Stochastic Root Finding via Retrospective Approximation

Huifen Chen
Department of Industrial Engineering
Chung Yuan Christian University
Chung Li, TAIWAN
E-mail: huifen@cycu.edu.tw

Bruce W. Schmeiser
School of Industrial Engineering
Purdue University
West Lafayette, Indiana 47907-1287, U.S.A.
E-mail: bruce@purdue.edu

ABSTRACT

Given a user-provided Monte Carlo simulation procedure to estimate a function at any specified point, the stochastic root-finding problem is to find the unique argument value to provide a specified function value. To solve such problems, we introduce the family of Retrospective Approximation (RA) algorithms. RA solves, with decreasing error, a sequence of sample-path equations that are based on increasing Monte Carlo sample sizes. Two variations are developed: IRA, in which each sample-path equation is generated independently of the others, and DRA, in which each equation is obtained by appending new random variates to the previous equation. We prove that such algorithms converge with probability 1 to the desired solution as the number of iterations grows, discuss implementation issues to obtain good performance in practice without tuning algorithm parameters, provide experimental results for an illustrative application, and argue that IRA dominates DRA in terms of the generalized mean squared error.

1 INTRODUCTION

The root-finding problem is to find the unique root x^* of the equation $g(x^*) = \gamma$, where $g : \Re^d \to \Re^d$. We consider stochastic root-finding problems (SRFPs), the case where $g(x)$ can only be estimated by a consistent estimator $\bar{y}(x)$ for any given value of x. SRFPs arise in designing stochastic systems with computer simulation: γ is the desired system performance, x is the value of a design variable, $g(x)$ is the corresponding system performance, and $\bar{y}(x)$ is the estimated performance obtained from a user-provided Monte Carlo simulation procedure. More specifically, the SRFP is defined as follows.

Stochastic Root Finding Problem (SRFP):

Given:

(a) a constant vector $\gamma \in \Re^d$,

(b) a (computer) procedure for generating, for any $x \in \Re^d$, a d-dimensional consistent estimate $\bar{y}(x)$ of $g(x)$.

Find: the unique root x^* satisfying $g(x^*) = \gamma$ using only the estimator $\bar{y}$.

Examples of SRFPs can be found in Chen and Schmeiser (1994a) and in Section 5. As an illustrative example, we consider here a one-dimensional problem arising in statistical inference: Compute the analogue of a Student's t critical value when the independent observations arise from an arbitrary distribution function, say F_V, rather than from the normal distribution. Specifically: given F_V, a sample size n, and a probability $1 - \alpha$, the problem is to determine the critical value $t_{1-\alpha,n}$ such that $\mathrm{P}\{T \le t_{1-\alpha,n}\} = 1 - \alpha$, where $T = \sqrt{n}(\bar{V} - \mu)/S$, $\bar{V} = n^{-1}\sum_{j=1}^{n} V_j$, $S^2 = (n-1)^{-1}\sum_{j=1}^{n}(V_j - \bar{V})^2$, the observations $V_1, V_2, ..., V_n$ are sampled independently from F_V, and $\mu = \mathrm{E}(V)$ assumed in the null hypothesis. This problem is an SRFP with true root $x^* = t_{1-\alpha,n}$, target $\gamma = 1 - \alpha$, and function $g(x) = \mathrm{P}(T \le x)$. The user-written computer procedure $\bar{y}(x)$ is the sample average of $y_1(x), ..., y_m(x)$, where $y_j(x) = I\{T_j \le x\}$, T_j is the jth observation of T, and I is the indicator function. That is, the computer procedure mimics the process: Each observation $y_j(x)$ is obtained by generating one random sample of size n from F_V, computing the value of T_j, and returning 1 if $T_j \le x$ and 0 otherwise.

This root-finding problem could be solved deterministically using numerical quadrature. Rather than $\bar{y}(x)$, the user-provided procedure would compute $g(x) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} I\{T \le x\} dF_V(v_1)...dF_V(v_n)$,

probably using an off-the-shelf quadrature routine. This value, in turn, could be given to an off-the-shelf deterministic root-finding routine. Even for this simple example, and even for small values of n, both simplicity of code and computing time favor the stochastic approach. Of course, for the important normal-distribution case, specialized deterministic routines are quite efficient.

We are interested in black-box algorithms to solve the SRFP. Given such algorithms, a practitioner needs only to provide a computer simulation procedure that provides the estimator $\bar{y}(x)$ of $g(x)$ by mimicking the behavior of the modeled system. Often, as in this example, $\bar{y}(x)$ is an unbiased sample average, but more generally, such as in problems involving steady-state behavior of stochastic systems, $\bar{y}(x)$ could be biased. Even more generally, $\bar{y}(x)$ need not be a sample average. The concepts necessary to develop such simulation procedures are discussed in standard textbooks, for example Law and Kelton (2000) and Banks et al. (1996).

The general problem is to solve d equations in d unknowns. As with deterministic root finding, many interesting problems lie in a single dimension. We consider only $d = 1$ in detail. The ideas, but not all of the details, of this paper extend to multiple dimensions, e.g., Chen (1998).

We propose, in Section 2, a family of *Retrospective Approximation (RA)* algorithms, which iteratively solve a sequence of *sample-path approximation problems* with increasing sample sizes. In each iteration, the sample-path approximation problem is solved to within an error tolerance; the root estimator is then a function of those solutions. Using results from M-estimators, we show in Section 3 that, under proper conditions, the RA root estimator converges to the true root with probability one (w.p.1). Section 4 is a discussion of implementation issues. Section 5 provides numerical results for two specific RA algorithms and empirically compares RA to stochastic-approximation algorithms (Robbins and Monro 1951), a classic stochastic root-finding approach. The family of RA algorithms has been discussed previously in Chen (1994) and Chen and Schmeiser (1994b).

That an algorithm converges is not sufficient to make the algorithm interesting; the finite-sample convergence can be so slow as to make the algorithm impractical, even when the asymptotic convergence rate is quite good. Our empirical experience with various retrospective approximation algorithms, however, has been good. Section 5.2 shows that RA yields root estimates with much smaller mean squared error (mse) than stochastic approximation for practical numbers of replications. Work continues on stochastic approximation, although usually for optimization rather than root finding (Andradóttir 1992, Fu 1994, Fu and Hill 1997, L'Ecuyer et al. 1994, and L'Ecuyer and

Glynn 1994). Simon (1998) proposes a natural stochastic root-finding procedure that iteratively solves a sequence of approximation functions of $g(x)$, assuming that the solution can be determined exactly. A given approximation function depends on all the past solutions and their function estimates. This procedure converges if the root-finding function g is continuous, the estimator $\bar{y}$ is uniformly consistent, and the approximation function satisfies the "cluster point property." Despite the convergence proof, it is not clear how to construct an approximation function that satisfies the cluster point property, is easy to implement, and is robust with respect to the algorithm parameters.

2 RETROSPECTIVE APPROXIMATION

Here we develop RA algorithms. In the following five subsections we discuss sample-path approximations, two variations of RA algorithms, the rationale behind the RA logic, estimators for the variance of the retrospective root estimators, and retrospective approaches for stochastic optimization.

2.1 Sample-Path Approximations

Fundamental to the retrospective approach is the concept of the sample-path approximation to the function g. At any point x, this approximation is simply $\bar{y}(x)$. The approximation to g is obtained by using common random numbers for every point x. We let $\underline{\omega} = \{\omega_1, ..., \omega_m\}$ denote the random numbers used to obtain $\bar{y}(x) = \bar{y}_m(x; \underline{\omega})$. Holding $\underline{\omega}$ constant over all points x yields a sample-path approximation $\bar{y}_m(\cdot; \underline{\omega})$ to g. Although given $\underline{\omega}$ the sample-path approximation $\bar{y}_m(x; \underline{\omega})$ is a deterministic function of x, we need not write it explicitly; rather we calculate its value only at each desired point x.

The sample-path equation

$$\bar{y}_m(x^*(\underline{\omega}) \ ; \ \underline{\omega}) = \gamma \tag{1}$$

defines the random root $x^*(\underline{\omega})$, which is an estimate of the true root x^*. We call $x^*(\underline{\omega})$ the *retrospective root*.

For the example in Section 1, the random numbers ω_j are used to generate the j^{th} observation $y_j(x) = y(x; \omega_j) = I\{T_j(\omega_j) \leq x\}$. Given independently generated random numbers $\omega_1, ..., \omega_m$, the

sample-path equation is

$$\bar{y}_m(x^*(\underline{\omega});\underline{\omega}) = m^{-1}\sum_{j=1}^{m} y(x^*(\underline{\omega});\omega_j) = m^{-1}\sum_{j=1}^{m} I\{T_j(\omega_j) \le x^*(\underline{\omega})\} = 1-\alpha \ . \tag{2}$$

Despite the uniqueness of the true root $t_{1-\alpha,n}$ of the strictly increasing, continuous (assuming F_V is continuous) function g, the retrospective root might not be unique or might not exist. The sample-path equation has many roots if $1-\alpha$ lies on a step and has no root otherwise. In the latter case, there is, however, a unique point $x^*(\underline{\omega})$ at which $\bar{y}_m$ crosses $1-\alpha$. In Section 3.1 we relax the definition of retrospective roots to allow "crossing" roots.

2.2 RA Algorithms

RA iteratively solves approximately a sequence of sample-path equations

$$\bar{y}_{m_i}(x^*(\underline{\omega}_i) \ ; \ \underline{\omega}_i) = \ \gamma \tag{3}$$

for $i = 1, 2,...$, where the components of $\underline{\omega}_i = \{\omega_{i,1}, ..., \omega_{i,m_i}\}$ are generated independently. RA uses a strictly increasing sample-size sequence $\{m_i\}$. At each retrospective iteration i, RA uses a numerical root-finding algorithm to evaluate $\bar{y}_{m_i}(\cdot;\underline{\omega}_i)$ at one or more x values to find a *retrospective solution* $x(\underline{\omega}_1, ..., \underline{\omega}_i)$ that is close to the retrospective root $x^*(\underline{\omega}_i)$. In particular, RA uses the retrospective-iteration stopping criterion $|x(\underline{\omega}_1, ..., \underline{\omega}_i) - x^*(\underline{\omega}_i)| < \epsilon_i$, where $\{\epsilon_i\}$ is a positive sequence converging to zero. Possibly $\{\epsilon_i\}$ is a function of the previous retrospective solutions and therefore random; in this case we assume only w.p.1 convergence to zero. One approach to satisfy the stopping criterion is to bound the retrospective root and use bisection search until the stopping criterion is satisfied.

The values of x at which the sample-path approximation $\bar{y}_{m_i}(\cdot;\underline{\omega}_i)$ is evaluated by the numerical root-finding algorithm, as well as the number of such x values, are random, depending upon the particular sample-path equation and (less directly) any information from earlier retrospective iterations. For example, in our implementation the numerical root-finding algorithm uses an initial x computed from earlier sample-path equations. In fact, the computational efficiency of RA arises because small sample sizes in the early retrospective iterations are used to find the region of the root before more expensive and precise computations are performed with larger sample sizes in the

later retrospective iterations.

Two strategies seem natural for seeding the retrospective iterations. In *Dependent RA (DRA)* all retrospective iterations use the same random-number stream. Therefore, the retrospective iterations are dependent because the m_{i-1} observations $\underline{\omega}_{i-1}$ in the retrospective iteration $i-1$ are the first m_{i-1} observations of the m_i observations $\underline{\omega}_i$ in retrospective iteration i. In *Independent RA (IRA)* all retrospective iterations are independently seeded. Therefore, the observations from iteration to iteration are independent, with retrospective iteration i containing m_i new observations $\underline{\omega}_i$ independent of $\underline{\omega}_1, ..., \underline{\omega}_{i-1}$.

After i retrospective iterations, the root estimator $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)$ differs between DRA and IRA. In DRA,

$$\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) = x(\underline{\omega}_1, ..., \underline{\omega}_i), \tag{4}$$

the i^{th} retrospective solution. In IRA, the root estimator is the weighted average of the solutions $x(\underline{\omega}_1)$, ..., $x(\underline{\omega}_1, ..., \underline{\omega}_i)$:

$$\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) \;=\; \sum_{j=1}^{i} m_j \; x(\underline{\omega}_1, ..., \underline{\omega}_j) \;/\; \sum_{j=1}^{i} m_j. \tag{5}$$

Section 5.1 empirically compares the efficiency of DRA and IRA algorithms.

Specifically, RA algorithms work as follows.

RA Algorithms:

Components:

1. an initial sample size m_1 and a rule for successively increasing m_i for $i \geq 2$,
2. a rule for computing an error-tolerance sequence $\{\epsilon_i\}$ that goes to zero w.p.1, and
3. a numerical root-finding method for solving sample-path equation (3) to within a specified error ϵ_i for each i.

Find: the root x^*.

Step 0. Initialize the retrospective iteration number $i = 1$. Set m_1 and ϵ_1.

Step 1. Generate $\underline{\omega}_i$. For IRA, $\underline{\omega}_i$ is m_i new observations, generated independently of $\underline{\omega}_{i-1}$. For DRA, $\underline{\omega}_i$ is obtained by generating $m_i - m_{i-1}$ new independent observations and appending them to $\underline{\omega}_{i-1}$.

Step 2. Use the numerical method to solve the deterministic sample-path equation (3) to obtain a retrospective solution $x(\underline{\omega}_1, ..., \underline{\omega}_i)$ that satisfies $|x(\underline{\omega}_1, ..., \underline{\omega}_i) - x^*(\underline{\omega}_i)| < \epsilon_i$.

Step 3. Compute the root estimator $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)$ using Equation (4) for DRA or (5) for IRA.

Step 4. Compute m_{i+1} and ϵ_{i+1}. Set $i \longleftarrow i + 1$ and go to Step 1.

In addition to choosing between DRA and IRA, a specific RA algorithm is obtained by choosing the three components: sample-size rule, error-tolerance sequence, and numerical root-finding method. We discuss these choices, as well as stopping rules (for the entire RA algorithm), in Section 4.

2.3 RA Rationale

Here we briefly discuss the rationale of the RA algorithm structure. We have three criteria for algorithm development: practical computational efficiency, guaranteed convergence to the root x^*, and a standard-error estimator for the root estimator. The first two underlie the discussion here, and the last we discuss in the next subsection.

RA algorithms use, by definition, multiple iterations with increasing sample sizes and decreasing error tolerances. The key idea for practical computational efficiency is that previous retrospective iterations, based upon small sample sizes, can provide information for efficient numerical solution in future retrospective iterations. Computing in early iterations is inexpensive because sample sizes are small; computing in later iterations is inexpensive because the approximate location of the retrospective root is known. Also for efficiency, the error tolerances need to be appropriately sized: there is no need to chase randomness. In earlier retrospective iterations the retrospective root has larger sampling error (due to smaller sample sizes), so the error tolerances should be larger; later the retrospective root has less sampling error, so the error tolerances should be smaller. The second criterion—guaranteeing convergence—is obtained by approaching two limits simultaneously: the sample-path approximation $\bar{y}_m$ approaches g, because of increasing sample size m, and the retrospective solutions approach the retrospective roots, because the error tolerances converge to zero.

Two other sample-size rules, both commonly used, have disadvantages compared to increasing sample sizes. The first—to use a single iteration, called a *static* algorithm in Shapiro (1996)—is typical, but often inefficient. Here only m_1 needs to be specified, and a solution $x(\underline{\omega}_1)$ of Equation (3) with $i = 1$ is returned. The sample size m_1 is necessarily large for Equation (3) with $i = 1$ to well approximate g, which causes the root-finding method to be inefficient because the lack of information from previous iterations causes the method to examine many points x, each requiring the computation of $\bar{y}_{m_1}(x; \underline{\omega}_1)$ based on a large sample size of m_1. We discuss this first sample-size rule in Section 3.1. A second sample-size rule, also not allowed within RA, is to use k iterations, each of the same sample size: $m_1 = m_2 = \cdots = m_k$. The multiple iterations allow subsequent iterations to begin with a good guess of the retrospective root $x^*(\underline{\omega}_i)$, saving computational effort. The disadvantage is lack of convergence as k goes to infinity because of fixed finite sample size m_1. Any bias in the retrospective root for the true root (or also in the retrospective solution for the retrospective root) does not go to zero. As with using k independent replications in any simulation experiment, reasonable practical performance can sometimes be obtained with such an approach (Healy 1992, p. 9). Although bias cannot be estimated, estimating the standard error of the root estimator is straightforward. The tradeoffs between these two sample-size rules are analogous to those between steady-state simulation using one-long run versus k shorter runs: less bias versus easy standard-error estimation. RA is designed to obtain both advantages.

By definition of RA, the new observations in Step 1 (m_i in IRA and $m_i - m_{i-1}$ in DRA) that are used to obtain $\bar{y}_{m_i}(x; \underline{\omega}_i)$ must be independent of previously generated observations so that the variance estimators can work. The new observations do not, however, need to be independent of each other. Any variance reduction via dependence-induction methods must be applied within the set of new observations, thereby hiding the dependence within $\bar{y}_{m_i}(x)$. For example, antithetic pairs can be applied directly for either IRA or DRA. Stratified sampling, however, is straightforward for IRA but not for DRA. Application to steady-state simulation is straightforward if $\underline{\omega}$ is chosen to be the random numbers, which are easily made independent; the dependence between the retrospective roots will become negligible as the sample size grows.

2.4 RA Variance Estimation

The third algorithm criterion is the ability to estimate the standard error of the algorithm's current root estimator $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)$. We derive variance estimators for both IRA and DRA here. As discussed in Section 4.1, these estimators are helpful for determining error tolerances, for use within the numerical root-finding algorithm, for stopping the overall algorithm, and for reporting precision of the final root estimator. These variance estimators are based on assumptions that are consistent with our intuition and computational results in Section 5, but certainly do not hold exactly in practice. For example, the computational results show that bias in the variance estimator is large in early iterations before becoming negligible.

First consider IRA. For now assume that IRA retrospective solutions are unbiased uncorrelated estimators of the root x^*, with variances inversely proportional to sample size; that is, $m_j \text{Var}(x(\underline{\omega}_1, ..., \underline{\omega}_j)) = \nu^2$ for each j. RA algorithms contain no systematic reason for estimates to be high or low, so biases should be small. That the variance is inversely proportional to sample size is natural, at least asymptotically (see also Lemma 2 in Section 3.1). The independent retrospective roots $\{x^*(\underline{\omega}_j)\}$ of IRA yield retrospective solutions $\{x(\underline{\omega}_1, ..., \underline{\omega}_j)\}$ that are nearly uncorrelated; some correlation might arise from the use of information from previous retrospective iterations, such as an initial search point. This correlation should be minor because RA forces the j^{th} retrospective solution to be close to (within ϵ_j of) the j^{th} retrospective root. Consider the covariance between two successive retrospective solutions. Let $b(\underline{\omega}_1, ..., \underline{\omega}_j)$ denote the error of the j^{th} retrospective solution, i.e.,

$$x(\underline{\omega}_1, ..., \underline{\omega}_j) = x^*(\underline{\omega}_j) + b(\underline{\omega}_1, ..., \underline{\omega}_j).$$

Then $\text{Cov}[x(\underline{\omega}_1, ..., \underline{\omega}_j),\ x(\underline{\omega}_1, ..., \underline{\omega}_{j+1})] = \text{Cov}[x^*(\underline{\omega}_j) + b(\underline{\omega}_1, ..., \underline{\omega}_j),\ x^*(\underline{\omega}_{j+1}) + b(\underline{\omega}_1, ..., \underline{\omega}_{j+1})] = \text{Cov}[x^*(\underline{\omega}_j),\ b(\underline{\omega}_1, ..., \underline{\omega}_{j+1})] + \text{Cov}[b(\underline{\omega}_1, ..., \underline{\omega}_j),\ \ b(\underline{\omega}_1, ..., \underline{\omega}_{j+1})]$ because $\underline{\omega}_1$, $\underline{\omega}_2$,... are generated independently in IRA. The first covariance is small because (i) $x^*(\underline{\omega}_j)$ and $b(\underline{\omega}_1, ..., \underline{\omega}_{j+1})$ are correlated only through $\underline{\omega}_j$ not the other $\underline{\omega}$'s, and (ii) the value of $b(\underline{\omega}_1, ..., \underline{\omega}_{j+1})$ depends more on $x(\underline{\omega}_1, ..., \underline{\omega}_j)$, for example as an initial point, but little on $x^*(\underline{\omega}_j)$. The second covariance is small because (i) the signs of the two errors $b(\underline{\omega}_1, ..., \underline{\omega}_j)$ and $b(\underline{\omega}_1, ..., \underline{\omega}_{j+1})$ are nearly independent, and (ii) the magnitude of the two errors depends mainly on their error tolerances ϵ_j and ϵ_{j+1}; and hence

if the error tolerances $\{\epsilon_l : l = 1, 2, ...\}$ are substantially smaller than the standard deviation of the current retrospective root, then the correlation between successive errors should be small. For these reasons assuming zero correlation seems reasonable.

These IRA assumptions yield two useful results. First, after i IRA retrospective iterations, the minimal-variance unbiased root estimator is $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)$, the weighted average of the i retrospective solutions, as given in Equation (5). Second, these assumptions provide an estimator of the variance of the root estimator. The variance is $\text{Var}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)) = \nu^2 / \sum_{j=1}^{i} m_j$. Therefore, after i retrospective iterations, an unbiased sum-of-squared-differences estimator of the variance of the IRA root estimator is, for $i > 1$,

$$\begin{aligned}
\widehat{\text{Var}}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)) &= \hat{\nu^2} / \sum_{j=1}^{i} m_j \\
&= \sum_{j=1}^{i} m_j [x(\underline{\omega}_1, ..., \underline{\omega}_j) - \bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)]^2 / [(i-1) \sum_{j=1}^{i} m_j], \\
&= [\sum_{j=1}^{i} m_j x^2(\underline{\omega}_1, ..., \underline{\omega}_j) - (\sum_{j=1}^{i} m_j) \bar{x}^2(\underline{\omega}_1, ..., \underline{\omega}_i)] / [(i-1) \sum_{j=1}^{i} m_j]. \qquad (6)
\end{aligned}$$

Now consider DRA. We make assumptions and arguments analogous to those of IRA. Assume that the DRA retrospective solutions are unbiased estimators of the root with variance inversely proportional to the sample size and that covariances are inversely proportional to the larger sample size; that is, $\text{Cov}[x(\underline{\omega}_1, ..., \underline{\omega}_j),\ x(\underline{\omega}_1, ..., \underline{\omega}_k)] = \nu^2 / \max\{m_j, m_k\}$ for each j and k. The arguments for unbiasedness and variances are identical to those for IRA. The argument for covariance is based on the assumption that when $j < k$, the k^{th} retrospective solution $x(\underline{\omega}_1, ..., \underline{\omega}_k) = m_j m_k^{-1}$ $x(\underline{\omega}_1, ..., \underline{\omega}_j) + (m_k - m_j) m_k^{-1}\ x(\tilde{\underline{\omega}})$, the weighted average of the j^{th} retrospective solution $x(\underline{\omega}_1, ..., \underline{\omega}_j)$ and the retrospective solution $x(\tilde{\underline{\omega}})$ of the sample-path equation $\bar{y}_{\tilde{m}}(x; \tilde{\underline{\omega}}) = \gamma$, where $\tilde{m} = m_k - m_j$ and $\tilde{\underline{\omega}} = \underline{\omega}_k \backslash \underline{\omega}_j = \{\omega_{m_j+1}, ..., \omega_{m_k}\}$, the set of $(m_k - m_j)$ new observations independent of $\underline{\omega}_j$. This covariance assumption is true if the retrospective solution equals the retrospective root at every iteration, g is linear, and $y(x; \omega) = g(x) + z(\omega)$, where the random noise $z(\omega)$ is functionally independent of x for every ω. If this assumption is not true, then observations are becoming either more or less valuable. Under the assumption, $\text{Cov}[x(\underline{\omega}_1, ..., \underline{\omega}_j), x(\underline{\omega}_1, ..., \underline{\omega}_k)] = \text{Cov}[x(\underline{\omega}_1, ..., \underline{\omega}_j),\ m_j m_k^{-1}$ $x(\underline{\omega}_1, ..., \underline{\omega}_j) + (m_k - m_j) m_k^{-1}\ x(\tilde{\underline{\omega}})] = m_j m_k^{-1} \text{Var}[x(\underline{\omega}_1, ..., \underline{\omega}_j)] = \nu^2 / m_k$. Hence the form of the

covariance is obtained.

These DRA assumptions suggest an estimator of its variance. The variance of the DRA root estimator can be estimated using

$$\begin{aligned} \widehat{\mathrm{Var}}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)) &= \hat{\nu^2}/m_i \\ &= \frac{\sum_{j=1}^{i-1} \alpha_{ij}[x(\underline{\omega}_1, ..., \underline{\omega}_j) - x(\underline{\omega}_1, ..., \underline{\omega}_i)]^2}{i-1}, \quad i > 1, \end{aligned} \tag{7}$$

where the coefficients $\alpha_{ij} = m_j/(m_i - m_j)$ are chosen to make each term of the sum an unbiased estimator of the variance. Because α_{ij} functionally depends on m_i, the estimator cannot be computed using cumulative sums as in Equation (6), so the values of the i retrospective solutions must be stored. For any fixed i, however, m_i is fixed; then expanding the squared differences allows computation using only the two cumulative sums for $\alpha_{ij}x(\underline{\omega}_1, ..., \underline{\omega}_j)$, and $\alpha_{ij}x^2(\underline{\omega}_1, ..., \underline{\omega}_j)$ over j.

2.5 Retrospective Approaches for Stochastic Optimization

The genesis of RA algorithms lies in retrospective optimization. The optimization problem is to find the optimal point of an objective function using only an estimator of the function at any feasible point x (Fu 1994). A retrospective approach optimizes sample-path problems that approximate the optimization problem of interest. Rubinstein and Shapiro (1993) use the phrase *stochastic counterpart* and Gürkan et al. (1994), for example, use *sample path* for this approach. We adopt Healy and Schruben's (1991) phrase *retrospective* to capture the idea of solving a random problem that happened in the past. Robinson (1996) contains a good summary of such optimization algorithms.

The design of the family of RA algorithms has five fundamental features. We discuss these features and how they distinguish RA from retrospective optimization algorithms as follows:

- We use a sequence of sample-path approximations with increasing sample sizes and decreasing error tolerances. This structure is central to our simultaneous goals of proving convergence and achieving good practical performance (as discussed in Section 2.3). Shapiro (1996) describes this approach as dynamic to distinguish it from the more-typical static algorithms, which solve a single sample-path problem. Shapiro and Homem-de-Mello (1997) and Homem-de-Mello et al. (1999) recently have used a structure similar to ours (also discussed earlier in

Chen and Schmeiser 1994b and Chen 1994), but with random sample sizes and a stopping rule based on a statistical test of hypothesis. Shapiro and Wardi (1996) consider gradient descent algorithms, which are both dynamic and Markovian. The bundle-type method (Plambeck et al. 1996) is dynamic, but the error tolerance does not decrease.

- Averaging solutions from a sequence of independent problems, which we argue is more efficient. That is, IRA is more efficient than DRA because DRA reprocesses old data whereas IRA is always working on new data (as discussed further in Section 5.1). Healy and Schruben (1991) average solutions of independent sample-path approximations, but each has the same sample size.

- A black-box approach, in which the problem structure is not used. Our work is in the spirit of methods such as bisection search or regula falsi rather than much of the recent literature that exploits problem structure. Stochastic approximation (for example, Andradóttir 1992, L'Ecuyer et al. 1994, and L'Ecuyer and Glynn 1994) and bundle-type methods (Plambeck et al. 1996) fall into this category. Healy (1992) and Healy and Xu (1994, 1995), on the other hand, develop problem-specific algorithms.

- Robustness. We strive for good practical performance without algorithmic tuning. RA logic is non-Markovian, in that information from previous sample-path approximations is used in the solution of the current approximation. The bundle-type algorithm used by Plambeck et al. (1996) is also non-Markovian. In contrast, the stochastic approximation algorithm is typically Markovian.

- Convergence proof. We prove convergence under fairly broad conditions. Shapiro (1996) proves convergence for various static algorithms. Shapiro and Wardi (1996) prove convergence of dynamic Markovian algorithms. Robinson (1996) proves convergence of the single-iteration bundle-type algorithm.

Substantial earlier work considered static algorithms. Rubinstein and Shapiro (1993) solve stochastic optimization problems by solving SRFPs, finding the zero of the associated gradient functions. They estimate the optimal point by finding the zero of a sample-path gradient function using the score-function gradient estimates from a single long-run simulation experiment. They

assume that a finite-time algorithm is available for solving the sample-path equation exactly; that is, in our notation they assume that $x(\underline{\omega}_1) = x^*(\underline{\omega}_1)$. Healy and Schruben (1991) solve stochastic optimization problems by analyzing each problem's structure to find the exact optimal point of a sample-path (or retrospective) objective function. A specific algorithm therefore differs from problem to problem (see Healy and Schruben 1991 and Fu and Healy 1992). Huber (1964) proposed M-estimators, which are obtained by optimizing an error function based on sample data; examples include maximum-likelihood and least-square estimators. In many applications, such optimization is obtained by solving for the zero of the gradient function. Following Serfling (1980, p. 243) we view M-estimators as the solution of a sample-path equation, such as Equation (1). The retrospective roots $x^*(\underline{\omega})$ satisfying Equation (1) and $x^*(\underline{\omega}_i)$ satisfying Equation (3) are therefore M-estimators.

3 CONVERGENCE OF RA ALGORITHMS

We show here that under weak conditions RA algorithms converge w.p.1 to the true root x^* that satisfies $g(x^*) = \gamma$. As discussed in Section 2.2, RA algorithms estimate x^* by solving, to within an error bound, a sequence of sample-path equations based on increasing sample sizes. To show the convergence of RA algorithms, we consider two cases: (i) a single RA iteration using a long-sample-path equation, and (ii) a complete RA algorithm using a sequence of equations of the form (3) for $i = 1, 2, ...$ The first case, discussed in Section 3.1, could be used to solve problems with a single iteration with a fixed value of m_1 and ϵ_1 (as with the static methods of Section 2.5), but our intent here is to use the single-iteration results of Case (i) to prove in Section 3.2 that RA algorithms converge to x^* w.p.1 as the iteration index i goes to infinity. The advantages of using multiple retrospective iterations rather than a single iteration is discussed in Section 2.3. In Section 4 we discuss the choice of RA algorithm components.

3.1 Case (i): A Single RA Iteration

Consider using an RA algorithm but stopping after the first retrospective iteration using only the first sample-path equation, Equation (3) with $i = 1$, and returning $x(\underline{\omega}_1)$ as the estimator of the root x^*. (IRA and DRA are identical in this case.) We consider here the asymptotic behavior of $x(\underline{\omega}_1)$ in the limit as the first-iteration sample size m_1 goes to infinity and ϵ_1 goes to zero. Although

in practice m_1 and ϵ_1 are fixed, the asymptotic analysis here is interesting in the same sense that it is interesting to show that the sample mean converges to the population mean as sample size goes to infinity even though in practice sample size does not grow.

To simplify notation and provide emphasis on the sample size, we denote the sample size m_1 by m, the error ϵ_1 (dependent on the sample size) by $\epsilon(m)$, the sample path $\underline{\omega}_1$ by $\underline{\omega}$, the retrospective root $x^*(\underline{\omega}_1)$ by $X^*(m)$, and the retrospective solution $x(\underline{\omega}_1)$ by $X(m)$. Equation (3) with $i = 1$ is then

$$\bar{y}_m(\, X^*(m);\, \underline{\omega}\,) = \gamma, \tag{8}$$

where $\bar{y}_m(x; \underline{\omega})$ is a consistent estimate of $g(x)$ for all real x (i.e., for any x, w.p.1 $\bar{y}_m(x; \underline{\omega})$ converges to $g(x)$ as $m \to \infty$.) The retrospective root $X^*(m)$ satisfying Equation (8) is an M-estimator (Section 2.5). Our purpose here is to show that $X(m)$, the approximation to $X^*(m)$, converges to the true root x^* as the sample size m goes to infinity.

We show convergence in two parts: (1) $X^*(m)$ converges to x^* w.p.1, and then (2) $X(m)$ converges to x^* w.p.1. Since $X^*(m)$ is an M-estimator, the first part follows from the results of M-estimators as stated in Lemma 1, based on the assumption that $\bar{y}_m(x; \underline{\omega})$ is an unbiased estimate that averages m monotonic functions. We further develop Lemmas 3 and 4 to extend the result to more-general functions g and sample-path approximations $\bar{y}_m(\cdot; \underline{\omega})$. The second part, the consistency of $X(m)$, is shown in Lemma 5.

Since Equation (8) may have no root or multiple roots, we define the concepts of a *crossing root* and a *crossing set*, which are then used in Lemmas 3, 4, and 5 and Theorems 1 and 2 in Section 3.2. Every root is a crossing root; a crossing root might additionally be a discontinuity point where $\bar{y}_m(\cdot; \underline{\omega})$ crosses γ. To be more specific, we define the crossing root and crossing set as follows. Let sets

$$\begin{aligned} R^N &= \{x :\, \bar{y}_m(x;\, \underline{\omega}) - \gamma < 0, \quad x \in \Re\}, \\ R^Z &= \{x :\, \bar{y}_m(x;\, \underline{\omega}) - \gamma = 0, \quad x \in \Re\}, \quad \text{and} \\ R^P &= \{x :\, \bar{y}_m(x;\, \underline{\omega}) - \gamma > 0, \quad x \in \Re\}. \end{aligned} \tag{9}$$

Assume that the function $\bar{y}_m(\cdot; \underline{\omega})$ is defined over the whole real-number line $\Re$. Then the three

mutually exclusive sets R^N, R^Z, and R^P partition $\Re$, i.e, $\Re = R^N \cup R^Z \cup R^P$. Further define

$$\begin{aligned} R^{NP} &= \{x : x \in R^N \text{ and } x \text{ on the boundary of } R^P \text{ or } R^Z\} \cup \\ &\quad \{x : x \in R^P \text{ and } x \text{ on the boundary of } R^N \text{ or } R^Z\}. \end{aligned}$$

Then R^Z contains all (true) roots of Equation (8) and R^{NP} contains all discontinuity points where $\bar{y}_m(\cdot;\underline{\omega})$ crosses γ but does not intersect the level γ. The crossing set $C_m(\underline{\omega})$ of Equation (8) is therefore defined as

$$C_m(\underline{\omega}) = R^Z \cup R^{NP}. \tag{10}$$

Every element of $C_m(\underline{\omega})$ is a "crossing root". The set $C_m(\underline{\omega})$ is empty if the function $\bar{y}_m(\cdot;\underline{\omega})$ lies entirely below or entirely above γ.

Lemmas 1 and 2 below use the concept of isolated root, which is a discrete point in $C_m(\underline{\omega})$. Of (true) roots, crossing roots, and isolated roots, the most general are crossing roots; all (true) roots and all isolated roots are also crossing roots. A crossing root, however, might be a (true) root, an isolated root, both, or neither.

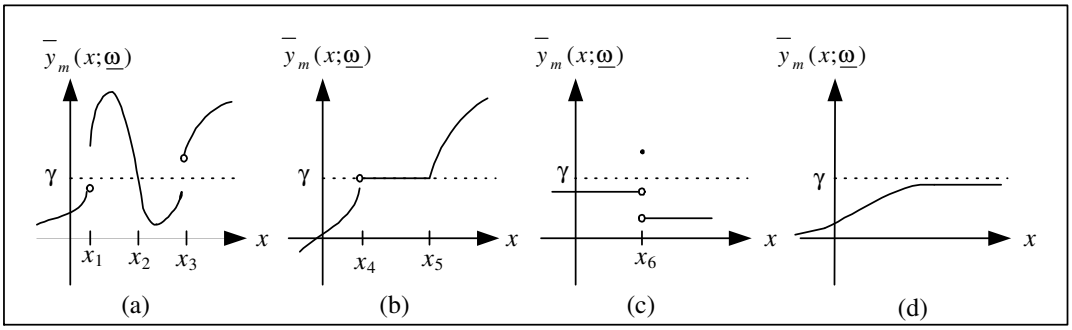

Figure 1: Four functions illustrating different types of roots.

We illustrate crossing roots, isolated roots, and (true) roots for four different functions $\bar{y}_m(\cdot;\underline{\omega})$ in Figure 1. The function $\bar{y}_m(\cdot;\underline{\omega})$ in Subfigure 1(a) crosses the value γ at three distinct points x_1, x_2, and x_3; hence, any of these points is a crossing root and $C_m(\underline{\omega}) = \{x_1, x_2, x_3\}$. All three of these points are also isolated roots. The only root is x_2; the points x_1 and x_3 are not roots because $\bar{y}_m(x_1;\underline{\omega}) \neq \gamma$ and $\bar{y}_m(x_3;\underline{\omega}) \neq \gamma$. The function $\bar{y}_m(\cdot;\underline{\omega})$ in Subfigure 1(b) crosses the value γ only once with an intersecting interval $(x_4, x_5]$ on the x-axis. Hence, all points in the interval $(x_4, x_5]$

are roots, all points in $[x_4, x_5]$ are crossing roots but none is an isolated root, and $C_m(\underline{\omega}) = [x_4, x_5]$. The function $\bar{y}_m(\cdot; \underline{\omega})$ in Subfigure 1(c) lies below the level γ except at one point x_6, where $\bar{y}_m(\cdot; \underline{\omega})$ crosses the value γ and then immediately drops down to below γ. Therefore, $C_m(\underline{\omega}) = \{x_6\}$; x_6 is also an isolated (crossing) root but not a (true) root. The function $\bar{y}_m(\cdot; \underline{\omega})$ in Subfigure 1(d) has no intersection with the level γ. Hence, there is no root and no crossing root; $C_m(\underline{\omega})$ is the empty set.

Equation (8) may have zero, one, or multiple crossing roots. Zero crossing roots occur when the sample size m is small, allowing $\bar{y}_m(x; \underline{\omega})$ to lie entirely below or above γ for all real x. Multiple crossing roots occur, for example, when $\bar{y}_m$ is a step function and one of the steps has height γ. Step functions occur in the two examples of Sections 1, 2 and 5, where $y(x)$ is an indicator function. Indicator functions occur, for example, when $y(x)$ indicates whether an event occurs or not.

We redefine the retrospective root $X^*(m)$ of Equation (8) to be

$$X^*(m) = \begin{cases} \text{any crossing root} & \text{if } C_m(\underline{\omega}) \text{ is not empty} \\ 0 & \text{otherwise} \end{cases} . \tag{11}$$

The retrospective root $X^*(m)$ can be selected arbitrarily from $C_m(\underline{\omega})$ using any rule that produces a solution sequence $\{X^*(m)\}$. This selection rule is usually implicit in the solution method. The choice of $X^*(m) = 0$ when $C_m(\underline{\omega})$ is empty is arbitrary because asymptotically $C_m(\underline{\omega})$ is empty with probability 0. A better practical value might be the current estimate of x^*.

Asymptotically the retrospective root $X^*(m)$ is an M-estimator. Lemmas 1 and 2 show the consistency and asymptotic normality of M-estimators, and hence can be used for retrospective roots. Proofs can be found in Huber (1964) and Serfling (1980, p. 251), respectively.

Lemma 1 *Let x^* be an isolated root of $g(x) = \gamma$. Suppose that $\bar{y}_m(x; \underline{\omega}) = \sum_{i=1}^m y(x; \omega_i)/m$ and each $y(x; \omega_i)$ yields an unbiased estimator of $g(x)$ for every x. Further, suppose that, for every ω, the function $y(x; \omega)$ is monotone in x. Then x^* is unique, and any solution sequence $\{X^*(m)\}$ satisfying Equation (8) converges to x^* w.p.1 as $m \to \infty$. Further, if with probability 1 there is a neighborhood of x^* in which $y(x; \omega)$ is a continuous function of x, then there exists such a solution sequence.*

Lemma 2 (Serfling 1980, p. 251) states conditions under which $\sqrt{m}(X^*(m) - x^*)$ is asymptotically normally distributed.

Lemma 2 *Let x^* be an isolated root of $g(x) = \gamma$. Suppose the unbiasedness and monotonicity of $y(x\ ;\ \omega)$ as in Lemma 1 . Further, suppose that $g(x)$ is differentiable at x^* with $g'(x^*) \neq 0$ and that $\mathrm{E}[y^2(x;\omega)]$ is finite for x in a neighborhood of x^* and is continuous at x^*. Then, as $m \to \infty$ any scaled solution sequence $\{\sqrt{m}(X^*(m) - x^*)\}$ of Equation (8) has an asymptotic normal distribution with mean zero and variance* $\mathrm{Var}[\ y(x^*;\omega)\ ]\ /\ [g'(x^*)]^2$.

Lemmas 3 and 4 show that a solution sequence $\{X^*(m)\}$ converges to x^* w.p.1 under certain conditions on the function g and the sample-path approximation $\bar{y}_m$. Both lemmas relax the assumption of unbiasedness and monotonicity of $\bar{y}_m(x;\underline{\omega})$, with Lemma 3 requiring only uniform convergence of $\bar{y}_m(x;\underline{\omega})$ to $g(x)$ w.p.1 as $m \to \infty$ and with Lemma 4 yet more relaxed with only point-wise convergence of $\bar{y}_m(x;\underline{\omega})$ to $g(x)$ w.p.1 as $m \to \infty$ but requiring $\bar{y}_m(\cdot;\underline{\omega})$ to cross γ no more than once. (That consistency is a relaxation of unbiasedness follows from the strong law of large numbers, Billingsley 1979, p. 70. Dropping the unbiasedness assumption allows, for example, the initial transient of steady-state simulation.) In addition, both lemmas allow the sample-path approximation $\bar{y}_m(\cdot;\underline{\omega})$ to have multiple crossing roots for finite values of m. Lemma 3 considers a monotonic function g. We list the proof in Appendix.

Lemma 3 *Assume that*

1. *the function $g : \Re \to \Re$ is a nondecreasing function with unique root x^* satisfying $g(x^*) = \gamma$, and*
2. *the sample-path approximation $\bar{y}_m(x;\ \underline{\omega})$ converges to $g(x)$ uniformly in x w.p.1 as $m \to \infty$.*

Then, for any selection rule defining the solution sequence $\{X^*(m)\}$ from $\{C_m(\underline{\omega})\}$,

$$\lim_{m\to\infty} X^*(m) = x^* \quad \text{w.p.1.}$$

Lemma 4 considers a function g that has a unique root but that is not necessarily monotonic. It assumes that $\bar{y}_m(\cdot;\underline{\omega})$ is pointwise convergent to g and that its crossing set, if not empty, is a unique interval. That is, for every value of m the crossing set $C_m(\underline{\omega})$ of Equation (8), if not empty, is an interval w.p.1, i.e.,

$$\Pr\{\ \underline{\omega}:\ C_m(\underline{\omega}) = [\ x^L(\underline{\omega}),\ x^U(\underline{\omega})\ ]\ \} = 1,$$

where $x^L(\underline{\omega}) = \sup\{\, x : \bar{y}_m(x; \underline{\omega}) < \gamma \,\}$ and $x^U(\underline{\omega}) = \inf\{\, x : \bar{y}_m(x; \underline{\omega}) > \gamma \,\}$. If $x^L(\underline{\omega}) > x^U(\underline{\omega})$, then $C_m(\underline{\omega})$ is empty. For example, $x^L(\underline{\omega}) = x_4$ and $x^U(\underline{\omega}) = x_5$ in Figure 1(b). This condition restricts the function $\bar{y}_m(\cdot; \underline{\omega})$ to cross the target value γ at most once, but allows multiple contiguous roots. In addition to Figure 1(b), examples include the step functions that arise in the two examples of Sections 1, 2 and 5 and any function $\bar{y}_m(x; \underline{\omega})$ monotonic in x. Like Lemma 3, Lemma 4 does not require an unbiased $\bar{y}_m(x; \underline{\omega})$. The proof is listed in Appendix.

Lemma 4 *Assume that*

1. *the function $g : \Re \to \Re$ has a unique root x^* and satisfies*

$$g(x) \begin{cases} > \gamma & \text{if} \quad x > x^* \\ = \gamma & \text{if} \quad x = x^* \\ < \gamma & \text{if} \quad x < x^* \end{cases},$$

2. *for every $x \in \Re$, the estimate $\bar{y}_m(x; \underline{\omega})$ converges to $g(x)$ w.p.1 as $m \to \infty$, and*
3. *for every positive integer m, w.p.1 the crossing set $C_m(\underline{\omega})$ of the function $\bar{y}_m(x; \underline{\omega})$ is either empty or an interval.*

Then, for any selection rule defining the solution sequence $\{X^*(m)\}$ from $\{C_m(\underline{\omega})\}$,

$$\lim_{m \to \infty} X^*(m) = x^* \quad \text{w.p.1.}$$

Lemmas 3 and 4 are stated for functions g that are increasing in the neighborhood of the root x^*. Trivial changes, e.g. redefining g as $-g$, allow them to be stated for functions g that are decreasing near x^*.

Lemma 5 states that, despite allowing an error $\epsilon(m)$ in finding the root of the sample-path equation, a single iteration of an RA algorithm will converge as m goes to infinity.

Lemma 5 *Assume that conditions in Lemma 1, 3, or 4 hold. If $X(m)$ is obtained by one RA iteration and if $\epsilon(m)$ converges to 0 w.p.1 as m goes to infinity, then*

$$\lim_{m \to \infty} X(m) \;=\; x^* \quad \text{w.p.1.}$$

Proof: By Lemma 1, 3, or 4, $X^*(m)$ converges to x^* w.p.1. Therefore, w.p.1 there exists an $N(\underline{\omega})$ such that for every $m > N(\underline{\omega})$ the sample-path equation has a crossing root. When there is a crossing root, the root-finding method returns a solution $X(m)$ that satisfies $|X(m) - X^*(m)| < \epsilon(m)$. Because $\epsilon(m)$ converges to zero w.p.1 as m goes to infinity, the random absolute numerical error $|X(m) - X^*(m)|$ converges to 0 w.p.1. Then Lemma 1, 3 or 4 implies that $X(m)$ converges to x^* w.p.1.

3.2 Case (ii): Solving a Sequence of Sample-Path Equations

We now show that RA algorithms, defined in Section 2.2, converge w.p.1. Recall that RA algorithms solve a sequence of equations of the form (3), for $i = 1, 2,...$, for the retrospective roots $\{x^*(\underline{\omega}_i)\}$, using an increasing sample-size sequence $\{m_i\}$. At each iteration i, RA returns a solution $x(\underline{\omega}_1, ..., \underline{\omega}_i)$, an approximation of $x^*(\underline{\omega}_i)$, within error tolerance ϵ_i. The dependently seeded RA, DRA, repeats m_{i-1} observations $\underline{\omega}_{i-1}$ in $\underline{\omega}_i$ while in IRA all retrospective roots are independent. After i iterations, the DRA root estimator of the root x^* is $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) = x(\underline{\omega}_1, ..., \underline{\omega}_i)$, the last retrospective solution; the IRA root estimator is $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) = \sum_{j=1}^{i} m_j x(\underline{\omega}_1, ..., \underline{\omega}_j) \,/\, \sum_{j=1}^{i} m_j$, a weighted average of solutions $x(\underline{\omega}_1), ..., x(\underline{\omega}_1, ..., \underline{\omega}_i)$ where each weight is proportional to the sample size.

Theorems 1 and 2 respectively show that DRA and IRA algorithms converge to the solution of an SRFP if its function g and associated estimator $\bar{y}_m$ are well behaved.

Theorem 1 *Let a specific DRA algorithm be used to find the unique root x^* of the equation $g(x) = \gamma$ using the estimator $\bar{y}_m(x; \underline{\omega})$ for $g(x)$. If g and $\bar{y}_m(\cdot; \underline{\omega})$ satisfy the conditions in Lemma 1, 3, or 4, then the DRA root estimator $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)$ converges to x^* w.p.1 as $i \to \infty$.*

Proof: The proof proceeds sequentially in three parts: (1) $\lim_{i\to\infty} x^*(\underline{\omega}_i) = x^*$ w.p.1; (2) $\lim_{i\to\infty} x(\underline{\omega}_1, ..., \underline{\omega}_i) = x^*$ w.p.1; (3) $\lim_{i\to\infty} \bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) = x^*$ w.p.1.

The sample-size sequence $\{m_i\}$ is increasing, so "$i \to \infty$" implies "$m_i \to \infty$"; therefore, Lemma 1, 3, or 4 yields the first part. The second part follows because, by definition of RA algorithms, the sequence of error tolerance $\{\epsilon_i\}$ converging to zero w.p.1 implies convergence using Lemma 5. The third part is trivial because $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) = x(\underline{\omega}_1, ..., \underline{\omega}_i)$ by definition of DRA.

Theorem 2 for IRA is analogous to Theorem 1 for DRA. The new condition—that $\sum_{j=1}^{\infty} \mathrm{E}|x^*(\underline{\omega}_j) - x^*|$ is finite—seems likely to be satisfied in practice. For example, assume that there is a finite constant α such that $\mathrm{E}|x^*(\underline{\omega}_j) - x^*| < \alpha/m_j$ for every j, as would be consistent with Lemma 2. Then $\sum_{j=1}^{\infty} \mathrm{E}|x^*(\underline{\omega}_j) - x^*| = \alpha \sum_{j=1}^{\infty} m_j^{-1}$, which is finite if, for example and as we recommend for other reasons, $m_j = c_1 m_{j-1}$ for some $c_1 > 1$. As further evidence that the condition is weak, notice that the above argument holds even if the assumption is changed to use the squared difference rather than the absolute difference.

IRA convergence differs from DRA convergence in that each retrospective iteration is independent. Define $\bar{\omega} = (\underline{\omega}_1, \underline{\omega}_2, ...)$, the infinite sequence of observed random numbers corresponding to the one realization from the sample space. Convergence w.p.1 in Theorem 2 is with respect to $\bar{\omega}$.

Theorem 2 *Let a specific IRA algorithm be used to find the unique root x^* of the equation $g(x) = \gamma$ using the estimator $\bar{y}_m(x; \underline{\omega})$ for $g(x)$. Assume that $\sum_{j=1}^{\infty} \mathrm{E}|x^*(\underline{\omega}_j) - x^*|$ is finite. Then the IRA root estimator $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)$ converges to x^* w.p.1 as $i \to \infty$.*

Proof: As with Theorem 1, the proof proceeds sequentially in three parts: (1) $\lim_{i\to\infty} x^*(\underline{\omega}_i) = x^*$ w.p.1, (2) $\lim_{i\to\infty} x(\underline{\omega}_1, ..., \underline{\omega}_i) = x^*$ w.p.1, and (3) $\lim_{i\to\infty} \bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) = x^*$ w.p.1.

The first part shows that the retrospective roots converge to the root x^* of g. The finiteness of $\sum_{j=1}^{\infty} \mathrm{E}|x^*(\underline{\omega}_j) - x^*|$ is sufficient, using Result 1.3.5 of Serfling (1980).

The second part shows that retrospective solutions converge to x^*. By definition of RA algorithms, $\epsilon_i(\bar{\omega})$ converges to zero w.p.1 and $|x(\underline{\omega}_1, ..., \underline{\omega}_i) - x^*(\underline{\omega}_i)| < \epsilon_i(\bar{\omega})$ for every i. Hence w.p.1 we have the following property: for every $\varepsilon > 0$ there exists an $\mathcal{I}(\varepsilon, \bar{\omega})$ such that$|x(\underline{\omega}_1, ..., \underline{\omega}_i) - x^*(\underline{\omega}_i)| < \epsilon_i(\bar{\omega}) < \varepsilon$ for every $i > \mathcal{I}(\varepsilon, \bar{\omega})$. Then the first part implies the second part.

The third part shows convergence of IRA algorithms. Let $b_i = \sum_{j=1}^{i} m_j$; then $\lim_{i\to\infty} b_i = \infty$. By definition

$$\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) \;=\; \sum_{j=1}^{i} m_j x(\underline{\omega}_1, ..., \underline{\omega}_j) \,/\, \sum_{j=1}^{i} m_j \;=\; b_i^{-1} \sum_{j=1}^{i} m_j x(\underline{\omega}_1, ..., \underline{\omega}_j).$$

By the Toeplitz Lemma (Loéve, 1977, p. 250), the event $\lim_{i\to\infty} x(\underline{\omega}_1, ..., \underline{\omega}_i) = x^*$ implies

the event

$$\lim_{i\to\infty} \bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i) = x^*.$$

Therefore, convergence of $x(\underline{\omega}_1, ..., \underline{\omega}_i)$ to x^* w.p.1 implies convergence of $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)$ to x^* w.p.1.

4 IMPLEMENTATION OF RA ALGORITHMS

We discuss here two issues associated with implementing RA algorithms: choice of a specific RA algorithm and choice of independent random variables ω. Although all RA algorithms converge under weak conditions, as shown in the previous section, computational effort to obtain roots to a specified precision depends upon the specific algorithm used. Similarly, computational effort depends upon the user's definition of the random variables ω; our definition of ω based on random numbers is a safe, but not necessarily efficient, default. We discuss these choices in the next two subsections, respectively. Despite sometimes giving quite specific implementation suggestions, our intention here is only to provide a sense of the issues.

4.1 Choice of a Specific RA Algorithm

Specifying an RA algorithm requires choosing three components: a rule for increasing the sample sizes $\{m_i\}$, a rule for decreasing the error bounds $\{\epsilon_i\}$, and a method for solving the sample-path equations. We discuss each, as well as stopping rules for the algorithm as a whole.

The rule to determine the sample-size sequence $\{m_i\}$ can take many forms. A reasonable family of sequences to consider is m_i being the integer part of $a + c_1(m_{i-1})^b$ for $a \geq 0$, $b \geq 1$, and $c_1 \geq 1$. If, as we argued earlier, $\text{Var}(x^*(\underline{\omega}_i))$ / $\text{Var}(x^*(\underline{\omega}_{i-1})) = m_{i-1}/m_i$, setting $a = 0$ and $b = 1$ is required to make the variance ratio independent of iteration number i. Therefore, $m_i = c_1 m_{i-1}$, with $c_1 > 1$, seems natural. Chen (1994) showed experimentally that computational performance is robust to choices of c_1 over the set $\{1.5, 2, 5, 10\}$. Smaller values of c_1 provide more times at which to stop the algorithm. To cleanly obtain only integer values of m_i, we typically use $c_1 = 2$. The remaining issue is the choice of the initial sample size m_1. Any small value, including $m_1 = 1$, is fine; some numerical root-finding methods might be able to use standard-error information about

$\bar{y}_{m_1}(x;\underline{\omega}_1)$, in which case a larger value of m_1 might be useful. But, m_1 should be small because on the first iteration nothing is known about the location of the retrospective root $x^*(\underline{\omega}_1)$, causing the root-finding method to examine many points x, with each examination requiring m_1 observations for $\bar{y}_{m_1}(x)$. The goal is that m_i is small when many points are examined in the early iterations, and that m_i grows large in later iterations when very few points need to be examined because the previous iterations provide a good initial guess $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1})$ of $x^*(\underline{\omega}_i)$.

The rule to determine the decreasing error-tolerance sequence $\{\epsilon_i\}$ also can take many forms. If the sample sizes $\{m_i\}$ increase by a factor of c_1, then our earlier assumptions about variance decreasing with sample size suggest the form $\epsilon_i = c_1^{-1/2}\epsilon_{i-1}$. The problem is then to specify an appropriate ϵ_1. Chosen too small, the root-finding method wastes computation finding a solution $x(\underline{\omega}_1, ..., \underline{\omega}_i)$ close to $x^*(\underline{\omega}_i)$, which might not be close to x^* because of small sample size. Because we are assuming that no prior information is available about sampling error, a value of ϵ_1 scaled to the problem at hand is not known. In our implementation discussed below and used in Section 5.1, we have used a very large value of ϵ_1 (such as 10^{50}), which eliminates the need to select a problem-dependent value; the error-tolerance logic is then reduced to being a device to prove convergence.

We allow random sequences of error tolerances to facilitate future development of RA algorithms whose ϵ_i at each retrospective iteration is based on an estimate of the standard error of $x^*(\underline{\omega}_i)$. Such algorithms would use a random sequence $\{\epsilon_i\}$, each a factor ϑ, say, of the standard-error estimate of $x^*(\underline{\omega}_i)$. This standard error could be estimated based on Equation (7) for DRA and Equation (6) for IRA, or the asymptotic formula in Lemma 2. Choosing $\vartheta \in [.1, 1.]$ seems reasonable; less than one-tenth of a standard error is certainly too much precision and more than one standard error introduces a substantial new source of error.

The deterministic root-finding method for solving Equations (3) for $i = 1, 2, ...$ can be either analytical or numerical. Analytical approaches require a known and simpler structure of Equation (3); Healy (1992) investigates optimization problems for which an analytical solution can be obtained. Our statement of the SRFP includes no structural information; in this black-box context numerical methods must be used. The advantage is that no analyst effort is required to estimate a root; the disadvantage is that the computational effort is greater than if Equation (3) could be solved analytically. Various numerical search methods are easily implemented, are reasonably efficient, and provide error bounds. Well-known examples include bisection and modified regula

falsi (also called the modified false-position method), as discussed, for example, in Conte and de Boor (1980). Such methods iterate from a starting point, which for RA algorithms would be $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1})$. They also require initially bounding the solution $x^*(\underline{\omega}_i)$, which can be done by searching points in increments or decrements of δ_i. If $x^*(\underline{\omega}_i)$ were known, the optimal δ_i would be $|\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1}) - x^*(\underline{\omega}_i)|$. Because $|\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1}) - x^*(\underline{\omega}_i)|$ and $[\mathrm{Var}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1}) - x^*(\underline{\omega}_i))]^{1/2}$ are proportional, we could choose $\delta_i = c_2[\mathrm{Var}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1}) - x^*(\underline{\omega}_i))]^{1/2}$, where $c_2 > 0$, for $i > 1$. Chen (1994) shows that RA is robust with respect to the choices of c_2 and δ_1; the empirical results favor a small δ_1. Larger values of c_2 accelerate the bound search but result in a bigger bounding interval. Setting $c_2 = 1$ seems reasonable. Using the assumptions for variances and the additional assumption $x^*(\underline{\omega}_i) = x(\underline{\omega}_1, ..., \underline{\omega}_i)$, then $\mathrm{Var}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1}) - x^*(\underline{\omega}_i))$ equals to $\nu^2(m_{i-1}^{-1} - m_i^{-1})$ for DRA and $\nu^2([\sum_{j=1}^{i-1} m_j]^{-1} + m_i^{-1})$ for IRA and hence can be estimated based on Equation (7) or (6). The chosen root-finding method must be implemented to handle multiple roots of $\bar{y}_{m_i}(.; \underline{\omega}_i)$ and to detect the difficult situation of no root in the range of the computer arithmetic.

The RA algorithms defined in Section 2.2 do not have stopping rules, which are irrelevant to our proofs of convergence (unlike the ϵ_i stopping rules for the numerical search during each retrospective iteration i.) Nevertheless, all but interactive implementations must automate stopping. Most natural stopping rules center on a standard-error estimate for the current root estimator, such as Equation (6) for IRA and Equation (7) for DRA. When solving problems iteratively with $c_1 = 2$, each iteration takes about twice as long as the previous iteration, so by retrospective iteration eight (or so) the results appear on the screen slowly enough to read in real time. Deciding when to stop is then easy. To obtain an automated stopping rule, after each retrospective iteration a standard-error estimate can be computed and the algorithm stopped when the estimate is less than a user-specified precision. Because the standard-error estimate can be misleading when the number of retrospective iterations is small, we might require at least four (or so) iterations to avoid premature stopping.

4.2 Choice of ω

In Section 2 we define ω as pseudorandom numbers. This is a natural and safe choice in simulation. Depending on the model, however, ω can be chosen differently for greater computational efficiency, while not affecting the solution returned by the RA algorithm. Alternative choices of ω must be functionally independent of x and ideally include more of the computations needed for generating

the observations used in $\bar{y}_m$. Whatever the choice, the sample $\{\omega_1, ..., \omega_m\}$ needs to be generated only once for $\bar{y}_m(x; \underline{\omega})$ to be computed at any number of points x during the process of solving the sample-path equation.

Consider again the example from Section 1, the analogy of Student's t distribution. There are five possible choices for each component ω in $\underline{\omega} = (\omega_1, ..., \omega_m)$. In order of increasing efficiency, they are (1) the random number seed, (2) the random numbers used to generate values for $V_1, ..., V_n$, (3) the random variates $V_1, ..., V_n$, (4) the pair of statistics $(\bar{V}, S)$, and (5) the statistic T. Because $\underline{\omega}$ needs to be computed only once for solving the sample-path equation, the higher-level definitions of ω lead to less total computing time. Of course, the values of ω need to be stored from the computation at the first x value for use at later x values.

The choice of ω affects the computing time of IRA and DRA. A high-level choice reduces computing time for both, but the ratio of computing times for IRA and DRA increases as the choice of ω becomes more efficient. Whatever the choice, if the sample-size rule is $m_i = c_1 m_{i-1}$, the ratio lies in the interval $[1, u)$, where $u = m_i/(m_i - m_{i-1}) = c_1/(c_1 - 1)$ is the rate at which the retrospective samples grow (as discussed in Section 4.1). The argument is straightforward. Of the four RA algorithm steps (Section 2.2), only the first two involve significant computing time. First consider Step 2, which solves the deterministic root-finding problem. Because at each retrospective iteration i the sample size m_i is the same for both DRA and IRA, the time for Step 2 is essentially identical, although usually non-trivial. Now consider Step 1, which generates the $\underline{\omega}$ values. At retrospective iteration i, IRA must generate m_i new values but DRA has the opportunity to save and reuse $m_{i-1} = m_i/c_1$ previous values. Therefore, if storing and saving the previous values is essentially free, the Step-1 computation time for IRA is about u times that for DRA. Thus, the ratio is approximately 1 if the Step-2 computation time dominates and approximately u if the Step-1 time dominates. Therefore, in practice the ratio lies in the interval $[1, u)$, completing the argument.

A particularly important case is when ω is only the initial random-number seed, in which case the ratio is one, as illustrated in our empirical results in Table 1. An efficient choice of ω moves computation from Step 2 to Step 1, and therefore increases the ratio, but not beyond u. Although inefficient in our example and often elsewhere, choosing the random-number seed for ω often simplifies the programming necessary to create the procedure $\bar{y}_m$. Storing a more-complex ω

in a discrete-event simulation would be difficult and the bookkeeping necessary to store and retrieve the values might become non-trivial, since the order in which the values are used would change from one x value to the next. The argument of the previous paragraph would then be invalid.

5 APPLICATION AND EMPIRICAL RESULTS

To illustrate SRFPs and RA algorithms, we now discuss the guaranteed-coverage tolerance interval (GCTI) problem that motivates our research interest. Numerical results are provided in Section 5.1. Thiokol Corporation asked us to develop an algorithm to determine, in real time, the constant x^* satisfying the nonnormal α-coverage γ-confidence tolerance-interval relationship

$$\Pr_{\bar{W},S} \{ \Pr_W \{W \geq \bar{W} - x^* \, S\} \geq \alpha \} = \gamma. \tag{12}$$

Here the product characteristic W is a continuous random variable with distribution function F_W having known shape but unknown mean and variance. (For example, possibly W is normally distributed with unknown mean and variance.) The sample mean $\bar{W}$ and sample standard deviation S are computed from product characteristics $W_1, W_2, ..., W_n$ previously generated from F_W independent of each other and of W. Given sample size n, coverage α, confidence γ, and distribution shape, the problem is to determine the value of x^* so that with $100\gamma\%$ confidence the random tolerance interval $[\bar{W} - x^* S, \infty)$ contains at least the proportion α of the distribution. The Thiokol application is to reliability design issues, but such non-normal tolerance-interval problems arise in many contexts (Chen and Schmeiser 1995 and Chen and Yang 1999).

For this application, the root-finding function is $g(x) = \Pr_{\bar{W},S} \{ \Pr_W \{W \geq \bar{W} - x \, S\} \geq \alpha \}$, an $(n+1)$-dimensional integral. Numerical integration would be inefficient even for small sample size n. The function $g(x)$ can be estimated easily, however, by $\bar{y}_m(x)$, the sample average of m realizations of the conditional random variable

$$Y(x) = \begin{cases} 1 & \text{if } \Pr_W \{W \geq \bar{W} - x \, S | \bar{W}, S\} \geq \alpha \\ 0 & \text{otherwise} \end{cases}.$$

Notice that the random variable $Y(x)$ is not a function of mean $\mathrm{E}(W)$ or variance $\mathrm{Var}(W)$ because

the random probability $\Pr_W\{W \geq \bar{W} - xS \mid \bar{W}, S\}$ does not depend on $\mathrm{E}(W)$ or $\mathrm{Var}(W)$. Hence the reliability $\Pr_W \{W \geq \bar{W} - x\ S \mid \bar{W}, S\}$ can be computed using arbitrary values of $\mathrm{E}(W)$ and $\mathrm{Var}(W) > 0$, given $\bar{W}$ and S based on a sample with these arbitrary values of the mean and variance. The Monte Carlo computer procedure for generating an observation of $Y(x)$ then consists of four steps: (1) generate a sample $W_1, W_2, ..., W_n$ from F_W with arbitrarily chosen mean and variance, (2) compute $\bar{W}$ and S from the sample, (3) compute $p = \Pr_W \{W \geq \bar{W} - x\ S | \bar{W}, S\} = 1 - F_W(\bar{W} - x\ S)$, and (4) set $Y(x)$ equal to 0 if p is less than α and to 1 otherwise.

As discussed in Section 4.2, to implement RA the independent variable ω needs to be chosen. Various choices are possible for this SRFP. As always, the pseudorandom number seed can be used, and all computations repeated for each value of x: compute the pseudorandom numbers, transform them to random variates $W_j, j = 1, ..., n$, compute $\bar{W}$ and S, compute the reliability p, and compute $y(x)$. Choices of ω, in order of increasing efficiency, are (1) the pseudorandom number seed, (2) the pseudorandom numbers, (3) $(W_1, ..., W_n)$, and (4) $(\bar{W}, S)$.

Chen and Schmeiser (1995) analyze the Thiokol SRFP in some detail. They develop an efficient Monte Carlo algorithm to solve the problem by viewing it as a quantile-estimation problem. As expected, the special-purpose quantile-estimation algorithm is more efficient than black-box RA algorithms, which have no information about the problem's structure. Here, however, we use this example to compare the efficiency of DRA and IRA algorithms in Section 5.1 and to compare IRA to a tuned version of classical stochastic approximation in Section 5.2.

5.1 Comparing DRA and IRA

Using a GCTI application, we illustrate the statistical and computational efficiency of specific DRA and IRA algorithms with a Monte Carlo experiment. For this application, IRA converges faster than DRA, the estimators of sampling error work reasonably well after the first few iterations, and saving the random numbers rather than only the seed value reduces computational effort by about one third for DRA and about one fifth for IRA.

Following the discussion in Section 4.1, we choose the following RA components:

- Sample-size sequence: $m_1 = 2$ and $m_{i+1} = c_1 m_i$, where $c_1 = 2$. With these choices, each new retrospective iteration uses a sample size almost equal to the total used by all previous

iterations.

- Error-tolerance sequence: $\epsilon_1 = 10^{50}$ and $\epsilon_{i+1} = c_1^{-1/2}\epsilon_i$. That is, we essentially remove the error-tolerance logic.

- A numerical root-finding method that returns a retrospective solution, an approximation of the root of Equation (3): The initial solution is, rather arbitrarily, $x_0 = 1$. At retrospective iteration i, $\bar{y}_{m_i}$ is computed at the current root estimate $\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1})$ for $i > 1$ and at x_0 for $i = 1$, which yields either an upper or lower bound. The other bound is found by searching either left or right at a distance δ_i. This distance is doubled until the retrospective root is bounded. The sequence $\{\delta_i\}$ defined by $\delta_1 = .0001$ and $\delta_i = c_2[\widehat{\text{Var}}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_{i-1}) - x^*(\underline{\omega}_i))]^{1/2}$, except when this value is zero, in which case $\delta_i = \delta_{i-1}$. See Section 4.1, where $c_2 = 1$ is suggested. Once the retrospective root is bounded, regula falsi search is used to find a retrospective solution within the error tolerance ϵ_i. Because of the large error tolerance values, the logic reduces to two steps: (a) bounding the retrospective solution and (b) returning the linear interpolate of the bounds as the retrospective solution. We refer to this variation as the Bounding RA algorithm.

We intentionally did not tune the algorithm parameters x_0, m_1, δ_1, c_1, and c_2 to this application. As discussed in Chen (1994), IRA performance is robust to these parameter values, both in that these values work well over many applications and in that small changes in the values have little effect on performance.

The application is the GCTI problem with $n = 10$, $\alpha = \gamma = .99$, and Johnson distribution (Johnson 1949) with skewness 4 and kurtosis 30. The true root is tolerance factor $x^* \approx 1.938$ (see Chen and Schmeiser 1995, Table 2).

Table 1 compares DRA and IRA performance based on 20 independent runs of 1000 independent Monte Carlo replications. Each replication solves the GCTI application once with $\bar{\omega} = (\underline{\omega}_1, \underline{\omega}_2, ...)$ generated independently of other replications. All digits shown are statistically significant, except possibly the last digits of cpu times. For iterations $i = 1, ..., 10$, the quality of the solutions are measured by the squared bias $[\text{E}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i)) - x^*]^2$, variance $\text{Var}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i))$, and mean square error (squared bias plus variance, denoted by MSE), shown in columns two through four, respectively. The fifth column shows $\text{E}[\widehat{\text{Var}}(\bar{x}_i)]$, the mean of the DRA and IRA variance estimators

$\widehat{\mathrm{Var}}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i))$ from Equations (7) and (6), respectively.

Table 1 shows that IRA produces better solutions than DRA for every retrospective iteration after the first, where the two algorithms are identical. For both DRA and IRA the squared biases quickly become negligible compared to the variance. The IRA bias accumulates the biases from past iterations and hence is bigger than, but decreases at about the same rate as, the DRA bias. Because the biases are small, the variances and mse's are nearly identical after the first few retrospective iterations. For both DRA and IRA these decrease by about 50% with each retrospective iteration, as is expected because $c_1 = 2$ and therefore the sample sizes are doubling.

The final few iterations suggest that asymptotically IRA has mse and variance that is 50% to 60% that of DRA. This seems reasonable because IRA has the benefit of using twice (or $c_1/(c_1 - 1)$ in general) as many independent observations as DRA (m_i for DRA and $\sum_{j=1}^{i} m_j$ for IRA).

The averages of the variance estimators $\widehat{\mathrm{Var}}(\bar{x}(\underline{\omega}_1, ..., \underline{\omega}_i))$ from Equations (7) and (6) are shown in the fifth column. Although these estimators correctly sense the order of magnitude of the sampling error, they underestimate the variance from the third column. The relative error is less for DRA than for IRA and both relative errors are decreasing with retrospective iteration number i. The biases in the variance estimators are caused by the early iterations not having the asymptotic behavior assumed in the derivations of Equations (7) and (6) due to the arbitrary starting point $x_0 = 1$. Maybe better variance estimators can be found, perhaps by taking a moving average of only the previous few iterations. The current estimators work well at their primary purpose, which is to predict the proper scaling (via δ_{i+1}) for the next retrospective iteration.

The two right-most columns show the number of cpu seconds required for 1000 Monte Carlo replications on a Sun SparcCenter 1000 computer. The times almost double with each retrospective iterations because the sample sizes are doubling ($c_1 = 2$). The times to generate the random numbers, transform them to random variates, and compute the observations y quickly dominate the fixed cost of each iteration.

Times for four RA algorithms, two versions of both DRA and IRA, are shown. Both versions, which correspond to implementations using two different ω's, create the same realizations and therefore have the same statistical properties; they differ only in the method of computing. The columns labeled *R.N.* correspond to an implementation in which the random numbers are stored when first generated. The columns labeled *Seed* correspond to storing only the initial random-

number seed, from which random numbers are recomputed as needed. The latter is simpler to code, primarily because it requires a fixed amount of storage. The R.N. versions are faster, but negligibly so for IRA. The decrease in DRA times is greater than for IRA because in DRA each random number is used again and again, whereas in IRA each random number is used in only one retrospective iteration. The ratio of the IRA time to the DRA time is about 1 for the Seed versions and 1.2 for the R.N. versions. These results are consistent with our argument in Section 4.2 that the ratio lies in $[1, u)$, where $u = 2$ here. The speed improvement could differ considerably with yet another choice of ω (e.g., storing the random samples or, better, only the sample means and standard deviations) or in another application.

A natural way to compare algorithm performance is via the generalized mse, the product of cpu time and mse. In this sense, IRA dominates DRA. The IRA mse is only half of the DRA mse, while the cpu-time ratio (IRA to DRA) is no larger than 2 for any choices of ω. Overall, IRA/R.N. has the best generalized mse. IRA/Seed, which is only a bit slower, is likely to be a better choice in practice because it is easier to implement and requires finite storage, qualities not measured in generalized mse. If the algorithms are to be run for many iterations, R.N. versions would eventually require disk (rather than random-access memory) storage, and the elapsed times would quickly be longer than Seed versions.

5.2 Comparing IRA and Stochastic Approximation

We now compare an IRA algorithm to a stochastic approximation (SA) algorithm; in the GCTI application IRA has much smaller sampling error than SA. As shown in Figure 2, RA algorithms are substantially more efficient than Robbins and Monro's SA, the classical black-box solution method, even when it is tuned to the application.

Figure 2 compares IRA/Seed and SA using the GCTI application with tolerance parameters $n = 5$, $\alpha = .5$, $\gamma = .9$, and normal population, for which $x^* = .6857$. The IRA/Seed algorithm used here is the same as that in Table 1 except that the initial point x_0 is generated randomly from the normal distribution with mean x^* and variance 10^4 (denoted by $N(x^*, 10^4)$). The early convergence rate of SA depends on the initial point and the sample size (for example, Chen (1994, p. 71) and Fu and Healy (1992)), although the asymptotic convergence rate does not. We generate the initial point from $N(x^*, 1)$ and define each SA iteration to use the average of 5 observations,

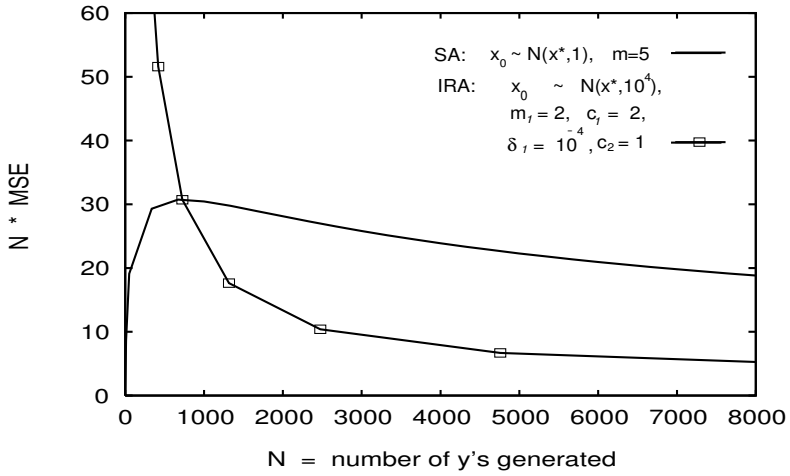

Figure 2: Comparison of IRA and SA Algorithms

which is the (empirically determined) optimal value associated with this random initial point (see Chen 1994, p. 71). The performance measure here is again generalized mse, except that now computing effort is measured by N, the total number of $y(x)$ observations computed. On a Sun SparcStation 2 computer, $N = 8000$ corresponds to about 30 seconds of cpu time. Figure 2 shows that IRA/Seed, despite having less information about the location of the root, and not being tuned to the application, converges faster than SA for these numbers of observations. The sample size N needed for the SA generalized mse to drop below that of IRA is, as can be inferred from the figure, so large that in practice IRA performance is substantially better.

That the SA asymptotic generalized mse is less might seem surprising. The asymptotic generalized mse ratio (IRA to SA) for this Thiokol example is about 2.3, which is quite close to the asymptotic number of points at which IRA evaluates $\bar{y}_{m_i}(x)$ at each retrospective iteration i (Chen 1994). Therefore, the asymptotic performance seems to be the same if we count sample-path approximations for IRA and function evaluations for SA. This makes sense because asymptotically the function g is linear and the slope at the root is known, so more than one function evaluation is wasted. But the additional function evaluations are, in fact, quite useful early, when g is not linear, maybe explaining why IRA performs better than SA in practice.

6 CONCLUSIONS

We introduce a family of *retrospective approximation* (RA) algorithms to solve SRFPs. The algorithms are based on solving sample-path approximations to the problem of interest; that is, pseudorandom data from the problem are generated and used to create a sequence of approximate problems, which have increasing sample sizes and decreasing solution-error tolerances. Our algorithms differ from retrospective optimization algorithms in that they are iterative and in that they explicitly allow some error in the solutions to the approximate problems. The latter is necessary if the approximate problems are to be solved numerically.

In addition to introducing retrospective approximation algorithms, we prove convergence for one-dimensional SRFPs, we discuss implementation issues and specify two algorithm variations: *dependent retrospective approximation* (DRA), which uses all past observations at each iteration, and *independent retrospective approximation* (IRA), which uses each observation in only one iteration. Monte Carlo results for an application, as well as some analytical arguments, indicate that IRA is superior to DRA. Monte Carlo results of a related application show that IRA has smaller generalized mean squared error than a version of stochastic approximation that is tuned to the application.

Recommendations for future work in this area include: (1) proposing specific RA algorithms and proving convergence for multi-dimensional SRFPs; (2) deriving asymptotic distributions for the root estimator; and (3) extending RA's application on SRFPs to more general optimization problems.

ACKNOWLEDGMENTS

This research is supported by Purdue Research Foundation Grant 690-1287-2104, Thiokol Corporation Contract A46111430, and NSF Grant DMS 93-00058. We thank Colm O'Cinneide for helpful discussions, Anton Kleywegt for helpful comments, and *IIE Transactions* editor Jim Wilson for thoughtful suggestions that substantially improved the presentation.

REFERENCES

Andradóttir, S. (1992). An Empirical Comparison of Stochastic Approximation Methods for Simulation Optimization. *Proceedings of the First Industrial Engineering Research Conference*, ed. Klutke, G., Mitta, D. A., Nnaji, B. O., and Seiford, L. M., 471–475. Norcross, Georgia: Institute of Industrial Engineers.

Banks, J., J. S. Carson, II, and B. L. Nelson (1996). *Discrete-Event System Simulation*, Upper Saddle River, NJ: Prentice Hall.

Billingsley, P. (1979). *Probability and Measure*, New York, NY: John Wiley & Sons.

Chen, H. (1994). *Stochastic Root Finding in System Design,* Ph.D. Dissertation, School of Industrial Engineering, Purdue University, West Lafayette, Indiana.

Chen, H.-S. (1998). *Multi-dimensional Independent Retrospective Approximation for Making Resource Allocation Decisions in Manufacturing Systems,* Master Thesis, Department of Industrial Engineering, Da-Yeh University, Chang-Hwa, TAIWAN. Chinese.

Chen, H. and B. W. Schmeiser (1994a). Stochastic Root Finding: Problem Definition, Examples, and Algorithms. *Proceedings of the Third Industrial Engineering Research Conference,* ed. L. Burke and J. Jackman, 605–610. Norcross, Georgia: Institute of Industrial Engineers.

Chen, H. and B. W. Schmeiser (1994b). Retrospective Approximation Algorithms for Stochastic Root Finding. *Proceedings of the 1994 Winter Simulation Conference,* ed. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 255–261. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Chen, H. and B. W. Schmeiser (1995). Monte Carlo Estimation for Guaranteed-Coverage Nonnormal Tolerance Intervals. *Journal of Statistical Computation and Simulation* **51**, 223–238.

Chen, H. and T.-K. Yang (1999). Computation of the Sample Size and Coverage for Guaranteed-Coverage Nonnormal Tolerance Intervals. *Journal of Statistical Computation and Simulation* **63**, 299–320.

Conte, S. D. and C. de Boor (1980). *Elementary Numerical Analysis: An Algorithmic Approach,* New York, NY: McGraw-Hill, Inc.

Fu, M. C. (1994). Optimization via Simulation: A Review. *Annals of Operations Research* **53**, 199–247.

Fu, M. C. and K. J. Healy (1992). Simulation Optimization of (s, S) Inventory Systems. *Proceedings of the 1992 Winter Simulation Conference,* ed. J.J. Swain, D. Goldsman, R.C. Crain, and J.R. Wilson, 506–514. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Fu, M. C. and S. D. Hill (1997). Optimization of Discrete Event System via Simultaneous Perturbation. *IIE Transactions* **29**, 233–243.

Gürkan, G., A. Y. Özge, and S. M. Robinson (1994). Sample-Path Optimization in Simulation. *Proceedings of the 1994 Winter Simulation Conference,* ed. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 247–254. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Healy, K. J. (1992). *Optimizing Stochastic Systems: A Retrospective/Deterministic Approach*, Ph.D. Dissertation, Department of Operations Research, Cornell University, Ithaca, NY.

Healy, K. J. and L. W. Schruben (1991). Retrospective Simulation Response Optimization. *Proceedings of the 1991 Winter Simulation Conference,* ed. B.L. Nelson, W.D. Kelton, and G.M. Clark, 954–957. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Healy, K. J. and Y. Xu (1994). Simulation Based Retrospective Approaches to Stochastic System Optimization. *Research Memorandum No. 94-14, School of Industrial Engineering, Purdue University, West Lafayette, IN.*

Healy, K. J. and Y. Xu (1995). Simulation Optimization of a Bulk Server. *Research Memorandum No. 95-3, School of Industrial Engineering, Purdue University, West Lafayette, IN.*

Homem-de-Mello, T., A. Shapiro, and M. L. Spearman (1999). Finding Optimal Material Release Times Using Simulation Based Optimization. *Management Science* **45**, 86–102.

Huber, P. J. (1964). Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics* **35**, 73-101.

Johnson, N. L. (1949). Systems of Frequency Curves Generated by Methods of Translation. *Biometrika* **36**, 149–176.

Law, A. M. and W. D. Kelton (2000). *Simulation Modeling and Analysis,* 3rd ed., New York, NY: McGraw-Hill, Inc.

L'Ecuyer, P., N. Giroux and P. W. Glynn (1994). Stochastic Optimization by Simulation: Numerical Experiments with the $M/M/1$ Queue in Steady State. *Management Science* **40**, 1245–1261.

L'Ecuyer, P. and P. W. Glynn (1994). Stochastic Optimization by Simulation: Convergence Proofs

for the $GI/G/1$ Queue in Steady State. *Management Science* **40**, 1562–1578.

Loéve, M. (1977). *Probability Theory I*, New York, NY: Springer-Verlag.

Plambeck, E. L., B.-R. Fu, S. M. Robinson, and R. Suri (1996). Sample-Path Optimization of Convex Stochastic Performance Functions. *Mathematical Programming* **75**, 137-176.

Robbins, H. and S. Monro (1951). A Stochastic Approximation Method. *Annals of Mathematical Statistics* **22**, 400–407.

Robinson, S. M. (1996). Analysis of Sample-Path Optimization. *Mathematics of Operations Research* **21**, 513–528.

Rubinstein, R. Y. and A. Shapiro (1993). *Discrete Event Systems—Sensitivity Analysis and Stochastic Optimization by the Score Function Method,* New York, NY: John Wiley & Sons.

Serfling, R. J. (1980). *Approximation Theorems of Mathematical Statistics,* New York, NY: John Wiley & Sons.

Shapiro, A. (1996). Simulation-Based Optimization—Convergence Analysis and Statistical Inference. *Communications in Statistics—Stochastic Models* **12**, 425–454.

Shapiro, A. and Y. Wardi (1996). Convergence Analysis of Stochastic Algorithms. *Mathematics of Operations Research* **21**, 615–628.

Shapiro, A. and T. Homem-de-Mello (1997). A Simulation-Based Approach to Two-Stage Stochastic Programming with Recourse. *Mathematical Programming* **81**, 301–325.

Simon, B. (1998). Convergence of a Stochastic Rootfinding Procedure. *Working Paper, Department of Mathematics, University of Colorado at Denver, Denver, CO.*

Appendix: Proofs of Lemmas 3 and 4

Proof of Lemma 3: Without loss of generality, assume that $\gamma = 0$. By the second assumption, for every $\varepsilon > 0$, w.p.1 there exists a positive integer $N(\varepsilon, \underline{\omega})$ such that

$$m > N(\varepsilon, \underline{\omega}) \;\text{ implies }\; g(x) - \varepsilon \;<\; \bar{y}_m(x; \underline{\omega}) \;<\; g(x) + \varepsilon \;\text{ for all } x \in \Re\ .$$

By the first assumption g is a nondecreasing function so the inverse function $g^{-1}(a) = \inf\{\tilde{x} :\ g(\tilde{x}) \geq a\}$ is also nondecreasing. Now choose a particular positive value of ε—small enough that both

$g^{-1}(-\varepsilon)$ and $g^{-1}(\varepsilon)$ are defined. Because $\varepsilon > 0$ and x^* is the unique root, $g^{-1}(-\varepsilon) \le x^* \le g^{-1}(\varepsilon)$.

Now, consider the region of the crossing set. Because "$x > g^{-1}(\varepsilon)$" implies "$g(x) \ge \varepsilon$", then w.p.1 for every $m > N(\varepsilon, \underline{\omega})$,

$$\bar{y}_m(x; \underline{\omega}) \;>\; g(x) - \varepsilon \;\ge\; \varepsilon - \varepsilon \;=\; 0, \quad \text{for every } x > g^{-1}(\varepsilon).$$

Similarly, w.p.1 for every $m > N(\varepsilon, \underline{\omega})$,

$$\bar{y}_m(x; \underline{\omega}) \;<\; g(x) + \varepsilon \;\le\; -\varepsilon + \varepsilon = 0 \quad \text{for every } x < g^{-1}(-\varepsilon)\ .$$

Therefore, for every $m > N(\varepsilon, \underline{\omega})$, the crossing set $C_m(\underline{\omega})$ is not empty and satisfies

$$C_m(\underline{\omega}) \;\subseteq\; [\, g^{-1}(-\varepsilon),\; g^{-1}(\varepsilon)\,] \;\text{ w.p.1.}$$

To prove that the crossing set $C_m(\underline{\omega})$ is nonempty, we let

$$x^+ \equiv \inf\{x : \bar{y}_m(z; \underline{\omega}) > 0 \;\text{ for every }\; z > x\} \tag{13}$$

and

$$x^- \equiv \sup\{x : \bar{y}_m(z; \underline{\omega}) < 0 \;\text{ for every }\; z < x\}. \tag{14}$$

Since $\bar{y}_m(z; \underline{\omega}) > 0$ for every $z > g^{-1}(\varepsilon)$ and $\bar{y}_m(z; \underline{\omega}) < 0$ for every $z < g^{-1}(-\varepsilon)$, we see that

$$x^+ \le g^{-1}(\varepsilon) \quad \text{and} \quad x^- \ge g^{-1}(-\varepsilon). \tag{15}$$

Next we prove that

$$x^- \le x^+ \tag{16}$$

by contradiction. Assume that $x^+ < x^-$. For every $x \in (x^+, x^-)$, we can pick $z \in (x^+, x)$ arbitrarily; and applying the definition (13) of x^+, we must have $\bar{y}_m(x; \underline{\omega}) > 0$ since $x > z > x^+$. Similarly we can pick $z' \in (x, x^-)$ arbitrarily; and applying the definition (14) of x^-, we must have $\bar{y}_m(x; \underline{\omega}) < 0$ since $x < z' < x^-$. This is a contradiction since $\bar{y}_m(x; \underline{\omega})$ cannot be both positive and negative; and thus the assumption that $x^+ < x^-$ must be false. This establishes the inequality (16).

Now if $x^- = x^+$, then we see that for every neighborhood $\mathcal{N}(x^+) = \mathcal{N}(x^-)$ of the point $x^+ = x^-$, we can find points z and z' satisfying

$$\left.\begin{array}{c} z > x^+ \ \text{ and } \ z \in \mathcal{N}(x^+) \ \text{ so that } \ \bar{y}_m(z;\underline{\omega}) > 0 \\ z' < x^- \ \text{ and } \ z' \in \mathcal{N}(x^-) = \mathcal{N}(x^+) \ \text{ so that } \ \bar{y}_m(z';\underline{\omega}) < 0 \end{array}\right\}; \tag{17}$$

and it follows immediately that $x^- = x^+$ is on the boundary of R^N and on the boundary of R^P. Thus we see that in this case, $x^- = x^+ \in R^Z \cup R^{NP} = C_m(\underline{\omega})$.

If $x^- < x^+$, then both x^- and x^+ are crossing roots. Here we prove for x^- only; the proof for x^+ is similar. For every neighborhood $\mathcal{N}(x^-)$ of x^-, we can find points z and z' satisfying

$$\left.\begin{array}{c} z > x^- \ \text{ and } \ z \in \mathcal{N}(x^-) \ \text{ so that } \ \bar{y}_m(z;\underline{\omega}) \geq 0 \\ z' < x^- \ \text{ and } \ z' \in \mathcal{N}(x^-) \ \text{ so that } \ \bar{y}_m(z';\underline{\omega}) < 0 \end{array}\right\}; \tag{18}$$

otherwise, $\sup\{x : \bar{y}_m(z;\underline{\omega}) < 0 \ \text{ for every } \ z < x\} > x^-$ and this contradicts the definition of x^-. There are now three cases to consider:

(a) We have $\bar{y}_m(x^-;\underline{\omega}) > 0$ so that $x^- \in R^P$; and it follows from (18) that x^- is on the boundary of R^N or R^Z so that $x^- \in R^{NP}$.

(b) We have $\bar{y}_m(x^-;\underline{\omega}) = 0$; and in this case we have immediately that $x^- \in R^Z$.

(c) We have $\bar{y}_m(x^-;\underline{\omega}) < 0$ so that $x^- \in R^N$; and it follows from (18) that x^- is on the boundary of R^P or R^Z so that $x^- \in R^{NP}$.

Therefore, x^- is a crossing root. Using a similar argument, we can show that x^+ is also a crossing root. We can summarize all of these results as follows:

$$\left.\begin{array}{c} x^- = x^+ \ \text{ implies that } \ x^- = x^+ \in R^Z \cup R^{NP} = C_m(\underline{\omega}) \\ x^- < x^+ \ \text{ implies that } \ C_m(\underline{\omega}) \subset [x^-, x^+] \end{array}\right\}. \tag{19}$$

Combining (16) and (19), we see that $C_m(\underline{\omega})$ must be nonempty.

Let $X^*(m)$ be any point in $C_m(\underline{\omega})$. Because x^* is in the interval $[\, g^{-1}(-\varepsilon),\, g^{-1}(\varepsilon)\,]$, for every $m > N(\varepsilon, \underline{\omega})$

$$|X^*(m) - x^*| \leq g^{-1}(\varepsilon) - g^{-1}(-\varepsilon) \ \text{ w.p.1.}$$

Since ε can be arbitrarily small and x^* is the unique root, for any selection rule defining the solution sequence $\{X^*(m)\}$,

$$\lim_{m\to\infty} X^*(m) = x^*, \quad \text{w.p.1.}$$

Proof of Lemma 4: Without loss of generality, assume $\gamma = 0$. By the second assumption, for every $x \in \Re$ and every $\eta > 0$, w.p.1 there exists a positive integer $N(x, \eta, \underline{\omega})$ such that

$$m > N(x, \eta, \underline{\omega}) \text{ implies } |\bar{y}_m(x; \underline{\omega}) - g(x)| < \eta.$$

Given any $\varepsilon > 0$, the first assumption implies

$$g(x^* + \varepsilon) > 0 \quad \text{and} \quad g(x^* - \varepsilon) < 0 \ .$$

Now, choose $x = x^* + \varepsilon$ and $\eta = g(x^* + \varepsilon)$, and define $N_1(x^*, \varepsilon, \underline{\omega}) = N(x, \eta, \underline{\omega})$. Then for $m > N_1(x^*, \varepsilon, \underline{\omega})$

$$|\bar{y}_m(x^* + \varepsilon; \underline{\omega}) - g(x^* + \varepsilon)| < g(x^* + \varepsilon) \text{ w.p.1.}$$

Therefore, w.p.1, $\bar{y}_m(x^* + \varepsilon; \underline{\omega}) > 0$ for every $m > N_1(x^*, \varepsilon, \underline{\omega})$. Similarly, choose $x = x^* - \varepsilon$ and $\eta = -g(x^* - \varepsilon) > 0$, and define $N_2(x^*, \varepsilon, \underline{\omega}) = N(x, \eta, \underline{\omega})$. Then, for $m > N_2(x^*, \varepsilon, \underline{\omega})$,

$$|\bar{y}_m(x^* - \varepsilon; \underline{\omega}) - g(x^* - \varepsilon)| < -g(x^* - \varepsilon) \text{ w.p.1.}$$

Therefore, w.p.1, $\bar{y}_m(x^* - \varepsilon; \underline{\omega}) < 0$ for every $m > N_2(x^*, \varepsilon, \underline{\omega})$.
Define $N_3(x^*, \varepsilon, \underline{\omega}) = \max\{N_1(x^*, \varepsilon, \underline{\omega}), N_2(x^*, \varepsilon, \underline{\omega})\}$. Then, for every $m > N_3(x^*, \varepsilon, \underline{\omega})$

$$\bar{y}_m(x^* - \varepsilon; \underline{\omega}) < 0 < \bar{y}_m(x^* + \varepsilon; \underline{\omega}) \text{ w.p.1.}$$

Therefore, w.p.1 there exists at least one crossing root in $[x^* - \varepsilon, x^* + \varepsilon]$ and hence the crossing set $C_m(\underline{\omega})$ is not empty. (The argument is similar to that in the proof of Lemma 3.) Furthermore, by the third assumption, $C_m(\underline{\omega}) = [\, x^L(\underline{\omega}),\, x^U(\underline{\omega})\,]$ for every m, where $x^L = \sup\{\, x : \bar{y}_m(x; \underline{\omega}) < \gamma \,\}$ and $x^U = \inf\{\, x : \bar{y}_m(x; \underline{\omega}) > \gamma \,\}$. Hence, for every $m > N_3(x^*, \varepsilon, \underline{\omega})$

$$x^* - \varepsilon \le x^L \le x^U \le x^* + \varepsilon \text{ w.p.1,}$$

and therefore $C_m(\underline{\omega}) \subseteq [\, x^* - \varepsilon,\, x^* + \varepsilon\,]$ w.p.1. Because $X^*(m)$ is a point selected from $C_m(\underline{\omega})$, then w.p.1 every $X^*(m) \in [\, x^* - \varepsilon,\, x^* + \varepsilon\,]$ for $m > N_3(x^*, \varepsilon, \underline{\omega})$. Hence, for any selection rule defining the solution sequence $\{X^*(m)\}$,

$$\lim_{m \to \infty} X^*(m) = x^* \quad \text{w.p.1.}$$

Biographies

Huifen Chen is an Associate Professor in the Department of Industrial Engineering at Chung Yuan Christian University in Taiwan. She completed her Ph.D. in the School of Industrial Engineering at Purdue University in August 1994. She received a B.S. degree in accounting from National Cheng-Kung University in Taiwan in 1986 and an M.S. degree in statistics from Purdue University in 1990. Her current research focuses on stochastic root finding, random-vector generation, non-normal tolerance intervals, and stochastic operations research applied in reliability and transportation.

Bruce Schmeiser is a professor in the School of Industrial Engineering at Purdue University. He received his Ph.D. from the School of Industrial and Systems Engineering at Georgia Tech in 1975; his undergraduate degree in the mathematical sciences and master's degree in industrial engineering are from The University of Iowa. His research interests include stochastic root finding, simulation output analysis, input modeling, random-variate generation, variance-reduction techniques, Markov chain Monte Carlo methods, and applied operations research. He has served in a variety of roles for the IIE, INFORMS and the Winter Simulation Conference.

Table 1: DRA and IRA comparison for a GCTI problem

	Squared Bias		Variance		MSE		$\widehat{\mathrm{E}[\mathrm{Var}(\bar{x}_i)]}$		CPU Time (sec./1000 repl.)			
									DRA		IRA	
i	DRA	IRA	DRA	IRA	DRA	IRA	DRA	IRA	R.N.	Seed	R.N.	Seed
1	.42	.42	.17	.17	.59	.59	—	—	2	3	2	3
2	.26	.27	.14	.09	.40	.36	.17	.075	6	9	6	9
3	.10	.13	.15	.08	.25	.21	.15	.054	8	12	8	12
4	.03	.05	.15	.07	.18	.12	.11	.040	12	18	13	18
5	.00	.01	.14	.06	.14	.07	.08	.030	19	27	20	32
6	.00	.00	.11	.04	.11	.04	.06	.021	33	46	40	52
7	.000	.000	.046	.024	.046	.024	.027	.012	61	87	75	96
8	.000	.000	.022	.012	.022	.012	.015	.007	121	168	143	178
9	.000	.000	.010	.006	.010	.006	.008	.004	227	322	278	345
10	.000	.000	.005	.003	.005	.003	.004	.002	435	660	540	670