty} P(G_k > \epsilon') \\
\leq \liminf_{k \to \infty} P(|G_k - \mu_k| > \epsilon' - \mu_k),
\]

and the final term is zero by A1 and A2.

### 4.2. Weaker Moment Conditions

In this section, we prove a variant of Theorem 1 that assumes finite moments up to order \( r \). Specifically, we relax the MGF assumption of the previous section to

\[ A7. \sup_{x \in X} E[|f(x, \tilde{\xi})|^r] < \infty, \] for some even integer \( r \geq 2 \).

With this relaxation, the sequential procedure requires a larger number of observations (see (14) below). Note that A7 implies \( \sup_{x \in X} E[|f(x, \tilde{\xi}) - f(y, \tilde{\xi})|^r] < \infty \). For reasons discussed in detail below, to obtain the desired property of asymptotic validity in this case, we confine our analysis to i.i.d. sampling. Under these assumptions, we select the sample size at each iteration \( k \) according to

\[
n_k \geq \left( \frac{1}{h - h'} \right)^2 (c_{p, q} + 2p k^{2q/r}),
\]

where \( q > 1, p > 0 \) and where \( c_{p, q} = \max \{2 \ln \sum_{i=1}^{\infty} \exp[-p j^{2q/r}]/\sqrt{2\pi} \alpha, 1 \} \). The growth in the sample size is of order \( O(k^{2q/r}) \) and we must choose \( q > 1 \). Therefore, if A7 holds for \( r = 2 \), we can choose \( q \) just larger than unity so that the sample size essentially grows at a linear rate, and if A7 holds for \( r = 4 \), we can obtain a rate that essentially grows with \( k^{1/2} \). In other words, a less restrictive assumption on the existence of moments implies a faster rate of growth for the sample sizes.

Under the MGF hypothesis, the sample-size formula (5) contains a parameter, \( p > 0 \), at our disposal. Now, under the moment hypothesis A7, if we elect the slowest possible rate of growth of \( n_k \) by choosing \( q \) just larger than unity, then we can again view the sample-size formula as being parameterized by a free scalar term, \( p > 0 \).

To prove the validity and finite stopping of our procedure under the finite \( r \)th moment Assumption A7, we need a lemma that establishes a bound on the central moments of a parameterized by a free scalar term, \( p > 0 \).

**Lemma 3.** There are no more than \( \lfloor \sqrt{rn} \rfloor \) ways to place \( r \) distinguishable balls in \( n \) distinguishable bins so that no bin contains exactly one ball.

**Lemma 4.** Let \( X_1, X_2, \ldots, X_n \) be i.i.d. random variables with mean \( \mu \) and \( \tilde{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \). If \( E|X_1 - \mu|^r < \infty \) for some integer \( r \geq 1 \), then

\[
E(\tilde{X}_n - \mu)^r \leq E|X_1 - \mu|^r \left( \frac{r}{n} \right)^{r/2}.
\]

We are now ready to prove the validity and finite stopping of our procedure under the finite \( r \)th moment Assumption A7.

**Theorem 2.** Assume that A1–A4 are satisfied, that A7 holds with \( r \geq 2 \) even, and that \( \tilde{\xi}_1, \tilde{\xi}_2, \ldots \) are i.i.d. as \( \tilde{\xi} \). Let \( \epsilon > \epsilon' > 0, p > 0, q > 1, \) and \( 0 < \alpha < 1 \) be fixed. Then, for the sequential sampling procedure where the sample size is increased according to (14), and the procedure stops at iteration \( T \) according to (3),

\[
P(T < \infty) = 1 \quad \text{and} \quad \liminf_{h \downarrow h'} P(\mu_T \leq hs_T + \epsilon) \geq 1 - \alpha.
\]
The proof of the finite stopping result is identical to that of Proposition 1. To prove asymptotic validity, we start as in the proof of Theorem 1 and proceed until (11) the same way. Then, instead of using a Chernoff bound, which is Markov’s inequality applied to the exponent of a random variable, we apply Markov’s inequality with the rth moment under Assumption A7, where r is even

\[ \sum_{k=1}^{\infty} \int_{\xi_k}^{\infty} P(D_k - \mu_k \leq -\Delta \varepsilon | \hat{\xi}_k) dP_{\xi_k} \]

\[ \leq \sum_{k=1}^{\infty} \int_{\xi_k}^{\infty} P(|D_k - \mu_k| \geq \Delta \varepsilon | \hat{\xi}_k) dP_{\xi_k} \]

\[ \leq \sum_{k=1}^{\infty} \int_{\xi_k}^{\infty} E[(D_k - \mu_k)^r | \hat{\xi}_k] \Delta \varepsilon^{-r} dP_{\xi_k} \]

\[ \leq \sup_{x \in X} E\left[\|f(x, \hat{\xi}) - f(x^{*, \xi}, \hat{\xi}) - \mu_i\| \right] r^{r/2} \Delta \varepsilon^{-r} \sum_{k=1}^{\infty} \frac{1}{n_k^{r/2}}, \quad (15) \]

where (15) follows from Lemma 4. From the definition of \( n_k \) given in (14), the right-hand side of (15) is bounded. The hypothesis that \( \hat{\xi}_1, \hat{\xi}_2, \ldots \) are i.i.d. along with Assumption A7 holding for \( r \geq 2 \) implies that Assumption A5 holds. As a result, the rest of the proof is analogous to that of Theorem 1. \( \square \)

Theorem 2 differs from Theorem 1 in two respects. First, the MGF Assumption A6 is replaced by the weaker moment condition A7. Under this weaker assumption, the sample sizes are chosen according to (14) instead of (5), requiring more observations. Second, Theorem 2 is only shown to hold under i.i.d. sampling rather than the weaker assumption of asymptotic normality in A5. This is because Theorem 2 hinges on Lemma 4, which we only establish for i.i.d. random variables.

So far, we have verified desirable theoretical properties of our sequential procedure. In the next section, we discuss issues that arise when implementing the procedure. In particular, we discuss how to choose \( p \) when using the sample-size formula (5) and how to choose \( q \) and \( p \) when using (14). Then, we apply the procedure to five test problems that are two-stage stochastic linear programs with recourse to examine its performance.

### 5. Choosing the Parameters of the \( n_k \) Formulae

At iteration \( k \) of the sequential procedure, we choose the sample size according to (5) and (14) under A6 and A7, respectively. Using these many samples, we solve a sampling problem (SP\(_{n_k}\)) to estimate the optimality gap of the current candidate solution and its associated variance. Suppose the procedure terminates in \( T \) iterations, having solved \( T \) sampling problems, (SP\(_{n_1}\), SP\(_{n_2}\), \ldots, SP\(_{n_T}\)). The computational effort exerted for the evaluation of the candidate solutions is then approximately proportional to \( \sum_{k=1}^{T} n_k \).

(For some decomposition methods, empirical studies suggest that the effort to solve a stochastic program grows linearly in the number of scenarios; see, e.g., Ruszczyński and Swietanowski 1997 and Verweij et al. 2003.) Therefore, the effort is proportional to

\[ S_M(p) = T \max \left\{ 2 \ln \left( \frac{\sum_{j=1}^{\infty} f_j \ln j}{\sqrt{2\pi \alpha}} \right), 1 \right\} \]

\[ + 2p \sum_{k=1}^{T} \ln^2 k \]

\[ S_W(p, q) = T \max \left\{ 2 \ln \left( \frac{\sum_{j=1}^{\infty} \exp[-p j^{2q/r}]}{\sqrt{2\pi \alpha}} \right), 1 \right\} \]

\[ + 2p \sum_{k=1}^{T} k^{2q/r}, \quad (16a) \]

under Assumptions A6 and A7, respectively, for fixed \( h - h' \). Assume, for the moment, that \( T \) is known. The parameter \( p > 0 \) for \( S_M(p) \) and the parameters \( p > 0 \) and \( q > 1 \) for \( S_W(p, q) \) are at our disposal. To reduce computational effort, we would like to choose them to minimize \( S_M(p) \) and \( S_W(p, q) \). The following result helps to do so.

**Proposition 2.** Let \( S_M(p) \) and \( S_W(p, q) \) be defined in (16). \( S_M(\cdot) \) is convex on \( \{ p : p > 0 \} \) and \( S_W(\cdot, q) \) is convex on \( \{ p : p > 0 \} \) for fixed \( q > 1 \). Furthermore, \( S_M(p) \) and \( S_W(p, q) \) are both bounded below by

\[ 2T \ln \left( \frac{T}{\sqrt{2\pi \alpha}} \right). \]

(17)

The proof of Proposition 2 is provided in the online appendix. When employing the sample-size formula (5), the associated function \( S_M(p) \) is convex, and its minimizers \( p^* \) and minimal function values \( S_M(p^*) \) for various values of \( T \) are shown in the second and third columns of Table 1. When we use sample-size formula (14) we seek to select \( p \) and \( q \) to solve \( \min_{p>0,q>1} S_W(p, q) \). By Proposition 2, \( S_W(\cdot, q) \) is convex, and so we can minimize this function for a fixed value of \( q \). The fourth and fifth columns of Table 1 show the results of doing so for \( q = 1.5 \) and \( r = 2 \). We know by the lower-bounding values \( LB = 2T \ln(T/(\sqrt{2\pi \alpha})) \) shown in the final column of the table that these are suboptimal by no more than 2.5–5.5%.

Table 1’s values for \( S_M(\cdot) \) and \( S_W(\cdot, q) \) are both bounded below by

<table>
<thead>
<tr>
<th>( T )</th>
<th>( p^* )</th>
<th>( S_M(p^*) )</th>
<th>( p^* )</th>
<th>( S_W(p^*, 1.5) )</th>
<th>( LB )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.07 × 10^{-1}</td>
<td>82</td>
<td>5.05 × 10^{-2}</td>
<td>78</td>
<td>74</td>
</tr>
<tr>
<td>50</td>
<td>1.91 × 10^{-1}</td>
<td>591</td>
<td>4.67 × 10^{-4}</td>
<td>552</td>
<td>530</td>
</tr>
<tr>
<td>100</td>
<td>1.53 × 10^{-1}</td>
<td>1,334</td>
<td>1.66 × 10^{-4}</td>
<td>1,243</td>
<td>1,198</td>
</tr>
<tr>
<td>500</td>
<td>1.04 × 10^{-1}</td>
<td>8,421</td>
<td>1.49 × 10^{-4}</td>
<td>7,822</td>
<td>7,598</td>
</tr>
<tr>
<td>1,000</td>
<td>9.08 × 10^{-2}</td>
<td>18,333</td>
<td>5.27 × 10^{-5}</td>
<td>17,031</td>
<td>16,583</td>
</tr>
</tbody>
</table>
\[ \alpha = 0.10, \text{ but the same } p^* \text{ values are optimal, e.g., for } \alpha = 0.05. \]

The results of Table 1 can guide selection of \( p \) given rough estimates for \( T \). Of course, the assumption that \( T \) is known is unrealistic. Although we could view \( T \) as a random variable and attempt to minimize \( S_M \) or \( S_W \) in expectation, we will not do so. When the procedure terminates at a different iteration than that of the assumed \( T \), the differences in sample sizes are quite modest. For instance, taking \( h - h' = 0.5 \) and \( \alpha = 0.10 \), and using \( p = 1.91 \times 10^{-1} \) (for \( T = 50 \)), we have \( n_T \geq 33, 56 \), and 65 samples when \( T = 1, 50, \) and 100, respectively, under the MGF assumption of A6. Similarly, we have \( n_T \geq 37, 55, \) and 63 when we instead use \( p = 1.53 \times 10^{-1} \) (for \( T = 100 \)). For the weaker moment condition with \( r = 2 \) and \( q = 1.5, n_T \geq 39, 52, \) and 77 when \( T = 1, 50, \) and 100, respectively, when we use \( p = 4.67 \times 10^{-3} \) and \( n_T \geq 45, 50, \) and 58 samples when we instead use \( p = 1.66 \times 10^{-3} \). Slightly smaller values of \( S_W(p, q) \) can be obtained with even larger values of \( q \), but the results are more sensitive to having assumed the “wrong” value of \( T \).

### 6. Application to Two-Stage Stochastic Linear Programs

We use five test problems from the literature, denoted CEP1, 4TERM, PGP2, APL1P, and GBD. Our first test problem, CEP1, is a capacity-expansion planning problem in a production facility with flexible machines facing uncertain demand (Higle and Sen 1996b). It has 3 independent stochastic parameters with 6 realizations each, resulting in 36 scenarios, whereas 4TERM has 1 depot and 4 terminal cities, with 8 stochastic parameters and 256 scenarios. We use 4TERM instead of the larger 20TERM because we can calculate all the desired performance measures, allowing us to test our procedures. PGP2 is an electric power generation model with 3 stochastic parameters and 576 scenarios (Higle and Sen 1996b). APL1P, another power generation model, has 5 independent stochastic parameters and 1,280 scenarios (Infanger 1992). GBD is an aircraft allocation problem originally described in Ferguson and Dantzig (1956). The original model has 750 scenarios and the model that we use has 646,425 scenarios, obtained by refining the original demand distributions. Table 2 lists summary characteristics of our test problems. The table’s second and third columns give the number of decision variables for a single-scenario problem, the fourth column the number of stochastic parameters, the fifth column the number of scenarios, and the final column the instance’s optimal value. All five test problems can be solved exactly, and hence allow us to assess the performance of our sequential sampling procedures.

#### 6.2. Gap and Variance Estimators

We use two different methods to form the gap and variance estimators to assess the quality of candidate solutions. These are the single replication procedure (SRP) and the averaged two-replication procedure (A2RP). In Bayraksan and Morton (2006), we introduced SRP and A2RP and focused on nonsequential estimation involving a single candidate solution, \( \hat{x} \). Below, we examine their performance within a sequential sampling procedure. We now briefly explain SRP and A2RP. For the SRP estimators, step 1 of the sequential procedure presented at the end of \$3 \$ computes gap and variance estimators according to the equations in (1), i.e., step 1 becomes

1.a. Solve \((SP_{n_k}) \) using \( \hat{\xi}_1, \hat{\xi}_2, \ldots, \hat{\xi}_{n_k} \) i.i.d. from \( \tilde{\xi} \)’s distribution to obtain \( x_{n_k}^* \).

1.b. Calculate \( G_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (f(\hat{x}_k, \tilde{\xi}) - f(x_{n_k}^*, \tilde{\xi})) \), and

\[
\hat{s}_k = \frac{1}{n_k - 1} \sum_{i=1}^{n_k} [(f(\tilde{x}_k, \tilde{\xi}) - f(x_{n_k}^*, \tilde{\xi})) - (\hat{f}_{n_k}(\hat{x}_k) - \hat{f}_{n_k}(x_{n_k}^*))^2].
\]

The implementation of the sequential procedure with A2RP involves small changes. We select \( n_k \) even and divide the observations into two random partitions and calculate \( G_{k,i} \).
and \( s_{k,i}^2 \) as above (steps 1.a and 1.b) with each sample containing \( n_i/2 \) observations, \( i = 1, 2 \). We then pool these gap and variance estimators to obtain \( G_k = \frac{1}{2}(G_{k,1} + G_{k,2}) \) and \( s_k^2 = \frac{1}{2}(s_{k,1}^2 + s_{k,2}^2) \). (Note that \( \frac{1}{2} \) is the correct factor for the pooled variance because \( s_{k,1}^2 \) and \( s_{k,2}^2 \) each estimate the variance of \( f(\tilde{x}_i, \tilde{\xi}) \) not \( \text{var} G_{k,1} \) and \( \text{var} G_{k,2} \), in which case the correct factor would have been \( \frac{1}{2} \)). The gap estimators, \( G_k \), are identical under SRP and A2RP, but the variance estimators, \( s_k^2 \), differ. For the sequential procedure that uses A2RP, we use these pooled estimators for the stopping criterion in step 3. Note that the gap and variance estimators formed by SRP and A2RP under i.i.d. sampling satisfy Assumptions A2–A5 stated in §2 (Bayraksan and Morton 2006).

6.3. Generating Candidate Solutions

The method to generate candidate solutions is an input to our procedure. Any method suffices, provided its sequence of candidate solutions, \( \{\tilde{x}_i\} \), has at least one optimal limit point (A1). In our computational results, we generate the candidate sequence by solving a separate sampling problem (SP) with increasing sample sizes \( m_k \) at iteration \( k \), as described below:

i. Set \( m_k = m_1 \). Sample i.i.d. observations (independent of those used in the evaluation procedures) \( \tilde{\xi}^1, \tilde{\xi}^2, \ldots, \tilde{\xi}^{m_1} \) from the distribution of \( \tilde{\xi} \);

ii. Solve (SP) using observations generated so far to obtain \( \tilde{x}_1^* \);

iii. Set \( \tilde{x}_k = \tilde{x}_1^* \). Calculate \( m_k+1 - m_k \) i.i.d. observations \( \tilde{\xi}^{m_k+1}, \tilde{\xi}^{m_k+2}, \ldots, \tilde{\xi}^{m_k+1} \) from the distribution of \( \tilde{\xi} \). Set \( k = k + 1 \) and go to ii.

In step i above, we use a separate stream of i.i.d. observations from the distribution of \( \tilde{\xi} \) independent from the ones generated in steps 0 and 2 of the sequential procedure. This ensures Assumption A3 from §2 holds. For our computational experiments, we set \( m_k = 2n_k \). We exert more computational effort to find high-quality candidate solutions, and we use smaller sample sizes to evaluate them. With this choice, a sequential procedure that stops at iteration \( T \) uses a total of \( 3(n_1 + n_2 + \cdots + n_T) \) observations. Finally, we note that all limit points of this sequence of candidate solutions are in the set of optimal solutions, \( X^* \), under mild conditions satisfied by our test problems; see, e.g., Rubinstein and Shapiro (1993). Therefore, Assumption A1 is satisfied.

6.4. Parameters Used

All of the test problems satisfy assumptions stated in §2: \( X \neq \emptyset \) and is compact, \( f(\cdot, \tilde{\xi}) \) is continuous on \( X, \) w.p.l., etc. Moreover, the random vector \( \tilde{\xi} \) for each of these test problems has a discrete distribution with independent components and bounded support. Therefore, the MGF Assumption A6 is satisfied for all \( \gamma_o \), and hence we use the sample-size formula (5). We set \( \alpha = 0.10 \) and design the procedure for \( T = 50 \). Minimizing computational effort, we set \( p = 1.91 \times 10^{-1} \) (see §5). Once the values of \( \alpha \) and \( p \), and therefore \( c_p \), are set, \( \Delta h = h - h' \) determines the initial sample size. For instance, with the above values of \( \alpha \) and \( p \), \( \Delta h = 0.287 \) corresponds to \( n_1 \geq 98.90 \) by (5). Rounding this up to an even integer, because we will be testing two-replication procedures, yields \( n_1 = 100 \) for this value of \( \Delta h \). At each iteration \( k \), we similarly round up the \( n_1 \) from (5) and use the same stream of random numbers for both SRP and A2RP.

Table 3 lists different values of \( \Delta h \) and the corresponding initial sample sizes we use. As \( \Delta h \) shrinks, the sample size grows. To examine the effect of the initial sample size, we use three values of \((h, h')\) for GBD that result in initial sample sizes of \( n_1 = 100, 300, \) and \( 500 \), and denote these as GBD:100, GBD:300, and GBD:500. Selection of an initial sample size dictates a value of \( \Delta h \) and can be chosen according to our computational ability to solve that problem. Specific values of \( h \) and \( h' \) with \( \Delta h = h - h' \) can then be determined as follows.

A preliminary computational study, with a modest sample size, \( n_1 \), can be performed to gain an understanding of \( G/s \), the main driver of the stopping rule. If, on average, \( G_n/s_n \) is small, then \( h' \) can be picked small. If, however, the average value of \( G_n/s_n \) is large, then \( h' \) cannot be chosen too small, otherwise, the procedure may require an excessive number of iterations. For instance, our preliminary computations indicate \( G/s \) is approximately three times smaller for GBD than for PGP2. Therefore, we pick \( h = 0.015 \) for GBD and pick \( h' = 0.045 \) for PGP2. Finally, \( h \) can be obtained from the resulting \( \Delta h \) value for a desired initial sample size.

When implementing the procedure, we set \( \epsilon = 2 \times 10^{-7} \) and \( \epsilon' = 1 \times 10^{-7} \). Here, \( \epsilon' \), in addition to ensuring finite stopping, serves to deal with nonzero numerical tolerances. For instance, suppose we are using a solver with a tolerance of \( 1 \times 10^{-8} \) and at an iteration \( k \), we calculate \( G_k = 1 \times 10^{-8} \) and \( s_k = 1 \times 10^{-12} \), which we can essentially treat as 0. However, if we have \( \epsilon' = 0 \) in (3), we would not stop for \( h < 10,000 \). Note that \( \epsilon' = 1 \times 10^{-7} \) used in the stopping criterion (3) is small enough not to interfere with stopping of the sequential procedure when \( G_k \) and \( s_k^2 \) are sufficiently large.

In summary, in our computational tests, the parameters \( \alpha = 0.10, p = 1.91 \times 10^{-1}, \epsilon = 2 \times 10^{-7}, \) and \( \epsilon' = 1 \times 10^{-7} \) are the same for all test problems, and the values of \( h \) and \( h' \) differ as shown in Table 3. In the next section, we
study the effect of resampling to help guide selection of parameter \(k_f\), the resampling frequency.

### 6.5. Computational Results

The sequential sampling method depends on our ability to efficiently solve a sequence of problem instances \((\text{SP}_{n_k})\) as \(n_k\) grows. In our computation, we use the regularized decomposition algorithm of Ruszczyński (1986). An accelerated implementation of this algorithm, in C++, is due to Ruszczyński and Świetanowski (1997). Our problem instances are two-stage stochastic linear programs, and the regularized decomposition method is a cutting-plane algorithm whose iteratively updated collection of cuts forms a piecewise-linear approximation of each \(f(x, \tilde{E})\), \(i = 1, \ldots, n_k\), in the sample-mean objective function \(f_{n_k}(x) = \frac{1}{n_k} \sum_{i=1}^{n_k} f(x, \tilde{E}_i)\). After solving \((\text{SP}_{n_k})\), we can then augment the sample with a small number, \(n_{k+1} - n_k\), of additional observations to form the new, but similar, objective function \(f_{n_{k+1}}(x)\). Importantly, when augmenting the sample in this way, the previously formed piecewise-linear approximations of \(f(x, \tilde{E})\), \(i = 1, \ldots, n_k\), remain valid. So, we modified the code of Ruszczyński and Świetanowski to warm-start the algorithm, retaining cuts generated when solving \((\text{SP}_{n_k})\), when we then solve \((\text{SP}_{n_{k+1}})\). This dramatically reduces the computational effort required to find a solution to this new sampling problem.

The above warm-start method extends to convex stochastic nonlinear programs, but a different approach would be required in a two-stage stochastic integer program with discrete decision variables in the second-stage problem. More generally, the ability to efficiently solve \((\text{SP}_{n_{k+1}})\), formed by adding a few samples to a recently solved problem instance \((\text{SP}_{n_k})\), will significantly improve the computational efficiency of our procedure. However, the means by which one effects such a warm-start procedure will depend on special structures of the problem class under consideration.

Even though adding additional samples to the existing problem can speed solution times for that new problem, it does not mean it is always desirable to do so. As mentioned before, our method allows for augmenting the current set of samples or generating an entirely new set. Augmenting is computationally attractive in the short run, but we may get trapped, for an extended number of iterations, in a “bad” sample path. Resampling helps the procedure move to a different sample path, but we must typically solve the new problem instance “from scratch.” In what follows, we first examine this trade-off between augmentation and resampling by changing the resampling frequency, \(k_f\). Then, we examine the performance of the sequential procedure that uses SRP and A2RP.

#### 6.5.1. Augmentation vs. Resampling.

To examine the effect of augmentation versus resampling, we varied the resampling frequency, \(k_f\), from 100 down to 1 using the SRP estimators on all test problems. Table 4 shows the results of our tests for GBD with three initial sample sizes, and Figure 1 shows results for all our test problems.

Each row of Table 4 is obtained with 300 independent runs of our procedure. We report the average solution time over all 300 runs (in seconds), which includes time for generating the sequence of candidate solutions and time.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution time</th>
<th>Assess time</th>
<th>(SP)</th>
<th>Assess time/ (SP)</th>
<th>90% CI on T</th>
<th>90% CI on p</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBD:100</td>
<td>100</td>
<td>2.66</td>
<td>0.84</td>
<td>31.98</td>
<td>0.026</td>
<td>102.82 ± 18.84</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>2.39</td>
<td>0.76</td>
<td>29.57</td>
<td>0.026</td>
<td>79.71 ± 14.32</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>2.05</td>
<td>0.64</td>
<td>25.98</td>
<td>0.025</td>
<td>53.09 ± 7.06</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1.72</td>
<td>0.57</td>
<td>22.54</td>
<td>0.025</td>
<td>38.85 ± 5.26</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1.41</td>
<td>0.50</td>
<td>18.42</td>
<td>0.027</td>
<td>27.29 ± 4.45</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.09</td>
<td>0.41</td>
<td>14.21</td>
<td>0.029</td>
<td>16.86 ± 1.98</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.93</td>
<td>0.38</td>
<td>11.59</td>
<td>0.033</td>
<td>12.20 ± 1.26</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.88</td>
<td>0.38</td>
<td>9.04</td>
<td>0.051</td>
<td>9.04 ± 1.08</td>
</tr>
<tr>
<td>GBD:300</td>
<td>100</td>
<td>4.60</td>
<td>1.36</td>
<td>20.88</td>
<td>0.065</td>
<td>26.92 ± 7.05</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>4.19</td>
<td>1.23</td>
<td>19.13</td>
<td>0.064</td>
<td>23.80 ± 6.64</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>4.18</td>
<td>1.23</td>
<td>18.45</td>
<td>0.067</td>
<td>22.95 ± 5.78</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>2.91</td>
<td>0.88</td>
<td>13.24</td>
<td>0.066</td>
<td>13.94 ± 2.40</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1.99</td>
<td>0.63</td>
<td>8.96</td>
<td>0.070</td>
<td>9.00 ± 1.25</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.31</td>
<td>0.42</td>
<td>5.58</td>
<td>0.075</td>
<td>5.58 ± 0.59</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.03</td>
<td>0.36</td>
<td>3.97</td>
<td>0.091</td>
<td>3.97 ± 0.39</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.85</td>
<td>0.35</td>
<td>2.93</td>
<td>0.120</td>
<td>2.93 ± 0.24</td>
</tr>
<tr>
<td>GBD:500</td>
<td>100</td>
<td>4.64</td>
<td>1.38</td>
<td>11.03</td>
<td>0.125</td>
<td>12.44 ± 4.53</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>3.73</td>
<td>1.10</td>
<td>9.75</td>
<td>0.113</td>
<td>10.37 ± 3.22</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3.98</td>
<td>1.20</td>
<td>10.24</td>
<td>0.117</td>
<td>11.86 ± 4.89</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>2.28</td>
<td>0.69</td>
<td>5.91</td>
<td>0.117</td>
<td>6.00 ± 1.48</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1.72</td>
<td>0.55</td>
<td>4.48</td>
<td>0.124</td>
<td>4.48 ± 0.69</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.42</td>
<td>0.48</td>
<td>3.45</td>
<td>0.139</td>
<td>3.45 ± 0.59</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.21</td>
<td>0.43</td>
<td>2.56</td>
<td>0.168</td>
<td>2.56 ± 0.37</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.94</td>
<td>0.36</td>
<td>1.81</td>
<td>0.199</td>
<td>1.81 ± 0.14</td>
</tr>
</tbody>
</table>
Figure 1. Percentage change in assess time per sampling problem relative to $k_f = 100$ (left) and coverage probabilities (right) across different values of $k_f$.

We performed the same analysis for the remaining test problems and found similar results, summarized in Figure 1. The left-hand plot shows the percentage change in assess time per sampling problem, compared to that of $k_f = 100$, and the right-hand plot shows the coverage probabilities out of 300 independent runs as a function of $k_f$. For CEP1 and 4TERM there is not much change, therefore, the lines in Figure 1 for these two problems remain flat. (We discuss these two problems in more detail in the next section.) For all other problems, assess time per sampling problem remains relatively flat as $k_f$ drops from 100 down to about $k_f = 25$ and then starts to increase, with increases as large as 159% (for APL1P) when $k_f = 1$. Also, the coverage probability remains relatively high until about $k_f = 25$, but deteriorates with further decrease in $k_f$.

When determining an appropriate value of $k_f$, we consider the solution time, the computational efficiency of warm-starts, and coverage probabilities. Based on the above results, to minimize the computation time, we would choose $k_f$ as small as possible. However, small values of $k_f$ increase the risk of undercoverage and result in more computational effort per problem. Therefore, we would like to choose $k_f$ as small as possible while keeping computational efficiency and coverage probability at similar levels as those of larger values of $k_f$. The results in Table 4 and Figure 1 suggest that $k_f = 25$ provides a good balance between these considerations. The solution time is quicker compared to larger values of $k_f$, but the computational burden per problem remains small and the coverage probability remains reasonably high. When $k_f$ decreases below 25, the coverage probability can deteriorate substantially and the computational burden per problem increases steadily. In the rest of the computational experiments, we set $k_f = 25$ for all test problems.

We close this subsection with an additional remark on the results in Table 4. The effect of the sublinear $O(\log^2 k)$ sample-size formula becomes apparent in Table 4 via the average number of (SP$_n$)s solved (column 5) versus the
number of iterations, $T$ (column 7). When $T$ is small, these values agree, because we solve exactly one (SP$_n$) for assessing solution quality each iteration. However, when $k$ is larger, we can increase to $k + 1$ and still have $n_k = n_{k+1}$, after rounding. If we are not at a resampling iteration, this means (SP$_n$) and (SP$_{n+1}$) are identical and we do not have to (re)solve to obtain the $n$th problem. As above, the results are based on 300 independent runs of the sequential procedures. To reduce the effect of sampling when comparing the results, we use the same stream of random numbers to feed each estimation procedure. Also, because A2RP uses sample sizes that are even, $h$s can be more efficient than SRP so that they are even as well. Table 5 reports the number of iterations ($T$), the CI width on the optimality gap of the obtained solution ($h^* + \epsilon$) and an estimate of the coverage probability, i.e., the fraction of the 300 CIs that contained the true optimality gap ($\hat{p}$). For each of these, we report the average of the 300 runs, along with their 90% CI half-widths. We also report the average time it took for the 300 runs to complete (in seconds). We now discuss the results of Table 5 for each test problem in more detail.

6.5.2. Empirical Performance of the Sequential Procedure. We now further analyze coverage properties of our sequential procedure for two different optimality gap estimators, namely, estimators based on SRP and on A2RP (§6.2). Table 5 provides a summary of results for all test problems. As above, the results are based on 300 independent runs of the sequential procedures. To reduce the effect of sampling when comparing the results, we use the same stream of random numbers to feed each estimation procedure. Also, because A2RP uses sample sizes that are even, we round up the sample sizes for SRP so that they are even as well. Table 5 reports the number of iterations ($T$), the CI width on the optimality gap of the obtained solution ($h^* + \epsilon$) and an estimate of the coverage probability, i.e., the fraction of the 300 CIs that contained the true optimality gap ($\hat{p}$). For each of these, we report the average of the 300 runs, along with their 90% CI half-widths. We also report the average time it took for the 300 runs to complete (in seconds). We now discuss the results of Table 5 for each test problem in more detail.

CEP1 and 4TERM are both amenable to solution via sampling, i.e., we can obtain optimal solutions to these problems with high probability by solving a sampling problem with a relatively small number of observations. Results in Table 5 indicate that the sequential procedure works well in this case, quickly recognizing an optimal solution and exiting in very few iterations. We note that CIs on optimality gaps for CEP1 are all within 0.1% of optimality (see $z^*$ values in Table 2) and all solutions found are indeed optimal. The solution time for 4TERM is larger because its second stage problems take longer to solve.

The sequential procedure also works well for APL1P. The CI widths on the optimality gap are within 0.3% of optimality for both methods, and coverage probabilities are 0.90 or higher. For GBD, the CI widths are again within 0.3% of optimality for all initial sample sizes (100, 300, and 500), but the sequential procedure that uses SRP does not yield the desired coverage probability of 0.90. SRP has undercoverage even for our largest initial sample sizes. The sequential procedure that uses A2RP does achieve the desired coverage probability, albeit with somewhat wider CIs. Solution time can grow with the number of iterations, and so A2RP can require more computational effort. However, note that, for instance, for GBD:100, the average $T$ using A2RP is about five times that of SRP, yet the solution time only grew three times. This is because solving two smaller problems (A2RP) can be more efficient than solving one large problem (SRP). As the initial sample size increases (i.e., as $h \downarrow h'$), the sequential procedures take fewer iterations, yield smaller CIs on optimality gaps with higher coverage probabilities, and the differences between SRP and A2RP in terms of the number of iterations and solution times shrink. Even though we are solving problems that are five times larger for GBD:500 than for GBD:100, the growth in solution times are quite small, because the sequential procedures stop in fewer iterations when started with a larger sample size.

The primary reason the stopping iteration, $T$, tends to be larger for A2RP compared to SRP is due to the larger bias in the A2RP estimators because $Ez^*_n \leq E\hat{z}^*_n \equiv z^*$. Even with these larger $T$ values, the sample sizes at the stopping iteration, $h$, for A2RP are less than twice that of SRP. For example, for PGP2, APL1P, and GBD:100, 90% CIs

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>90% CI on $T$</th>
<th>90% CI on $h^* + \epsilon$</th>
<th>90% CI on $\hat{p}$</th>
<th>Solution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEP1</td>
<td>SRP</td>
<td>1.31 ± 0.19</td>
<td>243.31 ± 113.91</td>
<td>1.00 ± 0.00</td>
<td>0.028</td>
</tr>
<tr>
<td></td>
<td>A2RP</td>
<td>3.33 ± 0.59</td>
<td>345.12 ± 120.64</td>
<td>1.00 ± 0.00</td>
<td>0.051</td>
</tr>
<tr>
<td>4TERM</td>
<td>SRP</td>
<td>1.00 ± 0.00</td>
<td>0.00 ± 0.00</td>
<td>1.00 ± 0.00</td>
<td>0.500</td>
</tr>
<tr>
<td></td>
<td>A2RP</td>
<td>1.00 ± 0.00</td>
<td>0.00 ± 0.00</td>
<td>1.00 ± 0.00</td>
<td>0.492</td>
</tr>
<tr>
<td>PGP2</td>
<td>SRP</td>
<td>13.94 ± 2.91</td>
<td>15.23 ± 2.23</td>
<td>0.79 ± 0.04</td>
<td>1.141</td>
</tr>
<tr>
<td></td>
<td>A2RP</td>
<td>73.15 ± 8.93</td>
<td>14.06 ± 1.95</td>
<td>0.78 ± 0.04</td>
<td>3.944</td>
</tr>
<tr>
<td>APL1P</td>
<td>SRP</td>
<td>25.60 ± 3.27</td>
<td>53.11 ± 5.34</td>
<td>0.90 ± 0.03</td>
<td>5.373</td>
</tr>
<tr>
<td></td>
<td>A2RP</td>
<td>90.65 ± 10.53</td>
<td>72.63 ± 4.52</td>
<td>0.99 ± 0.01</td>
<td>14.510</td>
</tr>
<tr>
<td>GBD:100</td>
<td>SRP</td>
<td>38.85 ± 5.26</td>
<td>5.14 ± 0.60</td>
<td>0.70 ± 0.04</td>
<td>1.722</td>
</tr>
<tr>
<td></td>
<td>A2RP</td>
<td>195.98 ± 20.02</td>
<td>7.41 ± 0.56</td>
<td>0.94 ± 0.02</td>
<td>5.262</td>
</tr>
<tr>
<td>GBD:300</td>
<td>SRP</td>
<td>13.94 ± 2.40</td>
<td>2.18 ± 0.22</td>
<td>0.85 ± 0.03</td>
<td>2.905</td>
</tr>
<tr>
<td></td>
<td>A2RP</td>
<td>35.58 ± 7.77</td>
<td>3.32 ± 0.23</td>
<td>0.96 ± 0.02</td>
<td>6.627</td>
</tr>
<tr>
<td>GBD:500</td>
<td>SRP</td>
<td>6.00 ± 1.48</td>
<td>1.29 ± 0.17</td>
<td>0.85 ± 0.03</td>
<td>2.277</td>
</tr>
<tr>
<td></td>
<td>A2RP</td>
<td>14.57 ± 1.86</td>
<td>2.18 ± 0.13</td>
<td>0.95 ± 0.02</td>
<td>5.341</td>
</tr>
</tbody>
</table>
on $n_1$ are as follows: $120.21 \pm 2.64$ (SRP) and $170.31 \pm 3.67$ (A2RP); $271.21 \pm 6.38$ (SRP) and $349.03 \pm 9.01$ (A2RP); and, $146.12 \pm 3.53$ (SRP) and $209.18 \pm 4.73$ (A2RP), respectively. Therefore, the bias in the A2RP estimators is larger at termination, and hence it is not surprising that the CI widths tend to be larger for A2RP compared to SRP.

All test problems except PGP2 reach the desired coverage probability using A2RP estimators, and all CI widths on optimality gaps except for PGP2 are within 0.5% of optimality. PGP2 yields relatively larger CI widths (about 3% of optimality) and has undercoverage. This is because for this problem, the variance associated with some of the frequently obtained solutions is quite large. Moreover, the frequent appearance of so-called coinciding nonoptimal solutions decreases coverage; see §6 of Bayraksan and Morton (2006) for details on this concept and its effect on sampling-based approximations of PGP2. In the context of sequential sampling, to obtain better results, one might try to control the sampling variance by adding the condition $\{s_k \leq h\}$ to the stopping criterion. However, we have seen that this can result in very long runs, with $T \geq 1,000$. Another way to decrease variance and improve coverage probability is to increase the sample size. Towards this end, we fix $h' = 0.015$, and by letting $h \downarrow h'$, we run the sequential procedures for increasing initial sample sizes from 100 ($h = 0.332$) to 400 ($h = 0.188$). As before, we performed 300 independent runs of SRP and A2RP at each initial sample size. Figure 2 summarizes the results, showing the average values of $T$, $h_{T^2} + \epsilon$, $\hat{p}$, and the solution time. The coverage probability for A2RP is almost 0.90 at $n_1 = 200$ and remains at desirable levels. SRP’s coverage can be low, but improves as the initial sample size increases. CI widths, the number of iterations as well as the difference between SRP and A2RP for these performance measures shrink as the initial sample size increases, i.e., as $h \downarrow h'$. Like GBD, the growth in solution time is modest. For instance, increasing the initial sample size by a factor of 4 results in less than a factor of 1.2 increase in solution time for A2RP and less than a factor of 2 increase for SRP.

Overall, the results of Table 5 and Figure 2 indicate that the A2RP method, on average, takes more iterations to terminate, typically yields larger CIs on the optimality gap, and has more conservative coverage probabilities. CEP1, 4TERM, and APL1P indicate that SRP can work well for some problems. However, GBD and PGP2 suggest that for other problems, SRP can have undercoverage even for relatively large initial sample sizes. As $h \downarrow h'$, A2RP’s coverage probability seems to stabilize, whereas the CI widths as well as the number of iterations decrease and become comparable to SRP. The increase in solution time as the initial sample size grows is quite modest. We view the results as consistent with our observations in the nonsequential setting (Bayraksan and Morton 2006), i.e., A2RP tends to reduce the risk of undercoverage that can arise from the SRP estimators. Here, the sequential sampling procedure suggests to a “user” that a solution to a decision problem is of high quality, and for that reason we lean toward being conservative. Therefore, although the A2RP requires somewhat greater computational effort, its results are more robust across the problems we have examined, and we recommend A2RP over SRP.

Figure 2. Performance of the sequential procedure as $h \downarrow h'$, i.e., as $n_1$ grows, for PGP2.
7. Conclusions
In this paper, we develop a sequential sampling procedure for stochastic programs. We assume that a sequence of candidate solutions with at least one limit point that solves (SP) is given as input to the procedure. Then, the procedure assesses the quality of these candidate solutions with increasing sample size and terminates according to a stopping criterion. That criterion depends on the optimality gap estimate of the current solution and its associated variance. If the gap estimate is sufficiently small relative to its variance, then the procedure stops and outputs a CI on the optimality gap of the current candidate solution. If not, the sample size is increased. Below we summarize the contributions of the paper and list insights gained from our computational experiments. Then, we conclude the paper with future research directions.

- We provide specific rules to sequentially increase the sample size and to terminate that yield a CI on the resulting solution’s optimality gap. The CI is ensured to be small, within a prespecified relative tolerance, and the CI is ensured to be asymptotically valid, with a prespecified probability under certain conditions such as consistency of the variance estimator and the existence of moments, or the MGF, of the performance measure. The procedure terminates in a finite number of iterations, with probability one under mild assumptions on the limiting behavior of the sequence of candidate solutions and the optimality gap estimator.

- The required rate at which the sample size grows depends on moment information known regarding the performance measure. When the MGF of the performance measure exists, then the growth is \( O(\log k) \) in the iteration \( k \). If only the second moment exists, the rate is essentially \( O(k) \).

- At each iteration, the sequential procedure allows for augmenting previously generated observations, or, generating an entirely new set of observations. Computational results indicate that as we resample with a higher frequency, solution time decreases, but coverage probability deteriorates. We recommend \( k_f = 25 \) as providing a good balance between these considerations, based on the test problems we have examined.

- We apply the sequential sampling procedure with the SRP and A2RP estimators of Bayraksan and Morton (2006). In our experiments, we observed that A2RP can require more iterations and computational effort, but the SRP-based procedure has a greater risk of undercoverage. Under larger initial sample sizes: (i) the difference in computational effort between SRP and A2RP, along with the difference in their CI widths, shrinks, and (ii) coverage results for A2RP tend to reach desired levels more quickly than for SRP, at least for problems with a risk of undercoverage. As the initial sample size increases, the growth in solution times is modest.

- Based on our computational results, we recommend the use of A2RP. If the associated computational effort is excessive, SRP can be used instead, albeit with a risk of undercoverage for some problems.

In this paper, we used instances of a sampling problem (SP\(_m\)) with growing sample sizes, \( m_k \), at iteration \( k \), to generate a sequence of candidate solutions. For specific problem classes, such a sequence can be generated more efficiently using a sampling-based algorithm such as the stochastic decomposition algorithm of Higle and Sen (1996b) or the stochastic approximation algorithm of Lan et al. (2008). Importantly, these two algorithms also generate lower-bound estimators, which our procedure requires. The focus of this paper has been on the structure of the sequential procedure needed for assessing solution quality, but an important area for future research is to embed our procedure in these types of algorithms.

Another area for future research is to develop more efficient sequential sampling procedures while maintaining the desired asymptotic properties. For instance, adaptive sequential methods can be designed where the sampling method takes into account the information obtained about the problem so far. Carefully designed adaptive methods can be more efficient while maintaining the desired asymptotic properties. The performance of the sequential procedure depends on the quality of the optimality gap estimators. There is ongoing work to improve the quality of these estimators by reducing their bias and variance.

8. Electronic Companion
An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

Acknowledgments
The authors thank the referees and the associate editor for helpful comments and Andrzej Ruszczyński and Artur Świętańowski for access to their regularized decomposition code. This research is partially supported by the National Science Foundation under grants CMMI-0653916 and EFRI-0835930 and the Defense Threat Reduction Agency through grant HDTRA1-08-1-0029. An earlier abbreviated version of this paper appeared in Bayraksan and Morton (2007).

References
Bayraksan and Morton: A Sequential Sampling Procedure for Stochastic Programming


