Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

RETROSPECTIVE APPROXIMATION ALGORITHMS FOR STOCHASTIC ROOT FINDING

Huifen Chen

Department of Industrial Engineering
Da-Yeh Institute of Technology
Chang-Hwa, TAIWAN, 51505

Bruce W. Schmeiser

School of Industrial Engineering
Purdue University
West Lafayette, Indiana 47907-1287, U.S.A.

ABSTRACT

The stochastic root-finding problem is to find the root of the equation $g(x) = \gamma$, where $g(x)$ can be estimated. There are many applications, including continuous and convex stochastic optimization, which is the problem of finding the zero of the gradient function. We propose a family of retrospective approximation algorithms that numerically solve a sequence of sample-path equations with increasing sample sizes. Algorithms in the family differ by the choice of several parameters including the deterministic root-finding method, sample sizes, the stopping rule of the numerical search, the point estimator, and the stopping rule of the entire algorithm. Under weak conditions, retrospective approximation converges. We also propose a simple version of the family: bounding retrospective approximation. General-use algorithm parameter values are suggested. In our empirical comparison with the classical approach of stochastic approximation, bounding retrospective approximation is more efficient and less sensitive to parameter values.

1 INTRODUCTION

We consider the stochastic root-finding problem (SRFP), which is to find the root x^* of the equation $g(x) = \gamma$, using only an unbiased estimator $Y(x)$ of $g(x)$. Such problems arise in stochastic systems design: x is the design parameter, $g(x)$ is the corresponding system performance, $Y(x)$ is the estimated performance obtained from a (simulation) experiment, and γ is the desired system performance. We want to find the design value x^* that gives the system performance γ. Chen and Schmeiser (1994) discuss examples of the SRFP, as well as solution approaches.

To be more specific, we define the

Stochastic Root Finding Problem (SRFP):

Given:

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

(b) a (computer) procedure for generating an observation of the d-dimensional $Y(x)$ for any $x \in \Re^d$,

find the root x^* satisfying $g(x^*) = \gamma$ using only the estimator Y, where $g : \Re^d \to \Re^d$, and $g(x) = E[Y(x)]$ for any $x \in \Re^d$.

We are interested in algorithms that do not use the structure of the root-finding function g, but rather only the computer procedure in (b). Evaluating g numerically may be possible, but computationally expensive. The computer procedure, however, conducts a simulation experiment that mimics the stochastic system to obtain an unbiased estimate of g. Although SRFPs are defined in multiple dimensions, we consider here only problems in one dimension, i.e., $d = 1$.

There are two approaches to solving the SRFP—prospective and retrospective. A *prospective* algorithm explores the set of feasible solutions x to look for the root x^*, much like a prospector searches for gold: a spot is chosen for exploration because good results are foreseen. A classical prospective algorithm is stochastic approximation, first proposed by Robbins and Monro (1951). (See Section 4.1.) Chen and Schmeiser (1994) propose a more-general framework of prospective algorithms that iteratively update root candidates based on the entire history. They discuss three specific approaches within the framework.

Rather than looking to the future, a *retrospective* algorithm looks to the past. A sample-path approximation to the real problem is generated, and the root to this existing problem is sought. As its sample size grows larger, the sample-path approximation is more precise and hence its root produces an approximation to the true root x^*.

We propose a subclass of retrospective algorithms, called *retrospective approximation (RA)*, which iteratively solve a sequence of sample-path approximation problems with increasing sample sizes. A solution satisfying an error tolerance is obtained for each sample-path approximation; the root estimator is then a function of those solutions. Under proper conditions, this estimator converges to the true root.

The organization of this paper is as follows. Section 2 defines the family of retrospective approximation algorithms. Related literature and RA's convergence are discussed. In Section 3, we propose a simple version of the family, *bounding* retrospective approximation. In Section 4, we empirically compare the efficiencies of our bounding retrospective approximation and Robbins and Monro's stochastic approximation algorithms.

2 RETROSPECTIVE APPROXIMATION

Here we propose a family of retrospective approximation algorithms for the SRFP: finding x^* satisfying $g(x^*) = \gamma$ using no more than a computer program that gives an unbiased estimate $y(x)$ of the system performance $g(x)$. We define RA in Section 2.1, which is followed by the review of related literature in Section 2.2. In Section 2.3, we discuss RA's consistency.

2.1 Definition of RA

RA iteratively solves a sequence of sample-path approximation problems. Solutions of these approximation problems are computed by a deterministic root-finding method. The root estimator is then a function of those solutions. Before stating the RA logic, we discuss the sample-path approximation.

Let $F_x(y) = \Pr\{Y(x) \le y\}$, the distribution function of $Y(x)$. Since $Y(x)$ is an unbiased estimator of $g(x)$, the root-finding equation $g(x^*) = \gamma$ can be rewritten as

$$\int y(x^*) \; dF_{x^*}(y) = \gamma. \tag{1}$$

Since the left-hand side of Equation (1) is unknown, a sample-path approximation of Equation (1) is the sample-path equation

$$\sum_{j=1}^{m} y_j(X^*)/m = \gamma, \tag{2}$$

where the root X^* is random, each $y_j(x)$ is generated from the distribution F_x, and typically the m observations are independent. How to solve Equation (2) may not be obvious because each $y_j(X^*)$ is generated from the distribution F_{X^*}, with unknown X^*, and many candidates x of X^* may be considered.

Hence, we consider a distribution function $F(\cdot)$ that defines $Y(x)$ and is independent of x. If ω is an observation from the distribution F, then $y(x)$ is a function of x and ω, say $y(x,\omega)$. For a fixed ω, the function $y(x,\omega)$ is a deterministic function of x. Let Ω be the support of the distribution function $F(\cdot)$. Then the root-finding equation $g(x^*) = \gamma$ is equivalent to Equation

$$(P^*) \qquad \int_{\Omega} y(x^*,\omega) \; dF(\omega) = \gamma.$$

Based on a random sample $\underline{\omega} = \{\omega_1, ..., \omega_m\}$ generated from the distribution F, the sample-path equation equivalent to Equation (2) is then

$$(P) \qquad \bar{y}(\, X^*, \, \underline{\omega}\,) = \gamma,$$

where $\bar{y}(x, \underline{\omega}) = \sum_{j=1}^{m} y(x,\omega_j)/m$. Using this fixed $\{\omega_1, ..., \omega_m\}$ for different values of x, the sample mean $\bar{y}(x,\underline{\omega})$ is a deterministic function of x. Then, to solve Equation (2) for X^*, we merely solve Equation (P) via a deterministic root-finding method. Notice that the random root X^* may not exist or may not be unique for a finite sample size.

The choice of F, which is the user's, affects RA efficiency. The least efficient choice is to let the ω's be the initial random-number seed, which is always independent of x. The most efficient choice is to include in F all random values that are not a function of x. The efficiency of RA depends on the choice of F because to solve the sample-path equation, the sample $\underline{\omega}$ is generated only once but $\bar{y}(x,\underline{\omega})$ may be computed for many x's. (See Chen, 1994, p. 34.)

RA, stated below, iteratively solves a sequence of sample-path equations

$$(P_i) \qquad \bar{y}(X_i^*, \underline{\omega}_i) = \gamma, \qquad i = 1, 2, ...,$$

where $\bar{y}(x,\underline{\omega}_i) = \sum_{j=1}^{m_i} y(x,\omega_j)/m_i$, and $\underline{\omega}_i = \{\omega_1, ..., \omega_{m_i}\}$ is generated independently for each i. RA uses a strictly increasing sample-size sequence $\{m_i\}$. At each iteration, RA finds a solution X_i satisfying $|X_i - X_i^*| < \epsilon_i$, where $\epsilon_i \in [0, \infty]$ is a specified constant. We call X_i^* the i^{th} retrospective root, X_i the i^{th} retrospective solution, and i the retrospective iteration number. The point estimator $\bar{X}_i$ after i retrospective iterations is a function $h(\cdot)$ of the past trajectory of solutions $X_1, ..., X_i$. More specifically, RA algorithms work as follows.

RA Algorithms:
Given:

1. method for solving Equations $\{(P_i)\}$,
2. initial sample size m_1 and rule for successively increasing m_i, for $i = 2, 3, ...$,
3. rule for computing error tolerance $\{\epsilon_i\}$,
4. functional form of h such that $\bar{x}_i = h(x_1, ..., x_i)$,
5. stopping rule of the entire algorithm.

Find: the root x^* using:

0. Initialize $i = 0$.
1. Let $i \longleftarrow i + 1$.
2. Independently generate $\underline{\omega}_i = \{\omega_1, ..., \omega_{m_i}\}$.
3. Solve Equation (P_i): $\bar{y}(x_i^*, \underline{\omega}_i) = \gamma$ for x_i^* by the given deterministic method. Return x_i, if x_i satisfies $|x_i - x_i^*| < \epsilon_i$.
4. Compute $\bar{x}_i = h(x_1, ..., x_i)$ and its variance estimate.
5. Stop? If yes, return $\bar{x}_i$. Otherwise, compute m_{i+1} and go to Step 1.

We call this algorithm RA (retrospective approximation) because each sample-path equation (P_i) retrospectively approximates the root-finding equation (P^*) and is solved only approximately. Chen (1994, Section 5.2.1) discusses the setting of the five given algorithm parameter values.

2.2 Related Literature

Literature related to RA includes M-estimators and retrospective approaches for stochastic optimization. We briefly discuss both. See Chen (1994, Section 5.2.2) for more discussion.

The retrospective root X_i^* satisfying Equation (P_i) is an M-estimator, as proposed by Huber (1964). M-estimators are used with finite samples to estimate a distribution property, e.g., the location or scale parameter. The estimate $y(x, \omega)$ is chosen so that $y(x^*, \omega)$ satisfies the root-finding equation (P^*), and a better choice of y leads to a more robust M-estimator. In the SRFP, however, $y(x, \omega)$ is a user-provided sampling routine that is made available to the RA algorithm. Convergence of RA requires the convergence of X_i^* (Section 2.3). Serfling (1980) discusses the consistency of the M-estimator.

A problem related to the SRFP is stochastic optimization, finding the optimal point x^* of an objective function that can only be estimated. Healy and Schruben (1991), Robinson (1994), and Rubinstein and Shapiro (1993) discuss retrospective methods for stochastic optimization.

The emphasis of both the M-estimator and the stochastic-optimization literature is on sample-path behavior as m increases, an important component of our algorithms where multiple sample paths are considered with increasing values of m. They assume, however, that the sample-path equations can be solved exactly, whereas we assume that the sample-path roots need to be approximated because the sample-path function is intractable.

2.3 Consistency of RA

Chen (1994, Section 5.2.3) shows that RA algorithms converge under mild regularity conditions. Recall that RA estimates the root x^* of the equation $g(x^*) = \gamma$ by solving a sequence of sample-path equations. We consider here two cases: (i) RA solving only a single and long sample-path equation, and (ii) RA solving a sequence of equations $\{(P_i)\}_{i=1}^{\infty}$.

In Case (i), RA solves only one sample-path equation $(P) : \bar{y}(X^*, \underline{\omega}) = \gamma$. The root estimator X is computed within ϵ precision of the retrospective root X^*. Since X is an approximation of X^*, convergence of X includes two parts: (1) X^* converges to x^* with probability one (w.p.1), and then (2) X converges to x^* w.p.1. If $\bar{y}(x, \underline{\omega})$ is a consistent estimate of $g(x)$ and other conditions hold, then the sample-path function converges to g as m goes to infinity. Hence, the retrospective root X^* converges to x^* and yields the first convergence. The second convergence requires that the absolute numerical error $|X - X^*|$ goes to zero, i.e., the deterministic root-finding method guarantees convergence and the error tolerance goes to zero.

In Case (ii), RA solves a sequence of equations $\{(P_i)\}$. The corresponding sample-size sequence $\{m_i\}$ is an increasing sequence. For each Equation (P_i), RA finds an approximation X_i of X_i^* within error tolerance ϵ_i. At the i^{th} retrospective iteration, the root estimator $\bar{X}_i$ is a function of all past solutions $X_1, ..., X_i$. Assume that $\bar{X}_i = \sum_{j=1}^{i} m_j X_j \,/\, \sum_{j=1}^{i} m_j$, a weighted average of $X_1, ..., X_i$ where each weight is proportional to the sample size. Then, if the convergence conditions for Case (i) holds, Chen (1994, Section 5.2.3) shows that $\bar{X}_i$ converges to x^* as i goes to infinity.

3 BOUNDING RA

We propose here a family of bounding retrospective approximation (Bounding RA) algorithms, which is a simple version of RA with good empirical performance. As mentioned in Section 2.1, RA has five types of algorithm parameters. Although RA converges within a wide range of parameter values, the real-time efficiency depends on the parameter set-

tings. Our goal is a ready-to-use RA algorithm that has no "magic parameters," parameters whose values strongly affect algorithm efficiency. Chen (1994, Chapter 6) develops a probability model for parameter-setting guidelines, which suggests Bounding RA algorithms.

Bounding RA sets the five types of parameter values as follows: (i) a numerical algorithm that stops when bounds of the retrospective root x_i^* are found, (ii) a geometric sequence of sample sizes $\{m_i\}$: $m_i = c_1 m_{i-1}$, where $c_1 > 1$, (iii) a retrospective solution x_i, obtained by interpolating the bounds, (iv) an iterative root estimate $\bar{x}_i = \sum_{j=1}^{i} m_j x_j \,/\, \sum_{j=1}^{i} m_j$, and (v) an estimate of $\text{var}(\bar{x}_i)$, stated below, to help users decide the stopping point. The name *Bounding* RA comes from parameter types (i) and (iii), which together find bounds and then return the linear interpolation point of the bounds. We list here the general Bounding RA algorithm.

Bounding RA Algorithms:

Given: default values of x_0, a positive integer m_1, $c_1 > 1$, $\delta_1 > 0$, and $c_2 > 0$.

Find: the root x^* using:

0. Initialize $i = 0$.
1. Let $i \longleftarrow i + 1$.
2. Independently generate $\underline{\omega}_i = \{\omega_1, ..., \omega_{m_i}\}$.
3. Numerically solve the sample-path equation $(P_i) : \bar{y}(x_i^*, \underline{\omega}_i) = \gamma$ for x_i^* as follows:
   - 3.1 Let $x = \bar{x}_{i-1}$, where $\bar{x}_0 = x_0$.
   - 3.2 If $i > 1$, compute
   $$\delta_i = c_2 \, \hat{\sigma} \sqrt{\frac{1}{\sum_{j=1}^{i-1} m_j} + \frac{1}{m_i}} \,,$$
   where $\hat{\sigma}$ is from Step 5 of the $(i-1)^{\text{th}}$ iteration.
   - 3.3 Find a bounding interval $[x_{i,l}, x_{i,u}]$ that brackets x_i^*:
     - (a) Initialize $x_{i,l} = x$, $x_{i,u} = x$, $\bar{y}_{i,l} = \gamma + 1$, and $\bar{y}_{i,u} = \gamma - 1$.
     - (b) Compute $\bar{y}(x, \underline{\omega}_i)$.
     - (c) Update upper bound or lower bound:
       If $\bar{y}(x, \underline{\omega}_i) < \gamma$, let $x_{i,l} = x$ and $\bar{y}_{i,l} = \bar{y}(x, \underline{\omega}_i)$.
       Otherwise, let $x_{i,u} = x$ and $\bar{y}_{i,u} = \bar{y}(x, \underline{\omega}_i)$.
     - (d) If the bounds are not found, update the iterate and step size:
       If $(\bar{y}_{i,l} - \gamma)(\bar{y}_{i,u} - \gamma) > 0$, let
       $$x \longleftarrow \begin{cases} x + \delta_i & \text{if } \bar{y}(x, \underline{\omega}_i) < \gamma \\ x - \delta_i & \text{otherwise} \end{cases},$$
       $$\delta_i \longleftarrow 2\,\delta_i,$$
       and go to Step 3.3 (b).
   - 3.4 Compute the linear interpolation:
   $$x_i = x_{i,l} + (x_{i,u} - x_{i,l})(\gamma - \bar{y}_{i,l}) / (\bar{y}_{i,u} - \bar{y}_{i,l}).$$
4. Compute the root estimate $\bar{x}_i = \sum_{j=1}^{i} m_j x_j \,/\, \sum_{j=1}^{i} m_j$.
5. Compute $\widehat{\text{Var}}(\bar{x}_i) = \hat{\sigma}^2 \,/\, \sum_{j=1}^{i} m_j$, where
   $$\hat{\sigma} = \sqrt{m_i}\,\left(\frac{x_{i,u} - x_{i,l}}{\bar{y}_{i,u} - \bar{y}_{i,l}}\right)\left\{\frac{(\bar{y}_{i,u} - \gamma)\, s_{i,l}}{\bar{y}_{i,u} - \bar{y}_{i,l}} + \frac{(\gamma - \bar{y}_{i,l})\, s_{i,u}}{\bar{y}_{i,u} - \bar{y}_{i,l}}\right\},$$
   $s_{i,u}^2 = [m_i(m_i - 1)]^{-1} \sum_{j=1}^{m_i} [y(x_{i,u}, \omega_j) - \bar{y}_{i,u}]^2$, and $s_{i,l}^2 = [m_i(m_i - 1)]^{-1} \sum_{j=1}^{m_i} [y(x_{i,l}, \omega_j) - \bar{y}_{i,l}]^2$.
6. Is $\widehat{\text{Var}}(\bar{x}_i)$ small enough? If yes, return $\bar{x}_i$. Otherwise, compute $m_{i+1} = c_1 m_i$ and go to Step 1.

In our implementation, for reasons discussed in Chen (1994, Section 7.1), we set the initial parameter values to be $x_0 = 1$, $m_1 = 2$, $c_1 = 2$, $\delta_1 = 10^{-4}$, and $c_2 = 1$. A Fortran implementation is in Chen (1994, Appendix B).

Convergence of Bounding RA is not guaranteed because parameter type (iii) results in an infinite error tolerance ϵ_i for $i \geq 1$. Chen (1994, Section 7.2) discusses a modification of Bounding RA that guarantees convergence; this modified algorithm is more complex, however.

The rationale of Bounding RA is our assumption, and the suggestion from empirical evidence, that as the retrospective iteration number i increases, the bounding interval becomes smaller. Since the sample-path function $\bar{y}(\cdot, \underline{\omega}_i)$ is nearly linear in small intervals, one-step linear interpolation is sufficient for the numerical search. Our empirical results show that, for a large i, the numerical search in the modified Bounding RA usually reaches the error tolerance after the first linear interpolation. Hence, we suspect that taking more linear interpolations is not necessary even for the first few retrospective iterations. Therefore, we propose the heuristic Bounding RA algorithm for simplicity. Our empirical results in Section 4 show that Bounding RA performs well in a real-time application.

4 EMPIRICAL COMPARISON

This section shows the empirical comparison of Bounding RA and Robbins and Monro's classical stochastic approximation (CSA), which is discussed

in Section 4.1. The Monte Carlo experiments consider a tolerance-constant SRFP, as described in Section 4.2. We evaluate Bounding RA and CSA based on the finite-sample convergence speed, as discussed in Section 4.3.

Our empirical results show that, despite the lack of guaranteed convergence, Bounding RA converges quickly for a large range of parameter values. CSA guarantees convergence, but the practical convergence speed strongly depends on parameter values.

4.1 Classical Stochastic Approximation

Robbins and Monro (1951) propose classical stochastic approximation (CSA), which iteratively estimates x^* by the formula

$$X_{i+1} = X_i - a_i(\bar{Y}_i - \gamma), \quad i = 0, 1, ...,$$

where X_0 is an arbitrary initial guess of x^*, $\bar{Y}_i = \sum_{j=1}^{m} Y_j(X_i)/m$, $\{Y_1(x), ..., Y_m(x)\}$ given x is a sample from the distribution F_x, and $\{a_i\}_{i=0}^{\infty}$ is a predetermined sequence of positive constants satisfying $\sum_{i=0}^{\infty} a_i = \infty$ and $\lim_{i\to\infty} a_i = 0$. Under weak conditions on g and the Y's, the sequence $\{X_i\}$ converges to x^* with probability one. The step-size sequence $\{a_i^*\}$ minimizing the asymptotic variance is $a_0^* = 1$, and $a_i^* = [i\, g'(x^*)]^{-1}, i = 1, ...$ (Sacks, 1958, Venter, 1967). Then $\sqrt{m}(X_i - x^*)$ is asymptotically normally distributed with mean 0 and variance $\sigma^2 = \mathrm{Var}[Y(x^*)]/[g'(x^*)]^2$. However, a_i^* is usually unknown because x^* and the first derivative g' are unknown. There are several modifications (Andradóttir, 1991, Kushner and Clark, 1978, Ljung, Pflug, and Walk, 1992, Wasan, 1969), but as the modifications become more sophisticated, the algorithms tend to have more parameters. Few guidelines exist for choosing these parameter values. Hence Section 4.3 uses CSA for comparison.

4.2 A Tolerance-Constant Example

The special case of the SRFP that motivated this research is Thiokol Corporation's need to find a guaranteed-coverage tolerance constant x^* for a non-normal continuous distribution F_W with known shape but unknown mean and variance. (See Chen and Schmeiser, 1993, 1994.) Let $W_1, ..., W_n$ be a random sample from the distribution F_W; let $\bar{W}$ and S^2 be the corresponding sample mean and sample variance (with denominator $n - 1$) . The α-coverage and γ-confidence tolerance constant x^* satisfies $g(x^*) = \gamma$, where

$$g(x) = \Pr_{\bar{W},S}\{\ \Pr_W\{W \geq \bar{W} - x\, S\} \geq \alpha\ \}$$

and $\alpha, \gamma \in (0, 1)$. That is, if a tolerance interval $[\bar{W} - x^* S, \infty)$ is built, then with γ confidence the tolerance interval contains population-coverage at least α. Given n, α, γ, and the distribution shape, we want to find x^*. For this root-finding problem, numerical evaluation of $g(x)$, a $(n+1)$-dimensional integral, would be inefficient. Nevertheless, we can estimate $g(x)$ by a (user-provided) unbiased estimator $Y(x) = I\{\ \Pr_W\{W \geq \bar{W} - x\ S\} \geq \alpha\}$, where $I(A) = 1$ if A is true and $I(A) = 0$, otherwise.

4.3 Empirical Results

We empirically compare the convergence speed of CSA and Bounding RA. Our empirical results show that CSA suffers slow convergence for finite sample sizes unless the parameter values are chosen magically. (Also see Fu and Healy, 1992.) Bounding RA seems to converge quickly with the speed $O(N^{-1/2})$ and the asymptotic variance $k\sigma^2$, where N is the cumulative number of estimates y generated, k is the average number of function ($\bar{y}$) evaluations to find bounds for the limiting sample-path equation, and σ^2 is the asymptotic variance of CSA. Although $k \geq 2$, because finding bounds takes at least two function evaluations, Bounding RA usually reaches asymptotic efficiency faster than CSA in our empirical results.

The simulation experiment is designed as follows. The parameter values of the tolerance-constant problem are sample size $n = 5$, minimum coverage $\alpha = .5$, normal population, and confidence $\gamma = .9$, for which $x^* = .68567$ and $\sigma^2 = 1.29$. Furthermore, the (user-provided) random ω is set to random-number seeds, which is the least efficient. (See Chen, 1994, Section 8.1). The two design points for CSA are $x_0 \in \{x^*, N(x^*, 1)\}$ each with the asymptotically optimal step size $\{a_i^*\}$ (available because x^* and g' are known for this experimental example) and the approximately optimal sample size (obtained from experiments). The two design points for Bounding RA are $x_0 \in \{N(x^*, 1), N(x^*, 10^4)\}$ each with the default values of parameters m_1, c_1, δ_1, and c_2.

The performance measure for comparison is "E(work $\times$ error)". We use N to measure the computational work because of ω being random-number seeds; we use "MSE = squared error of the root estimate using N observations" to measure the error. If the algorithm converges, $N \times$ MSE goes to the asymptotic variance as N goes to infinity.

Figure 1 shows the comparison of CSA and Bounding RA with the same initial point distribution, $x_0 \sim N(x^*, 1)$. CSA uses the corresponding approximately optimal sample size $m = 5$ (see Chen, 1994, Figure 8.1), which would not be known in practice. Never-

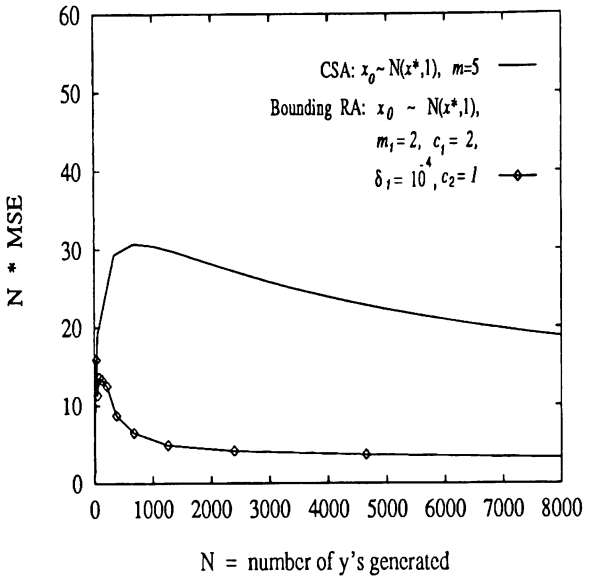

Figure 1: Comparison of CSA and Bounding RA: CSA's Parameter Values Are $x_0 \sim N(x^*, 1)$ and $m = 5$; Bounding RA's Parameter Values Are $x_0 \sim N(x^*, 1)$, $m_1 = 2$, $c_1 = 2$, $\delta_1 = 10^{-4}$, and $c_2 = 1$

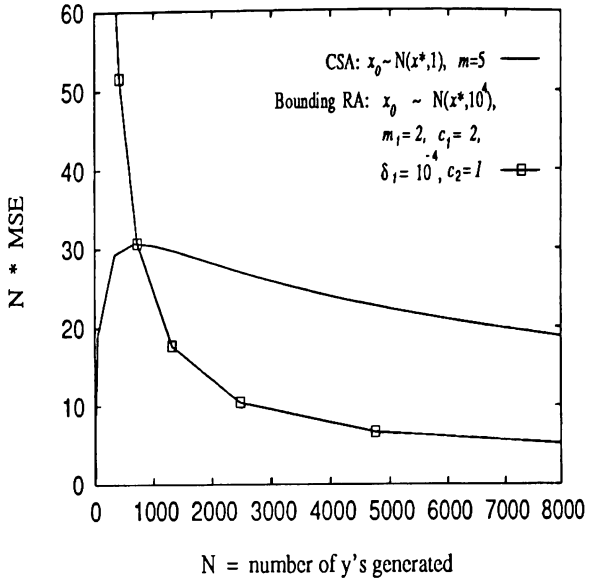

Figure 2: Comparison of CSA and Bounding RA: CSA's Parameter Values Are $x_0 \sim N(x^*, 1)$, and $m = 5$; Bounding RA's Parameter Values Are $x_0 \sim N(x^*, 10^4)$, $m_1 = 2$, $c_1 = 2$, $\delta_1 = 10^{-4}$, and $c_2 = 1$

theless, CSA converges much slower than Bounding RA for these values of N. Recall, however, that CSA is asymptotically more efficient than Bounding RA.

Figure 2 illustrates that Bounding RA is relatively insensitive to the initial point x_0. Here, the initial Bounding RA point is sampled with a standard deviation of 100, yet for $N > 1000$, Bounding RA is more efficient than CSA with optimal sample size $m = 5$ and initial point distribution with a standard deviation of 1.

Figure 3 illustrates how well CSA would work if the root were known in advance and a correspondingly high sample size m were used. The figure shows CSA starting at the true root ($x_0 = x^*$) and using the large sample size $m = 100$ (see Chen, 1994, Figure 8.1). Bounding RA's parameter values are the same as those in Figure 1. We see that CSA converges immediately to its asymptotic variance σ^2 (Recall that $\sigma^2 = 1.29$) when the initial point is the true root x^* and the sample size is large. This magic situation usually does not happen in practice. Figure 3 also shows that Bounding RA has an asymptotic variance a bit more than twice as big as that of CSA. Each iteration of Bounding RA requires a bit more than two function evaluations, because at least two are required to bound the root. Hence, we speculate that if Bounding RA converges, its asymptotic variance is k times as large as CSA's asymptotic variance.

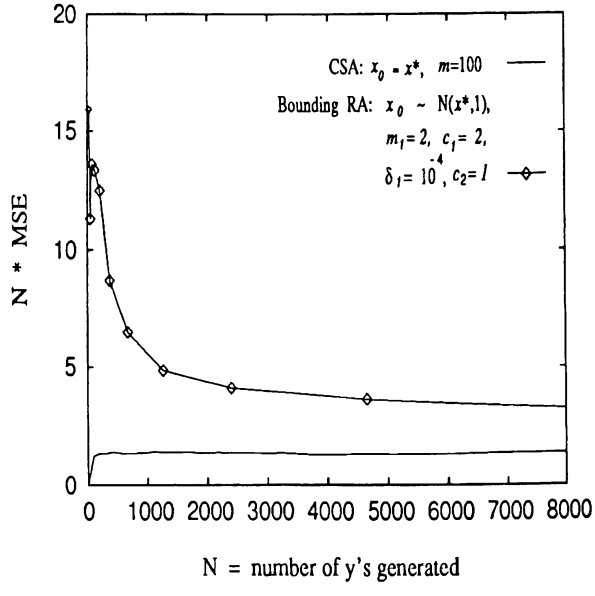

Figure 3: Comparison of CSA and Bounding RA: CSA's Parameter Values Are $x_0 = x^*$, and $m = 100$; Bounding RA's Parameter Values Are $x_0 \sim N(x^*, 1)$, $m_1 = 2$, $c_1 = 2$, $\delta_1 = 10^{-4}$, and $c_2 = 1$

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 and Kevin Healy at Purdue University, Michael R. Taaffe at University of Minnesota, and Sigrún Andradóttir at University of Wisconsin for helpful discussions.

REFERENCES

Andradóttir, S. (1991). A Projected Stochastic Approximation Algorithm. In *Proceedings of the 1991 Winter Simulation Conference,* ed. B.L. Nelson, W.D. Kelton, and G.M. Clark, 954–957.

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

Chen, H. and B. W. Schmeiser (1993). Monte Carlo Estimation for Guaranteed-Coverage Nonnormal Tolerance Intervals. In *Proceedings of the 1993 Winter Simulation Conference,* ed. G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 509–515.

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

Fu, M. C. and K. J. Healy (1992). Simulation Optimization of (s, S) Inventory Systems. In *Proceedings of the 1992 Winter Simulation Conference,* ed. J.J. Swain, D. Goldsman, R.C. Crain, and J.R. Wilson, 506–514.

Healy, K. and L. W. Schruben (1991). Retrospective Simulation Response Optimization. In *Proceedings of the 1991 Winter Simulation Conference,* ed. B.L. Nelson, W.D. Kelton, and G.M. Clark, 954–957.

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

Kushner, H. and D. Clark (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems,* New York: Springer-Verlag.

Ljung, L., G. Pflug, and H. Walk (1992). *Stochastic Approximation and Optimization of Random Systems,* Berlin: Birkhauser Verlag.

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

Robinson, Stephen M. (1994). Analysis of Sample-Path Optimization. In *Proceedings of the 1994 Winter Simulation Conference,* this volume.

Rubinstein, R. Y. and A. Shapiro (1993). *Discrete Event Systems—Sensitivity Analysis and Stochastic Optimization by the Score Function Method,* New York: John Wiley & Sons, Chapter 2.

Sacks, J. (1958). Asymptotic Distributions of Stochastic Approximation Procedures. *Annals of Mathematical Statistics* **29**, 373–405.

Serfling, R. J. (1980). *Approximation Theorems of Mathematical Statistics,* New York: John Wiley & Sons, Chapter 7.

Venter, J. H. (1967). An Extension of the Robbins-Monro Procedure. *Annals of Mathematical Statistics* **38**, 181–190.

Wasan, M. T. (1969). *Stochastic Approximation,* Cambridge, UK: Cambridge University Press.

AUTHOR BIOGRAPHIES

HUIFEN CHEN joined the faculty at Department of Industrial Engineering, Da-Yeh Institute of Technology in August 1994. 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 research interests include simulation methodology applied to quality control, reliability, and stochastic operations research.

BRUCE W. 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 primary research interests are the probabilistic and statistical aspects of simulation. He is an active participant in the Winter Simulation Conference, including being Program Chair in 1983 and chairing the Board of Directors during 1988-1990.