The Stochastic Root-Finding Problem: Overview, Solutions, and Open Questions

RAGHU PASUPATHY, Virginia Tech
SUJIN KIM, National University of Singapore

The stochastic root-finding problem (SRFP) is that of finding the zero(s) of a vector function, that is, solving a nonlinear system of equations when the function is expressed implicitly through a stochastic simulation. SRFPs are equivalently expressed as stochastic fixed-point problems, where the underlying function is expressed implicitly via a noisy simulation. After motivating SRFPs using a few examples, we review available methods to solve such problems on constrained Euclidean spaces. We present the current literature as three broad categories, and detail the basic theoretical results that are currently known in each of the categories. With a view towards helping the practitioner, we discuss specific variations in their implementable form, and provide references to computer code when easily available. Finally, we list a few questions that are worthwhile research pursuits from the standpoint of advancing our knowledge of the theoretical underpinnings and the implementation aspects of solutions to SRFPs.

Categories and Subject Descriptors: G.1.5 [**Numerical Analysis**]: Roots of Nonlinear Equations—*Iterative methods*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Probabilistic algorithms (including Monte Carlo)*; I.6.1 [**Simulation Modeling**]: Simulation Theory

General Terms: Algorithms, Design, Theory

Additional Key Words and Phrases: Stochastic root finding, stochastic approximation, sample-average approximation

ACM Reference Format:
Pasupathy, R. and Kim, S. 2011. The stochastic root-finding problem: Overview, solutions, and open questions. ACM Trans. Model. Comput. Simul. 21, 3, Article 19 (March 2011), 23 pages.
DOI = 10.1145/1921598.1921603 http://doi.acm.org/10.1145/1921598.1921603

1. INTRODUCTION

In this article we consider the stochastic root-finding problem (SRFP)—that of identifying a solution x to the vector equation $g(x) = 0$ ($g : \mathbb{R}^q \to \mathbb{R}^q$ is a vector-valued function) when a stochastic simulation capable of "generating" a consistent estimator of g is all that is available. The reader might recognize SRFPs as the stochastic analogue of the problem of solving a nonlinear system of equations—something that has been investigated in tremendous detail ever since Isaac Newton introduced a method to successively approximate the roots of a polynomial in the mid-seventeenth century. SRFPs, by contrast, have only recently gained attention, as the modeling of complex physical processes on a digital computer has progressively become easier with advances in simulation methodology and software.

R. Pasupathy was supported in part by the Office of Naval Research grants N000140810066 and N000140910997. S. Kim was supported in part through ARF grant R-266-000-049-133.
Contact author's email address: pasupathy@vt.edu
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2011 ACM 1049-3301/2011/03-ART19 $10.00
DOI 10.1145/1921598.1921603 http://doi.acm.org/10.1145/1921598.1921603

Our aim is an exposition of the topic of SRFPs at the level of an "advanced tutorial," with sufficient technical and implementation focus. After providing some context for SRFPs, we present three broad solution categories that are available. We list the major results that are currently known in each category, preceded and followed by intuitive explanations of the crucial ingredients composing the results. We include an implementation section for two of the three categories where we discuss and list algorithm variants in a form that can be readily used, and demonstrate their application on a numerical example. We also subjectively discuss SRFP contexts where such implementations are likely to "work" or "fail". Towards charting directions for future research, we briefly discuss three open questions.

As Michael Fu [1994] notes in his review article on simulation optimization, it is inevitable that a survey article such as this may be seen by some as disproportionately representing certain parts of the literature. This may be a legitimate criticism, but is (only) a consequence of our expertise in particular areas and lack in others. Nevertheless, we have made an earnest attempt to expose all available methods to the reader, often referring the reader to articles that delve deeper into specific aspects that may be of interest. Our guiding philosophy has been to focus on the basic results, provide plenty of references to more nuanced results, and list algorithms with useful subjective discussion.

2. MOTIVATION

The deterministic root-finding problem (DRFP), that is, the question of solving a nonlinear system of equations has been widely studied for the last several decades. As pointed out in the classic references by Rheinboldt and Ortega [Ortega and Rheinboldt 1970; Rheinboldt 1987], DRFPs are ubiquitous, and appear routinely in wide-ranging disciplines including statistics, engineering, biology, medicine, and the social sciences. Such has been its appeal that entire conferences and journals have been devoted to the subject, with the aim of communicating research advances and acquainting applied scientists and practitioners with developments in the field. We have chosen not to discuss the details of DRFPs for the simple reason that several voluminous accounts have already been written, and any attempts on our part will fall well short of these widely available classics. Key references that may serve as starting points for the interested reader include the following: Traub [1964] and Householder [1970] for solving one-dimensional DRFPs; Young [1971] and Stewart [1973] for solving linear DRFPs; and Rheinboldt [1987], Rheinboldt and Ortega [1970], Todd [1976] and Kelly [1995, 2006] for solving general DRFPs.

It may be convincingly argued that the relevance of SRFPs is implied by that of DRFPs. When a DRFP is posed by a practitioner, the function g whose zero is sought is frequently a "closed-form" approximation of some underlying unknown function—a fact that is traditionally not discussed. The practitioner, either for lack of a better alternative or simply because he/she is convinced of the validity of such approximation, is content with the solution of the surrogate problem. Prototypical examples of such abound in contexts where regression is used—see for example Kushner and Yin [2003]. In these scenarios, the practitioner is actually faced with an SRFP, but chooses, albeit implicitly, to solve it as a DRFP!

Whereas appearance of SRFPs in contexts such as the above is not explicit, SRFPs have recently been recognized much more explicitly. Practitioners are now routinely constructing large-scale simulation models of complex physical phenomena, and posing root-finding problems that involve performance measures that can only be estimated by the simulations. The practitioner in such cases is well aware that while he really seeks a root of the equation $g(x) = 0$, he only has a simulation capable of generating

an estimator of the function g. Furthermore, if the function g is to be replaced by an approximation obtained through a simulation (and solved as a DRFP), the practitioner would like to rigorously quantify the error in the estimated solution due to replacing the original problem with a surrogate. Toward further motivation, we now present an example. (We present two additional examples of SRFPs in the online appendix.)

Example (*Individual Choice Theory*): Consider the often encountered context of an individual faced with a set of alternatives available to perform a desired task (e.g., a driver having to choose one of various available routes to reach his destination; a cell phone user having to choose one of multiple connection plans that differ in "available minutes" and monthly cost). The theory of individual choice behavior [Ben-Akiva and Lerman 1985] is an enormously popular and useful branch of economics devoted to describing an individual's behavior in such settings. Specifically, suppose the set $\mathcal{C} = \{1, 2, \ldots, q\}$ comprising the alternatives $1, 2, \ldots, q$ is available to a user. Then the theory asserts that the user assigns individual utilities $U_1, U_2, \ldots, U_q$ to each of the competing alternatives $1, 2, \ldots, q$, and then makes the final choice $j = \text{argmax}_i\{U_i, i \in \mathcal{C}\}$. From a planner's perspective, the individual utilities are random variables, since they are mathematical constructs that are neither stated nor revealed. We then write $U_i(x(i)) = V(\beta_i, x(i)) + \epsilon(i)$, where $V(\beta(i), x(i)) = \mathrm{E}[U_i(x(i))]$ is a deterministic utility function of the known attribute value $x(i)$; β_i is the vector of unknown parameters associated with alternative i; and $\epsilon(i)$ is the random component. The probability of the individual choosing alternative i for a given vector of attribute values $x = (x(1), x(2), \ldots, x(q))$ is thus $p(x; i) = \Pr\{V(\beta_i, x(i)) - V(\beta_j, x(j)) \geq \epsilon(j) - \epsilon(i), j \in \mathcal{C}\backslash\{i\}\}$. Individual choice theory then usually assumes a functional form for $V(\beta_i, x(i))$ (e.g., a linear model), a distributional form for the random vector $(\epsilon(1), \epsilon(2), \ldots, \epsilon(q))$ (e.g., multivariate normal), and estimates parameters β_i using observed choices within a population of "identical" individuals.

Towards constructing a concrete example of an SRFP in the above context, consider the case of a traffic network having q routes between an origin and a destination. Drivers in the network choose one of the various available routes by maximizing utilities they assign to each route. (The utility assigned to a route may be a driver-specific function of various route attributes, e.g., cost, travel time, congestion.) Classical traffic theory [Sheffi 1985] states that under certain conditions, and in the long term, such networks settle down to what is called a *user equilibrium*—a state (characterized by path flows) where no driver can unilaterally switch routes and accrue additional utility [Sheffi 1985]. In other words, the "resting" route choices of the drivers can be assumed to be the solution to a game among the drivers that maximizes their individual utilities. Suppose now that a traffic planner has the ability to set toll prices $x(i), i = 1, 2, \ldots, q$ on the individual routes [Lawphongpanich et al. 2006], and thereby influence the equilibrium attained by the driver population. Suppose further that the planner would like the equilibrium resulting from the drivers' game to be $(\gamma(1), \gamma(2), \ldots, \gamma(q))$, where $\gamma(i), i = 1, 2, \ldots, q$ is the proportion of drivers choosing the ith route. This desired equilibrium may represent the *system-optimal* state as defined by the planner (e.g., that state which minimizes system-wide expected travel time, expected fuel consumption, expected revenue, or some combination of these). It is then natural for the planner to seek those values of $x(i)$ that will possibly achieve the target probabilities $\gamma(1), \gamma(2), \ldots, \gamma(q)$. Formally, the planner would thus like to choose the decision vector x such that $p(x; i) = \gamma(i), i \in \mathcal{C}$, where β_i is known. This is an SRFP if the function $p(x; i), i = 1, 2, \ldots, q$ is not known in "closed-form," but estimates of $p(x; i)$ can be constructed using simulation, for any specified vector x.

We note here that the toll-pricing problem as presented is formulated so that the dimensionality of the decision vector $(x(1), x(2), \ldots, x(q))$ is the same as the dimensionality of the output vector $(\gamma(1), \gamma(2), \ldots, \gamma(q))$. This results in a $q \times q$ stochastic

nonlinear system, and hence falls within the scope of this article. (Later in the article, we will illustrate solving such a problem in fifteen dimensions using two different methods.) Admittedly, the formulated example is somewhat specific, in that a planner may want to make a decision on more than just the toll price. So a more general formulation would have involved $\nu \geq 1$ factors on each route, resulting in a stochastic nonlinear system having νq variables and q equations. We note two points in this regard. First, it is not at all uncommon, due to matters relating to simplicity or expediency, that planners focus on just a single decision variable per product. (This is routine when pricing travel mode choices; see for example, Ben-Akiva and Lerman [1985].) So while our example is not general, it is still arguably representative. Second, as noted by Ortega and Rheinboldt [1970], the theory pertaining to solving nonlinear systems having different numbers of equations and variables is not well understood, even in the deterministic context. This is particularly so in contexts having nonisolated roots, which tends to be common in nonlinear systems having different numbers of equations and variables. Understandably then, what is known in the stochastic context is even less, and we will say nothing further on this class of problems.

We close our discussion of applications by noting that SRFPs are closely related to simulation optimization. Specifically, the first-order conditions characterizing the minimum of a real-valued function f, assuming that a consistent estimator of $g(x) = \nabla f(x)$ is available, also constitute an SRFP. Simulation optimization problems are currently of great interest (see, e.g., Henderson and Nelson [2006]). It so happens that all SRFPs can be reformulated as equivalent SOPs. For more on this, including when such a strategy might be appropriate, see Section E in the online appendix.

3. NOTATION AND TERMINOLOGY

The following is a list of key notations and definitions adopted in the article: (i) π^* and x^* denote the set of true solutions to the SRFP and a solution in the set π^*, respectively; (ii) If $x = (x(1), \ldots, x(q))^T$ is a $q \times 1$ vector, then the L_2 norm of x is defined by $\|x\| = \sqrt{x(1)^2 + x(2)^2 + \cdots + x(q)^2}$; (iii) If x is a $q \times 1$ vector, then $|x|$ is a $q \times 1$ vector having as elements the absolute values of the elements of x; (iv) If x and y are $q \times 1$ vectors, by $x \geq y$ we mean that each element of x is not less than the corresponding element in y; (v) the sequence of random vectors $\{X_n\}$ is said to almost certainly converge to a random vector X, written as $X_n \to X$ a.s., if $\Pr\{\lim_{n\to\infty} X_n = X\} = 1$; (vi) the sequence of random vectors $\{X_n\}$ is said to converge to a random vector X in probability, written as $X_n \xrightarrow{p} X$, if for all $\varepsilon > 0$, $\lim_{n\to\infty} \Pr\{\|X_n - X\| > \varepsilon\} = 0$; (vii) the sequence of random vectors $\{X_n\}$ is said to converge to a random vector X in distribution, written as $X_n \xrightarrow{d} X$, if $\lim_{n\to\infty} \Pr\{X_n \leq t\} = \Pr\{X \leq t\}$ for all continuity points t of $\Pr\{X \leq t\}$; (viii) for a sequence of real numbers $\{a_n\}$, we say $a_n = o(1)$ if $\lim_{n\to\infty} a_n = 0$; and $a_n = O(1)$ if $\{a_n\}$ is bounded, that is, there exists $c > 0$ with $|a_n| < c$ for all n; (ix) for a sequence of random variables $\{X_n\}$, we say $X_n = O_p(1)$ if $\{X_n\}$ is stochastically bounded (or bounded in probability), that is, for all $\varepsilon > 0$, there exists $M > 0$ such that $\Pr\{|X_n| > M\} < \varepsilon$, for all n; (x) $\text{dist}(x, B) = \inf\{\|x - y\| : y \in B\}$ denotes the Euclidean distance between a point $x \in \mathbb{R}^q$ and a set $B \subset \mathbb{R}^q$; (xi) $\text{dist}(A, B) = \sup\{\text{dist}(x, B) : x \in A\}$ denotes the distance between two sets $A, B \subset \mathbb{R}^q$; (xii) $\mathbb{H}(A, B) = \max(\text{dist}(A, B), \text{dist}(B, A))$ refers to the Hausdorff distance between two sets A and B; (xiii) if D is a finite set, then $|D|$ is the cardinality of D; (xiv) the notation $\mathbb{I}\{\cdot\}$ is used to denote the indicator function, that is, $\mathbb{I}\{\mathcal{A}\} = 1$ if the statement $\mathcal{A}$ is true, and zero otherwise; (xv) the words "root" and "zero" of a function $f : D \subset \mathbb{R}^q \to \mathbb{R}^q$ are used interchangeably to refer to $x \in D$ satisfying $f(x) = 0$; (xvi) $N(\mu, \Sigma)$ is a Gaussian random vector with mean μ and covariance matrix Σ. We assume that Σ is positive semidefinite.

4. THE PROBLEM SETTING AND SCOPE

Rheinboldt [Ortega and Rheinboldt 1970, Rheinboldt 1987], in his now famous treatises on solving nonlinear systems of equations, states that a thorough investigation into DRFPs involves answering questions that generally fall into one of two classes: (i) solvability or existence-type questions; and (ii) identification or approximation-type questions. In the former class there are loosely stated questions such as (a) How many (if any) solutions to the given system lie in a specified set? (b) How do the solutions vary under small changes to the problem? In the latter class are questions such as (c) Find a set containing at least one solution to the given system; (d) approximate a solution to the given system; and (e) approximate all solutions to the given system.

While these classes fully describe an investigation into SRFPs as well, the extent of what is known in the context of SRFPs is far less than that of DRFPs. Accordingly, our focus in this article will be limited to describing what is known in the context of answering the first and second questions in the second category, i.e., questions (c) and (d). Specifically, the operative definition of SRFPs that will be in effect in this article is stated as follows.

Given: A simulation that generates, for any $x \in D \subset \mathbb{R}^q$, an estimator $G_m(x)$ of the function $g : D \to \mathbb{R}^q$ such that $G_m(x) \xrightarrow{d} g(x)$ as $m \to \infty$, for all $x \in D$.

Find: A zero $x^* \in D$ of g, that is, find x^* such that $g(x^*) = 0$, assuming one exists.

As stated, the SRFP makes no assumptions about the nature of $G_m(x)$, except that $G_m(x) \xrightarrow{d} g(x)$ as $m \to \infty$, where m is the "sample size," that is, some measure of simulation effort that is usually well-defined depending on the context. In the context of terminating simulations [Law 2007, pp. 490 to 491], m usually refers to the number of times the simulation is called in computing the estimator $G_m(x)$. In nonterminating simulations, m usually refers to the "length of time" the simulation is executed when computing the estimator $G_m(x)$. That $G_m(x) \xrightarrow{d} g(x)$ as $m \to \infty$ is a standing assumption, and particular methodologies that we will discuss make further assumptions about G_m as needed. For the purposes of this article, the feasible set D is assumed to be known, that is, any constraint functions involved in the specification of D are observed without error. While this is a limitation, the question of identifying stochastic roots in the presence of stochastic constraints appears to have been studied very little.

5. SOLVING SRFPs

Methods for solving SRFPs can be classified broadly into three groups: (i) stochastic approximation type methods; (ii) sample-path methods; and (iii) parametric/semiparametric methods. The first two of these groups are discussed in detail in the sections that follow. The third group—parametric/semiparametric methods—is included in Section C of the online appendix.

5.1. Stochastic Approximation

Recall Newton's method for DRFPs on the function $g : \mathbb{R}^q \to \mathbb{R}^q$: $x_{n+1} = x_n - (J_g(x_n))^{-1} g(x_n)$, where $J_g(x)$ is the *Jacobian* matrix of g at the point x, that is, the ith row of $J_g(x)$ is the gradient of the ith component of g at x. Robbins and Monro [1951] introduced an analogous iteration for SRFP contexts, that is, contexts where g can only be estimated, by substituting the estimator G for g and then including a carefully chosen sequence $\{a_n\}$ aimed at nullifying the effects of randomness. (Robbins and Monro's original iteration was aimed at target functions g that were assumed to have a certain structure, which will be clarified.) Formally, the Robbins and Monro iteration is written as

$$X_{n+1} = X_n - a_n G(X_n). \tag{1}$$

While the recursion in (1) is simple, nearly five decades of research have gone into analyzing it since its introduction. Interestingly, and in a simplistic sense, much of such analysis has centered around just two questions.

—What conditions on the underlying function g, its estimator G, and the sequence $\{a_n\}$ guarantee that the iterates $\{X_n\}$ converge to a zero of the function g?
—At what "rate" do the iterates converge to a zero when they do, and what choice of $\{a_n\}$ guarantees that the iterates converge at a "fast" rate?

In the process of answering these questions, a large class of algorithms—henceforth called Stochastic Approximation (SA) type methods—has emerged and been thoroughly analyzed. Numerous variations, primarily based on the choice of the sequence $\{a_n\}$, have been devised to balance simplicity, economy of computation, and speed of convergence of the iterates in (1). In what follows, we provide a broad overview of some of the important variations, while discussing their convergence characteristics and practical performance. For practitioners, we include a separate section that summarizes three of these variations in a form that is readily implementable. Excellent survey articles on the theoretical properties of SA type methods are widely available, for example Ruppert [1991] and Lai [2003]. We thus limit our exposition to an overview of the key results, and direct the interested reader to available material.

5.1.1. Basic Convergence Results. In understanding the convergence of the recursion in (1), it is useful to write it in the slightly different form

$$X_{n+1} = \overbrace{X_n - a_n g(X_n)}^{\text{“deterministic”part}} - \underbrace{a_n \left(G(X_n) - g(X_n)\right)}_{\text{“stochastic”part}}. \tag{2}$$

As noted in the very readable book by Wasan [1969], the decomposition in (2) mimics the deterministic Newton's recursion more closely. Specifically, it can be seen that the SA recursion for SRFPs includes a Newton-like term appearing as the "deterministic" part, and an extra term appearing as the "stochastic" part. (We write "deterministic" within quotes because X_n is a random vector, and so this component is really not deterministic.) Accordingly, in order for the recursion in (2) to converge in some precise sense, two types of assumptions need to be made: (i) assumptions on the structure of the function g akin to those that one routinely sees in deterministic Newton recursions (e.g., monotonicity); and (ii) assumptions on the sequence $\{a_n\}$ to ensure that the stochastic part is driven to zero.

It may be argued that most of the papers on SA-type methods (for SRFPs) in the period between 1951 and the early 1970s made assumptions that almost invariably fell into the categories (i) and (ii). For example, the original paper by Robbins and Monro [1951], written for the one-dimensional case, made three assumptions to guarantee that the iterates in (1) converged in the L_2 norm: (a) the function g satisfies $g(x^*) = 0$, $g(x) < 0$ for $x < x^*$, and $g(x) > 0$ for $x > x^*$; (b) the positive-valued sequence $\{a_n\}$ satisfies $\sum_{n=1}^{\infty} a_n = \infty$; and (c) $\sum_{n=1}^{\infty} a_n^2 < \infty$. Of these, the first two assumptions "work" on the deterministic part, to ensure that the iterates head toward a zero of the function g. The last assumption, $\sum_{n=1}^{\infty} a_n^2 < \infty$, is aimed purely at driving the stochastic component to zero through a stipulated minimum decrease rate on the sequence $\{a_n\}$. Specifically, it is easy to show that under mild conditions on the second moment of $G(x)$, for example, $\mathrm{E}[(G(x) - g(x))^2] \le \sigma^2 < \infty$ for all x, the stochastic part $a_n(G(X_n) - g(X_n))$ will disappear in the limit. (However, this is not the way Robbins and Monro actually establish convergence.) Blum [1954a] establishes a.s. convergence of the iterates in (1) under slightly weaker conditions on the structure of the target function g. Blum [1954b]

was also the first to prove the a.s. convergence of the iteration in (1) in multiple dimensions. The conditions for such convergence are again remarkably similar to the stipulations in one dimension. They essentially involve structural conditions on the function g (along with $\sum_{i=1}^{\infty} a_n = \infty$) to ensure convergence of the deterministic part, and conditions on the estimator (along with $\sum_{i=1}^{\infty} a_n^2 < \infty$) to ensure the vanishing of the stochastic part.

Primarily on account of Ljung [1977a, 1977b] the modern method of analyzing SA type iterations usually follows a markedly different route. To illustrate this, consider again the following decomposition of the SA algorithm:

$$X_{n+1} = X_n - a_n \underbrace{g(X_n)}_{\text{mean term}} - a_n \underbrace{(G(X_n) - g(X_n))}_{\text{noise term}}. \tag{3}$$

Current methods of proving convergence usually proceed along four steps: (i) assume that the "noise term" in (3) is a martingale difference with bounded variance; (ii) make appropriate assumptions on the rate at which the sequence $\{a_n\}$ diminishes to zero and show that the cumulative effect due to the noise term vanishes "on average"; (iii) use a certain extended form of the Arzelà–Ascoli theorem (see Kushner and Yin [2003, p. 102]) to show that the iterates $\{X_n\}$ converge to a limit or invariant set for the ordinary differential equation (ODE) $\dot{x} = g(x)$, as $n \to \infty$; and (iv) analyze the behavior of the resulting ODE using traditional methods of Lyapunov stability (see Kushner and Yin [2003, Ch. 4]) to show that the iterates converge to a subset of the limit or invariant set for the ODE. Interestingly, the framework of step (iv), in addition to helping prove convergence, often facilitates further insight into the nature of the solution upon performing a classic phase-plane analysis.

In detailing the preceding recipe in a more general context, we consider the following projected form of (1):

$$X_{n+1} = \Pi_D[X_n + a_n G(X_n)] = X_n + a_n G(X_n) + a_n Z_n, \tag{4}$$

where D is a compact set, $\Pi_D[x]$ denotes the closest point in D from x, and Z_n is the correction term $\Pi_D[X_n + \epsilon_n G(X_n)] - X_n + a_n G(X_n)$. If D is convex, the projection map Π_D is well defined. In particular, when D is a box with the form $D = \{x \in \mathbb{R}^q : l(i) \le x(i) \le u(i), i = 1, \ldots, q\}$, the projection $y = \Pi_D[x]$ is a pointwise projection defined by

$$y_i = \begin{cases} l(i) & \text{if } x(i) \le l(i), \\ x(i) & \text{if } l(i) < x(i) < u(i), \\ u(i) & \text{if } u(i) \le x(i). \end{cases} \text{ and}$$

for each $i = 1, \ldots, q$. For the discussion on more general constraint sets, see Kushner and Yin [2003, pp. 107–108]. We now formally list a result on the convergence of SA-type methods within the ODE context (see Kushner and Yin [2003, p. 132].)

THEOREM 5.1 (CONVERGENCE OF SA THROUGH ODE METHOD). *Consider the constrained SA algorithm* (4) *with box constraints. Assume the following.*

A_1. $\sum_{n=1}^{\infty} a_n = \infty, a_n \ge 0$.
A_2. $\sum_{n=1}^{\infty} a_n^2 < \infty$.
A_3. $\sup_n E||G(X_n)||^2 < \infty$.
A_4. *There exists a function $g(\cdot)$ and random vectors β_n such that*
(a) $\mathrm{E}[G(X_n)|X_0, G(X_i), i < n] = g(X_n) + \beta_n$, *and*
(b) $\sum_{i=1}^{\infty} a_i \|\beta_i\| < \infty$ *w.p.1.*

Then dist$(X_n, \pi^*) \to 0$ *a.s. as* $n \to \infty$, *where* π^* *is the limit set of the ODE*

$$\dot{x} = g(x) + z, \quad z \in -C(x), \tag{5}$$

where the set $C(x)$ *is* $\{0\}$ *when* x *lies in the interior of* D, *and is the infinite cone generated by the outward normal vectors at* x *when* x *lies on the boundary of* D.

The assumption A_1 is, as usual, to provide the iterates adequate opportunity to head back to the vicinity of a zero at any point during algorithm evolution. The assumption A_3 holds, for example, if $E||G(x)||^2$ is continuous in x and D is compact. However, this condition is often difficult to verify in practice. For more discussion on verifying the continuity and uniform integrability of a function with random noise, see Kim and Henderson [2008]. The assumption A_4 is the famous martingale noise assumption which implies that $G(X_n)$ can be written as

$$G(X_n) = g(X_n) + \delta M_n + \beta_n, \tag{6}$$

where the term δM_n is a martingale difference. This assumption means that the observation $G(X_n)$, given the history $G(X_1), G(X_2), \ldots, G(X_{n-1})$, depends only on X_n. In other words, the dependence of the said conditional random vector on the history of observations is embedded entirely within the current iterate X_n.

Plugging (6) in (4), we see that the SA iteration becomes

$$X_{n+1} = X_n + a_n g_n(X_n) + a_n Z_n + a_n \delta M_n + a_n \beta_n. \tag{7}$$

The assumption A_4, when combined with assumptions A_2 and A_3, ensures that the term $a_n \delta M_n$ vanishes a.s. This shows that the iterates in (7) reduce to that of a "mean" iteration involving just the first three terms on the right-hand side of (7). The extended form of the Arzelà–Ascoli theorem can then be invoked to show that the asymptotic behavior of the mean iteration is essentially the same as the asymptotic behavior of the projected ODE defined in Theorem 5.1. Specifically, under the assumptions A_1–A_4, the iterates $\{X_{n+1}\}$ in (7) are such that the distance between X_n and a limit set of the ODE defined in Theorem 5.1 tends to zero as $n \to \infty$, a.s. See Chapter 6 in Kushner and Yin [2003] for further extensions, including relaxations on the sequence $\{a_n\}$.

5.1.2. Convergence Rates. Since Chung [1954] and Sacks [1958] developed the first asymptotic normality results, different types of convergence rate results have been developed for SA-type iterations. A very readable and interesting review of results up to around 1990 is provided by Ruppert [1991]. A fairly general result on the asymptotic normality of multidimensional SA algorithms can be found in Fabian [1968]. To facilitate understanding of the basic ideas, we introduce a simplified version of the result in Fabian [1968]. For a corresponding analysis through the ODE method, see Kushner and Yin [2003, Ch. 10].

THEOREM 5.2 (CONVERGENCE RATES FOR SA). *Consider the unconstrained SA algorithm* (1). *Assume that the assumptions for almost sure convergence hold and the iterates* $\{X_n\}$ *converge to a limit point* x^* *in probability . Furthermore, assume*

A_1. *g has a positive definite Jacobian matrix* $J_g(x^*)$ *at* x^*;
A_2. α *is in* $(1/2, 1]$, $n^\alpha a_n \to a$ *for some* $a > 0$; *and if* $\alpha = 1$, $a > \frac{1}{2\lambda}$, *where* λ *is the smallest eigenvalue of* $J_g(x^*)$;
A_3. $\sup_{\|x-x^*\| \le \epsilon} E[\|G(x) - g(x)\|^2 \mathbb{I}\{\|G(x) - g(x)\| > K\}] \to 0$ *as* $K \to \infty$, *for some* $\epsilon > 0$;
A_4. *The matrix function* $\Sigma_x = \mathrm{Var}(G(x))$ *is continuous at* x^*.

Then, $n^{\alpha/2}(X_n - x^*) \xrightarrow{d} N(0, V(x^*))$. *The covariance matrix* $V(x^*)$ *depends on* a_n, α, Σ_{x^*} *and* $J_g(x^*)$.

The proof of Theorem 5.2 and the explicit form of $V(x^*)$ can be found in Fabian [1968]. The assumption A_2 provides the decay rate of the sequence $\{a_n\}$, and the assumption A_3 implies that the variance of the stochastic components converges at x^*. The assumption A_4 is an essential condition to obtain the asymptotic normality of the iterates. One way to satisfy A_3 is to ensure that $\sup_{\|x-x^*\|\le\epsilon} \mathrm{E}[\|G(x)-g(x)\|^{2+v}] < \infty$ for some $\epsilon > 0$ and some $v > 0$.

Theorem 5.2 shows that the asymptotic convergence rate of SA is essentially determined by the sequence $\{a_n\}$. If $\alpha = 1$, the optimal convergence rate $O(1/\sqrt{n})$ can be guaranteed, which is akin to a typical central limit theorem. Also, it so happens that the norm (largest eigenvalue) of the covariance matrix $V(x^*)$ is minimized when $a_n = (nJ_g(x^*))^{-1}$, thereby resulting in the optimal covariance $((J_g(x^*))^{-1})^T \Sigma_{x^*} (J_g(x^*))^{-1}$. Note that the form of the minimizer is identical to the multiplier of $g(x)$ in Newton's method. However, unlike the deterministic Newton's method, the gain a_n decreases at a rate of $1/n$, meaning that the effect of the stochastic noise term diminishes as the algorithm progresses.

In practice, since the Jacobian matrix $J_g(x^*)$ is usually unknown, the optimal value of a_n can only be estimated. This led to the development of what is now popularly known as adaptive SA methods. Most notably, Venter [1967] developed an adaptive scheme to estimate $g'(x^*)$ using finite differences of g for one-dimensional problems. Wei [1987] studied Venter's scheme for the multidimensional case, and developed a consistent estimator for $J_g(x^*)$. This work also established a set of sufficient conditions under which the optimal convergence rate $O(1/\sqrt{n})$ can be obtained. Ruppert [1985] studied an adaptive multivariate SA method for the least squares estimation problems—an analog of Newton's method applied to $\|g\|^2$.

Another popular adaptive SA method stems from the now famous simultaneous perturbation method [Spall 2000] of estimating the Jacobian matrix $J_g(x^*)$ within SA iterations. The key idea in simultaneous perturbation is to estimate all entries in the desired Jacobian matrix $J_g(x^*)$ using just one or two simulation calls at appropriately chosen locations. In high-dimensional problems, this method thus admits efficient implementation by greatly reducing the number of simulation calls expended per iteration. The resulting estimators are strongly consistent, and result in no deterioration of the asymptotic convergence rate, under certain conditions. Bhatnagar [2005] extends this method into a three-timescale version.

In addition to the above attempts at incorporating efficient gradient estimation mechanisms into the SA iteration, there have been other attempts at accelerating convergence of SA's iterates. Most notable among these is the remarkably simple averaging strategy advocated by Polyak and Juditsky [1992]. Specifically, Polyak and Juditsky [1992] demonstrate that under some assumptions that are similar to those for a nonaveraging estimator, the average $\bar{X}_n = \sum_{i=1}^n X_i/n$ achieves the optimal asymptotic convergence rate $O(1/\sqrt{n})$ without needing to know or estimate the Jacobian matrix! The limiting covariance matrix in such a case turns out to be identical to the covariance matrix based on the optimal choice of a_n in nonaveraging SA algorithms. This result is quite impressive, in the sense that by simply averaging the iterates, the algorithm achieves the optimal convergence rate with covariance matrix having the smallest possible eigenvalues. We now formally state the key result in Polyak and Juditsky [1992].

THEOREM 5.3 (CONVERGENCE RATES FOR THE AVERAGED ITERATES). *Consider the unconstrained SA algorithm* (1). *Assume that the assumptions for almost sure convergence and asymptotic normality hold. Furthermore, assume that* $a_n = O(n^{-\alpha})$ *for some* $\alpha \in (1/2, 1)$. *Then,* $\bar{X}_n \to x^*$ *a.s. as* $n \to \infty$, *and* $\sqrt{n}(\bar{X}_n - x^*) \xrightarrow{d} N(0, V(x^*))$, *where* $V(x^*) = ((J_g(x^*))^{-1})^T \Sigma_{x^*} J_g(x^*)^{-1}$.

5.1.3. Implementation. In principle, SA methods are easy to implement—choose the sequence $\{a_n\}$, and then use the recursion (1) to update the iterates $\{X_n\}$. It is clear, however, that the performance of SA is very sensitive to the choice of $\{a_n\}$, and finding "good" values for the sequence $\{a_n\}$ is a nontrivial problem [Spall 2000, 2003; Yakowitz et al. 2000; Pasupathy 2010; Pasupathy and Schmeiser 2009; Broadie et al. 2009]. Spall [2000] suggests setting $a_n = a/(n+n_0)^\alpha$, with $a > 0, n_0 \geq 0, \alpha \in (1/2, 1]$, resulting in the following SA iteration:

$$X_{n+1} = X_n - \frac{a}{(n+n_0)^\alpha} \left(U(X_n)\right)^{-1} G(X_n), \tag{8}$$

where $U(X_n)$ is an estimator of the Jacobian matrix $J_g(X_n)$. If we set $U(X_n) = I$, the above iteration gives us the first order SA algorithm.

The asymptotic distribution results in Theorem 5.2 and Theorem 5.3 give us some insight into the choice of $\{a_n\}$. For instance, they suggest setting $a_n = a/(n+n_0)^\alpha$ with $a = 1, \alpha = 1$. Interestingly, however, these asymptotically optimal choices do not result in uniformly good finite-time performance. Free and Ruppert [1990] report that $a = 1$ may not be the best choice for small sample sizes. They find that choosing $a > 1$ works better than $a = 1$ for small sample sizes by allowing iterate evolution with larger step sizes. Furthermore, it may be best (from a finite-time standpoint) to not choose $\alpha = 1$ either. By choosing $\alpha < 1$, a_n decays slowly and the algorithm behaves more like the deterministic Newton's method in the early iterations. This allows the algorithm to take larger step sizes and quickly move toward a neighborhood of x^*. The value of the stability constant n_0 does not affect the asymptotic performance of the SA algorithm, but it is an important tunable parameter. It is found that a positive value of n_0 results in stable behavior for a large a than $n_0 = 0$ (see Free and Ruppert [1990], Spall [2003]).

More recently, Broadie et al. [2009] have made some progress on this question of choosing parameters within SA. Their idea is to tune the sequence $\{a_n\}$ adaptively using "scaling" and "shifting" operations to ensure that the SA algorithm takes steps that are neither "too small" nor "too big," that is, the iterates make sufficient progress towards the solution without repeatedly overstepping. Numerical experiments on one-dimensional problems using such a strategy have shown promise in improving the finite-sample performance of SA, without sacrificing asymptotic properties.

For the estimator $U(X_n)$ of the Jacobian matrix $J_g(X_n)$ appearing in (8), we suggest using Venter's scheme for mid- to low-dimensional problems, and Spall's adaptive SP method for high-dimensional problems. In illustrating these methods, suppose $c_n = cn^{-\gamma}$ for some $c > 0$ and some $\gamma > 0$. Let e_j denote the jth column of the $q \times q$ identity matrix. For Venter's scheme (see Wei [1987]), set $Y_{nj}^{(1)} = G(X_n + c_n e_j), Y_{nj}^{(2)} = G(X_n - c_n e_j), j = 1, \ldots, q$ and $W_n = n^{-1}\sum_{i=1}^n Z_i$, where Z_i is the matrix whose jth column is $Z_{nj} = 0.5c_n^{-1}(Y_{nj}^{(1)} - Y_{nj}^{(2)})$. Then define $G(X_n) = q^{-1}\sum_{j=1}^q 0.5(Y_{nj}^{(1)} + Y_{nj}^{(2)})$ and $U(X_n) = W_n \mathbb{I}\{\det(W_n) \neq 0\} + I\mathbb{I}\{\det(W_n) = 0\}$, where I is the $q \times q$ identity matrix. Under a set of regularity conditions, X_n and $U(X_n)$ are consistent estimators and $\{X_n\}$ achieves the optimal convergence rate $O(1/\sqrt{n})$. Free and Ruppert [1990] demonstrate that the finite sample performance of X_n significantly depends on the quality of the estimator $U(X_n)$. The same is true with the difference sequence $\{c_n\}$, with small values of c_n decreasing the bias and increasing the variance of $U(X_n)$, and vice versa. There is no asymptotically optimal $\{c_n\}$, but the choice $\gamma = 4$ has been recommended in practice Frees and Ruppert [1990]. At each iteration, it might also be desirable to use the average of several replications of $U(X_n)$ to diminish the effect of noise.

For adaptive SPSA methods (see Spall [2003, Ch. 7]), the recursion $U(X_n) = f_n(\frac{n}{n+1}U(X_{n-1}) + \frac{1}{n+1}H_n)$ is used to estimate the Jacobian matrix $J_g(X_n)$. The

SA Algorithms for SRFPs

User-Specified "Parameters":

P1. A rule to compute the gain sequence $\{a_n\}$.

P2. A procedure to construct the estimator $U(x)$ of the Jacobian matrix $J_g(x)$ at a given solution x.

P3. A simulation having the following input and output.
Input: (i) initial random seed vector ω_0; (ii) a "design" or "candidate solution" x; and (iii) the constraint set D (iv) the computational burden b and the sample size m.
Output: (i) next random seed vector ω_0; and (ii) the estimator $G(x, \omega_0)$ of the root finding function $g(x)$.

Algorithm Logic:

0. Initialize the iteration number $n = 0$. Choose X_0.
1. Use components P2 and P3 to generate an estimate $G(X_n)$ of $g(X_n)$, and to construct $U(X_n)$.
2. Compute $X_{n+1} = \Pi_D(X_n - a_n(U(X_n))^{-1}G(X_n))$.
3. Set $n \leftarrow n + 1$ and go to 1.

Recommended "Parameter Values":

– Choose X_0 believed to be near x^*.
– Choose m depending on the noise level of the problem. Less than 10% of the total allowed computational budget b might be reasonable.
– First order SA:
  (i) set $U(x) = I$;
  (ii) use $a_n = a/(n + n_0)^\alpha$;
  (iii) choose $a \in [0.1, 0.5]$, $n_0 = 5 - 10\%$ of the total number of iterations, and $\alpha = 0.5$ or 0.6.
– Venter's procedure:
  (i) estimate $U(x)$ using (13) and (14) with the difference sequence $\{c_n\}$;
  (ii) use $a_n = a/(n + n_0)$ and $c_n = c/n^\gamma$;
  (iii) choose $a = 1.5$ or 2, $n_0 = 1$, $c = 0.5$ and $\gamma = 1/4$;
– Second order SPSA:
  (i) estimate $U(x)$ using (15) and (16) with the difference sequence $\{c_n\}$;
  (ii) use $a_n = a/(n + n_0)^\alpha$ and $c_n = c/n^\gamma$;
  (iiii) choose $\alpha = 0.6$, $a = 1$, $n_0 = 5 - 10\%$ of the total number of iterations, $c = 0.5$ and $\gamma = 0.1$.

Implementable Code Location:
`<https://filebox.vt.edu/users/pasupath/pasupath.htm>`.

Fig. 1. A self-contained illustration of two SA-type variations for solving SRFPs, along with parameters required from the user, algorithm logic, and recommended parameter values.

per-iteration estimate H_n is defined as $H_n = \frac{1}{2}[0.5c_n^{-1}\Delta_n^{-1}Z_n^T + (0.5c_n^{-1}\Delta_n^{-1}Z_n^T)^T]$, where $\Delta_n = (\Delta_{n1}, \Delta_{n2}, \ldots, \Delta_{nq})^T \in \mathbb{R}^q$ is a user-generated mean zero random vector, $\Delta_n^{-1} = (\Delta_{n1}^{-1}, \Delta_{n2}^{-1}, \ldots, \Delta_{nq}^{-1})^T$, and $Z_n = G(X_n + c_n\Delta_n) - G(X_n - c_n\Delta_n)$. f_n is a projection map onto the set of positive definite matrices, which is designed to cope with the possibility of the matrix $\bar{H}_n$ being nonpositive definite. One useful form of f_n is to take $f_n(\bar{H}_n) = \bar{H}_n + \delta_n I$, where δ_n is large enough to ensure positive definiteness of $f_n(\bar{H}_n)$. The regularity conditions on Δ_n are given in Spall [2000]. In general, Spall [2000] recommends generating each element of Δ_n from a Bernoulli ± 1 distribution.

With a view toward implementation, Figure 1 gathers the above recommendations and provides three variations of the SA algorithm logic for solving SRFPs. The figure includes the list of parameters that need to be provided by the user, the algorithm logic, and some recommended parameter choices. The user-provided components P1 through P3 are self-explanatory. We have assumed that the "simulation" component P3 assumes an oracular form—it takes as inputs a random (vector) seed, a candidate solution x in question, and the computational burden b; it returns the next (vector) seed and the estimator $G(x)$ realized using the chosen seed. We note that in some cases

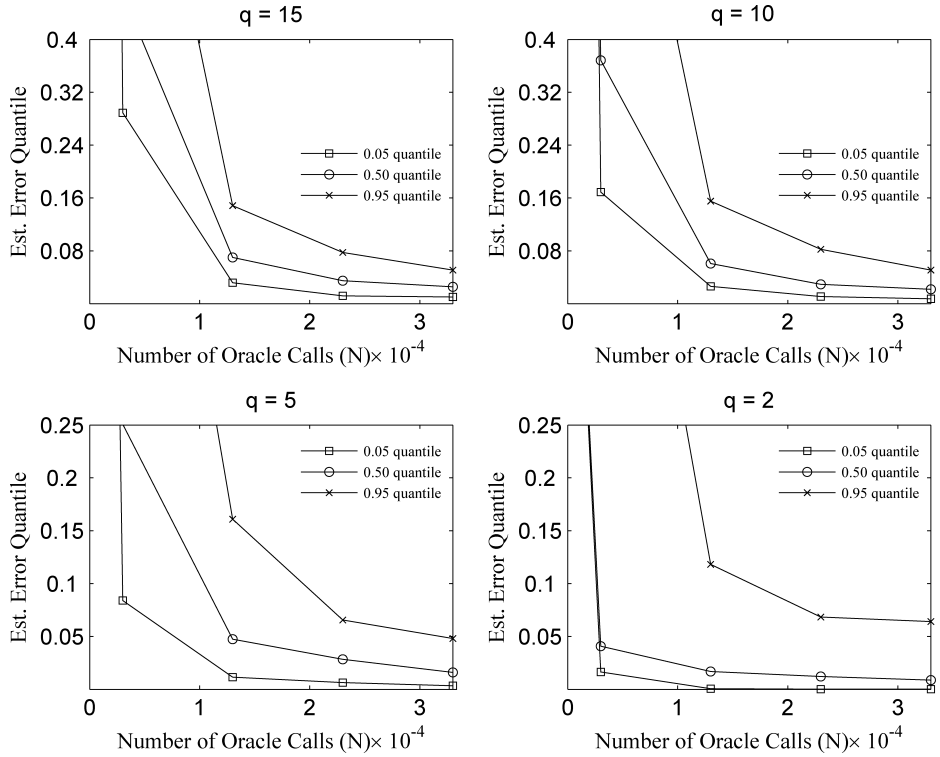

Fig. 2. Performance of second-order SA on the toll-pricing problem. The curves represent the estimated quantiles of the mean squared error (in solution) as a function of the number of oracle calls. The variable q represents the dimensionality of the problem. The initial guess was chosen to be 100 units away from the true solution in each coordinate.

(see for instance Glasserman [1991]) the Jacobian estimator $U(x)$ is directly provided by the simulation oracle, in which case component P2 becomes redundant.

Figure 2 shows results from the application of adaptive SA (second variation listed in Figure 1) on the toll-pricing problem discussed in Example 1 of Section 2. In this implementation, we assume that the utility functions $V(\beta, x)$ are linear functions of a single attribute (toll price) x with coefficient β, and that the error vector $(\epsilon_1, \epsilon_2, \ldots, \epsilon_q)$ is multivariate normal having zero mean and covariance matrix equal to the q-dimensional identity matrix. The target proportions vector γ, for simplicity, was chosen to be $\gamma = (1/q, 1/q, \ldots, 1/q)$; and the initial solution X_0 was chosen to be 100 units away from x^* in every direction.

A few aspects of Figure 2 are notable. Starting at $100 \times \sqrt{q}$ units, the error in SA's iterates as a function of the number of oracle calls falls rapidly, at least initially. (This explains why the left part of the graph is essentially out of the display panels.) Thereafter, the decay seems classically $O(1/\sqrt{N})$, where N is the number of simulation calls. For the 15 dimensional problem, for instance, the 50th percentile error, estimated with negligible sampling error, is about 0.07 after 11000 simulation calls, and about 0.02 after 31000 simulation calls. The corresponding error quantiles in two dimensions can be seen to be about 0.02 and 0.01 respectively. It is important to note here that the absolute number of simulation calls is a matter of scaling, and to this extent is specific to the problem being implemented. Across dimensions, the increase in computational budget for solving to a fixed precision seems somewhat less than linear. This is

explained by the use of the Jacobian estimates for direction choice—the algorithm does not approach the solution in a coordinate-wise fashion, but instead exploits information provided by the Jacobian estimate of the function g.

Consistent with past experience of numerous authors [Broadie et al. 2009; Spall 1998, 2003; Pasupathy and Schmeiser 2009], the performance of SA as depicted through Figure 2 was achieved only after some tuning of algorithm parameters. We were able to get adaptive SA to work robustly for different starting solutions (and different q values) after some tuning of the constant a that appears in the expression for the step size a_n. Specifically, we tried different scales for the constant a, and also resorted to *blocking* [Spall 2003], that is, not updating an iterate if the distance between successive iterates exceeded a user-specified threshold. The performance depicted by Figure 2 reflects the choice $a = 1000$ and a blocking factor of 100. We chose not to incorporate Polyak averaging as discussed in Section 5.1.2 because of the obvious presence of bias in the iterates. It is likely that substantial gains in performance on this problem are achievable with further tuning of the parameters $\{a_n\}$. (Additional discussion pertaining to the general performance of SA-type methods, and their comparison with SP methods, can be found in Section 5.2.5 and Section D of the online appendix.)

5.2. SAMPLE-PATH METHODS

The sample-path (SP) method, also known by various other names, including sample average approximation, stochastic counterpart method, and retrospective approximation, is another general technique used for solving SRFPs. It appears that the first reference to this technique is by Healy and Schruben [1991] and Shapiro [1991] in the context of optimization. Several other authors have subsequently used the technique in various other contexts, but predominantly optimization (see for instance, Atlason et al. [2004, 2005]; Plambeck et al. [1996]; Herrer et al. [2006]; Verweij et al. [2003]; and Rubinstein and Shapiro [1993].)

The conceptual idea of SP for SRFPs is easily stated. Instead of solving the actual SRFP, solve an approximate problem S_m obtained by substituting the unknown underlying target function g by its consistent estimator G_m. The function G_m is often not known analytically. Instead, akin to "black box" numerical optimization, the function value $G_m(x)$ at any x is "revealed" when the search routine visits x.

The appeal of SP methods is twofold. First, the method is conceptually very simple—replace the underlying target function g by a deterministic function obtained by "realizing" the estimator G_m, and then solve the resulting problem as a DRFP. Second, since SP methods essentially involve solving a DRFP approximation to the SRFP, the entire array of tools developed over the past many decades (for solving DRFPs) can be brought to bear. In other words, it may be argued that SP methods are not stochastic adaptations in the sense of SA type methods. Instead they directly use deterministic methods within an appropriately posed stochastic framework.

Notwithstanding the appealing simplicity, SP methods are only infrequently implemented in the generic form described. This is because the sample size required to obtain a stipulated quality of solution (e.g., through Theorem 5.7) often tends to be impractically large [Luedtke and Ahmed 2008; Bayraksan and Morton 2009; Pasupathy and Schmeiser 2009]. Furthermore, even if the stipulated sample size is manageable, the idea of solving a single deterministic approximation, while straightforward in principle, tends to be generally inefficient from the standpoint of computational efficiency. Obvious exceptions are contexts such as stochastic linear programs [Luedtke and Ahmed 2008; Higle and Sen 1991], where the function G_m is not a "black box" function, that is, the functional form of G_m is known.

5.2.1. SP Refinements. The need to make SP methods implementable and computationally efficient have led to the construction of refinements, most prominent being Retrospective Approximation (RA). Such refinements, instead of generating and solving a *single* sample-path problem with a "large enough" sample size, generate a *sequence* of sample-path problems with progressively increasing sample sizes $\{m_k\}$, and then solve these to progressively decreasing error tolerances $\{\epsilon_k\}$. This elaborate structure in the SP refinements is explicitly constructed to gain overall efficiency through the use of "warm starts" in each iteration. We now formally list a nonterminating version of RA.

RA Algorithms

0. Initialize iteration number $k(=1)$, sample size m_1, error tolerance ϵ_1, and solution X_0.
1. Using X_{k-1} as the initial solution, solve the kth sample-path problem to within error-tolerance ϵ_k, i.e., find X_k such that $\|G_m(X_k)\| \leq \epsilon_k$.
2. Calculate the root estimate $\overline{X}_k$ as the weighted sum of solutions $\{X_i\}_{i=1}^k$, i.e., $\overline{X}_k = (\sum_{i=1}^k w_{ik})^{-1} \sum_{i=1}^k w_{ik} X_i$, $w_{ik} \geq 0$, $\sum_{i=1}^k w_{ik} > 0$.
3. Update iteration number $k = k+1$, sample size m_k, and error tolerance ϵ_k. Goto Step 1.

The sequence $\{m_k\}$ of sample sizes, and the sequence $\{\epsilon_k\}$ of error tolerances, are chosen to tend to ∞ and 0 respectively. In principle, RA is designed to not expend too much effort in the early iterations, even if the numerical procedure expends a large number of search steps. This is because the small sample sizes in the early iterations ensure that each step taken by the numerical procedure (in accomplishing Step 1 above) is not too expensive. The later iterations are efficient, again in principle, because the starting solution for the generated problem in Step 1 is probably close to the true solution (due to "warm start"), and not too many steps are required to solve the sample-path problems. Solving the individual sample-path problems, as in generic SP, is accomplished by choosing any numerical procedure. RA methods were first designed in the context of one-dimensional SRFPs by Chen and Schmeiser [2001], and subsequently extended to multiple dimensions by Pasupathy and Schmeiser [2010]. A similar refinement for SOPs called *variable sample methods* is introduced and analyzed in detail by Homem-de-Mello [2003].

5.2.2. Basic Convergence Results. In this section, we present two basic results on the convergence characteristics of SP's iterates. We note that the results in this section are stated for the generic SP context. Since their extension to SP's refinements are fairly straightforward, we have chosen to omit them.

Unlike SA-type methods where convergence has generally been in the design space (i.e., distance between the iterate and a zero of the target function is shown to approach zero in some sense), convergence results for SP methods have tended to be both in the design space and in the function-value space. Specifically, suppose the random vector X_m^* is a zero of the random function $G_m(x)$. We might then ask under what conditions do the sequences $\{\text{dist}(X_m^*, \pi^*)\}$ and $\|g(X_m^*)\|$ go to zero a.s.? The answers to these questions turn out to be simple extensions of what is widely known in the SP literature for SOPs.

We now state two theorems that address convergence in the function-value space and the design space respectively. Versions of both of these theorems, along with proofs, can be found in Pasupathy [2010].

THEOREM 5.4 (CONVERGENCE OF FUNCTION VALUES IN SP METHODS). *Assume*

A_1. *The set of zeros $\pi^* \subset D \subset \mathbb{R}^q$ of the function g is nonempty;*
A_2. *The functional sequence $\{G_m(x)\}$ is such that the set of zeros Π_m^* of the function G_m is nonempty as $m \to \infty$ a.s.;*

A_3. *The functional sequence* $\{G_m(x)\} \to g(x)$ *uniformly as* $m \to \infty$ *a.s.*

Then, $\Delta_m = \sup_{x \in \Pi_m^*}\{\|g(x)\|\} \to 0$ *a.s. as* $m \to \infty$. *(Assume* $\Delta_m = \infty$ *if* $\Pi_m^* = \emptyset$*.)*

Note that Theorem 5.4 makes no assumptions about the continuity of g. Moreover, the assumption A_3 about the uniform convergence of $\{G_m(x)\} \to g(x)$ cannot be relaxed. For a counterexample, choose $D = \mathbb{R}, g(x) = x - 1$, and $G_m(x) = -(2m + x)\mathbb{I}\{x \in (-\infty, -m]\} + x\mathbb{I}\{x \in (-m, \infty)\} - 1$ for all outcomes in the probability space in which $G_m(x)$ is defined. Then $\pi^* = \{1\}$, $\Pi_m^* = \{1, -2m - 1\}$, and $\Delta_m = -2m - 1$.

Theorem 5.4 demonstrates a.s. convergence in the function-value space. In order to prove a.s. convergence in the design space, further assumptions need to be made on the domain D and the target function g, specifically that D is compact and g is continuous.

THEOREM 5.5 (CONVERGENCE OF SOLUTIONS IN SP METHODS). *Assume that*

A_1. *The set of zeros* $\pi^* \subset D \subset \mathbb{R}^q$ *of the function* g *is nonempty;*
A_2. *The functional sequence* $\{G_m(x)\}$ *is such that the set of zeros* Π_m^* *of the function* G_m *is nonempty as* $m \to \infty$ *a.s.;*
A_3. *The functional sequence* $\{G_m(x)\} \to g(x)$ *uniformly as* $m \to \infty$ *a.s.;*
A_4. *The function* g *is continuous on* D*; and*
A_5. *The set* D *is compact.*

Then, the set of sample-path solutions Π_m^* *converges a.s. to the set of true solutions* π^**, i.e.,* $dist(\Pi_m^*, \pi^*) \to 0$ *as* $m \to \infty$ *a.s.*

Again, the extra assumptions on the compactness of D and the continuity of g are required and cannot be relaxed. Simple counterexamples akin to those in Bartle [Bartle 1976, p. 123] can be constructed where, in the absence of the continuity of g and/or the compactness of D, the sample-path zeros $\{X_m^*\}$ can "escape" towards a location where there exists no zero for the function g. Not surprisingly, the conditions listed in the Theorems 5.4 and 5.5 match the corresponding theorems in the SOP context (e.g., Shapiro [2004]).

5.2.3. Convergence Rates. In this section, we present further results that characterize the speed of convergence of SP's iterates. Here again, for ease of exposition, the results are stated for the generic SP context.

In discussing the convergence rates that are obtainable using SP methods, we first list a basic result that ties the convergence rate of the estimator $G_m(x)$ to the rate at which solutions (or their function values) converge to their limit. Specifically, Theorem 5.6 asserts that in SP methods, under the conditions laid down in Theorems 5.4 and 5.5, convergence in the function-value space and the design space simply inherit the rate exhibited by the estimator $G_m(x)$ in use. A corresponding theorem for the SOP context can be found in Pflug [1996].

THEOREM 5.6 (CONVERGENCE RATES IN SP METHODS). *Consider the following conditions.*

A_1. *The functional sequence* $\{G_m(x)\}$ *is such that the set of zeros* Π_m^* *of the function* G_m *is nonempty as* $m \to \infty$ *a.s.*
A_2. $\beta(m)$ *is a (deterministic) function satisfying* $\lim_{m\to\infty} \beta(m) = \infty$ *so that the sequence* $\{\sup_{x\in\mathcal{D}} \|G_m(x) - g(x)\|\}$ *is* $O_p((\beta(m))^{-1})$*, i.e., for* $\epsilon > 0$*, there exists* K_ϵ *such that* $\sup_m \Pr\{\beta(m) \sup_{x\in\mathcal{D}} \|G_m(x) - g(x)\| \geq K_\epsilon\} \leq \epsilon$.
A_3. *There exists* $s > 0$ *such that* $\|g(x)\| \geq s\, dist(x, \pi^*)$ *for all* $x \in D$.

If assumptions A_1, A_2 *hold, then the sequence* $\{\sup_{x\in\Pi_m^*} \|g(x)\|\}$ *is* $O_p((\beta(m))^{-1})$*. If all three assumptions hold, then the sequence* $\{dist(\Pi_m^*, \pi^*)\}$ *is* $O_p((\beta(m))^{-1})$.

PROOF. See Section B in the online appendix. □

Something more specific than Theorem 5.6 can be said if we make further assumptions about the estimator G_m, the target function g, and the numerical procedure in use within the SP framework; see Pasupathy [2010] for one such version.

5.2.4. Sample Sizing Results. From an implementation standpoint, a natural question is whether a minimum sample size result can be derived to ensure that a solution obtained using an SP method is of a stipulated quality. Specifically, how large should the sample size m be to ensure that the solution X_m^* satisfies $\Pr\{\|g(X_m^*)\| \geq \epsilon\} \leq \alpha$ for given ϵ, α? Theorem 5.7 answers this question. (We refer the reader to Section 3 to clarify some of the notation used in Theorem 5.7.)

THEOREM 5.7 (MINIMUM SAMPLE SIZE IN SP METHODS). *Let $D \subset \mathbb{R}^q$ be a finite set. Let $\epsilon = (\epsilon(1), \epsilon(2), \ldots, \epsilon(q))$, $\delta = (\delta(1), \delta(2), \ldots, \delta(q))$ be vectors with strictly positive elements, and satisfying $\delta < \epsilon$. Define $\pi^*(\epsilon) = \{x \in D : |g(x)| \leq \epsilon\}$, $\Pi_m^*(\delta) = \{x \in D : |G_m(x)| \leq \delta\}$. Let $Y_i(x) = (Y_i(x,1), Y_i(x,2), \ldots, Y_i(x,q))$, $i = 1, 2, \ldots, m$ be i.i.d. random vectors having a moment-generating function that is finite in some neighborhood of zero, and such that $\sigma^2(j) = \sup_{x \in \mathcal{D}} Var(Y_1(x,j)) < \infty$ for $j = 1, 2, \ldots, q$. Suppose $G_m(x) = m^{-1} \sum_{i=1}^m Y_i(x)$ and $\alpha > 0$. Then,*

$$m \geq \text{Min}\left\{\frac{3\sigma^2(j)}{(\epsilon(j) - \delta(j))^2} \log \frac{|D|}{\alpha} : j = 1, 2, \ldots, q\right\}$$

implies $Pr\{\Pi_m^*(\delta) \not\subseteq \pi^*(\epsilon)\} \leq \alpha$.

PROOF. See Section B in the online appendix. □

Theorem 5.7 is constructed to closely match the result for the optimization context developed in Kleywegt et al. [2001] for stochastic discrete optimization. The minimum sample size expression in Theorem 5.7 shows a weak dependence on the size of the set D and the error probability α, and a strong dependence on the difference between the error tolerance δ used to stop the numerical procedure and the stipulated overall error tolerance ϵ. Theorem 5.7 has been stated for finite sets D. Its extension into continuous sets D is done in the usual fashion by assuming a Lipschitz condition on the underlying function g and then "approximating" the continuous set D by a finite set [Shapiro 2004, p. 375]. We emphasize that a limitation of Theorem 5.7 is that it applies only to estimators $G_m(x)$ that are constructed as the sample mean of i.i.d random variables. More importantly, its applicability is limited due to the fact that the bound for m tends to be overly conservative.

Theorem 5.7 addresses the question of sample sizing in generic SP methods. The corresponding question of choosing sample sizes in RA methods is arguably more relevant from the standpoint of implementation. Unlike in the generic SP method, where a single sample size needs to be chosen, in RA methods, we have to choose an entire sequence of sample sizes that tends to infinity. As detailed in Pasupathy [2010], it so happens that the optimal rate at which this sequence increases to infinity depends explicitly on the efficiency of the numerical procedure in use within RA. Specifically, more efficient numerical procedures allow faster increases in the sample size sequence $\{m_k\}$ for use within RA. This makes intuitive sense because it is best to delay the use of large sample sizes until RA's iterates are operating very close to the true solution, that is, the "warm start" is so good that only a few steps are required by the numerical procedure to solve the sample-path problem to within the specified tolerance. Table I, reproduced from Pasupathy [2010], offers recommendations on the sequence of sample sizes to use within RA, depending on the operative numerical procedure. A "Y" in Table I indicates that the combination is recommended, an "N" indicates that the combination is not

Table I. Recommended Sample-size Growth Rates for Different Convergence Rates of the Numerical Procedure to Solve Sample-path Problems.

	Exponential Growth e.g., $m_k = \lceil e^{1.1m_{k-1}} \rceil$	Polynomial Growth e.g., $m_k = \lceil m_{k-1}^{1.1} \rceil$	Linear Growth e.g., $m_k = \lceil 1.1m_{k-1} \rceil$
Polynomial Conv.	N	Y	Y
Linear Conv.	N	N	Y
Sublinear Conv.	N	N	NA

(Adapted from Pasupathy [2010].)

RA Algorithms for SRFPs

User-Specified "Parameters":

P1. A procedure for solving deterministic root-finding problems.
P2. A rule to compute the sample size sequence $\{m_k\}$.
P3. A rule to compute the error-tolerance sequence $\{\epsilon_k\}$.
P4. A detection rule for stopping the numerical procedure in P1.
P5. Weights $\{w_{ik} : i = 1, \ldots, k\}$ assigned to solutions obtained during the iterations.
P6. A simulation having the following input and output.
Input: (i) initial random seed vector ω_0; (ii) a "design" or "candidate solution" x; and (iii) the computational burden m.
Output: (i) next random seed vector ω_0; and (ii) the estimator $G_m(x)$ of the root finding function $g(x)$.

Algorithm Logic:

0. Initialize the retrospective iteration number $k = 1$. Set m_1, ϵ_1.
1. Use component P1 to solve the deterministic root-finding problem $G_m(x) = 0$ to within ϵ_k. Obtain the solution X_k.
2. Calculate the root estimate $\overline{X}_k$ as the weighted sum of solutions $\{X_i\}_{i=1}^k$, i.e., $\overline{X}_k = (\sum_{i=1}^k w_{ik})^{-1} \sum_{i=1}^k w_{ik} X_i, w_{ik} \geq 0, \sum_{i=1}^k w_{ik} > 0$.
4. Use RA Components P2 and P3 to get m_{k+1} and ϵ_{k+1}. Set $k \leftarrow k + 1$ and go to 1.

Recommended "Parameter Values":

– Choose P1 per convenience or based on any knowledge of the sample-path problems. For example, the currently most popular DRFP implementations tend to be variations of the Newton-Krylov method.
– Choose $m_1 = 1$, $m_k = \lceil c_1 m_{k-1} \rceil$ for $k \geq 2$, and $c_1 = 1.1$.
– Choose $\epsilon_k = c_2/\sqrt{m_k}$ where c_2 is chosen intuitively based on how much precision is required. For example, if the user needs a "precision" of 10^{-3} or 10^{-4} (in function value), choosing $c_2 = 1$ might be reasonable.
– Stop the numerical procedure after the kth iteration if $\|G_{m_k}(x)\| \leq \epsilon_k$
– After k iterations, set $w_{ik} = m_i$ for all $i \leq k$; or set $w_{ik} = 0$ for $i < k$ and $w_k = 1$ for $i = k$.

Implementable Code Location:
`<https://filebox.vt.edu/users/pasupath/pasupath.htm>`.

Fig. 3. A self-contained illustration of RA algorithms for solving SRFPs, along with parameters required from the user, RA logic, and recommended parameter values.

recommended, and an "NA" indicates that a more specific characterization is needed to make a recommendation. Pasupathy [2010] also demonstrates that, in RA methods, in order to ensure maximal efficiency under general conditions, the error tolerance sequence $\{\epsilon_k\}$ should be chosen so that $\epsilon_k = O(1/\sqrt{m_k})$. These recommendations are incorporated in the implementation discussed next.

5.2.5. Implementation. With a view toward implementation, Figure 3 provides the RA algorithm logic for solving SRFPs, along with the list of parameters that need to be provided by the user, and some recommended parameter choices. The user-provided

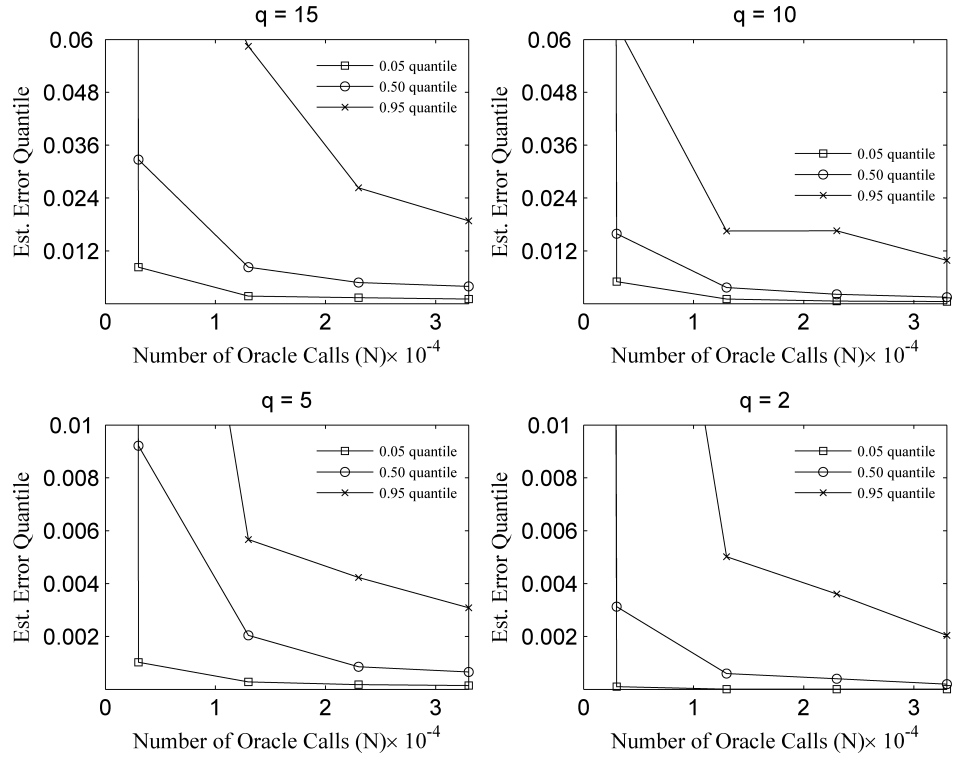

Fig. 4. Performance of second-order RA on the toll-pricing problem. The curves represent the estimated quantiles of the mean squared error (in solution) as a function of the number of oracle calls. The variable q represents the dimensionality of the problem. The initial guess was chosen to be 100 units away from the true solution in each coordinate.

components P1 through P5 are self-explanatory. We have assumed that the "simulation" component P6 has an oracular form. In other words, it takes as inputs a random (vector) seed, the computational effort m, and the candidate solution x in question, and returns the next (vector) seed and the estimator $G_m(x)$ realized using the chosen seed. Such an oracular structure is very routine in simulation contexts and gives the opportunity to employ "common random numbers" [Law 2007; Bratley et al. 1987] within the algorithm logic, thereby preserving any structural properties inherent in the sample-paths $G_m(x)$.

Implementable RA code is available through the Web site listed in Figure 3. We implemented this version of RA on the toll-pricing problem discussed in Example 1 of Section 2. The problem parameters, particularly the utility functions, error vector, target proportions, and the initial solution, were chosen to match the numerical example discussed in Section 5.1.3. The corresponding performance of RA is depicted through Figure 4.

As depicted in Figure 4, and just like adaptive SA, the error in RA's iterates as a function of the number of simulation calls (N) initially falls rapidly, followed by a classic $O(1/\sqrt{N})$ decay. For the 15 dimensional problem, the 50th percentile error is about 0.01 after 11000 simulation calls, and about 0.005 after 31000 simulation calls. The corresponding error quantiles in two dimensions are about 8×10^{-4} and 5×10^{-4} respectively. Just as in the case of adaptive SA, the computational effort expended

to solve to a fixed precision increases somewhat less than linearly as a function of dimension.

We emphasize that during implementation on this example problem, the algorithm parameters displayed in Figure 3 were used "as is," that is, with no tuning on the given problem. Such robustness is generally representative of RA and stands in contrast to SA, where tuning algorithm parameters tends to be a crucial requirement for reliable performance. Also, note that RA's performance in this example dominates that of adaptive SA by almost an order of magnitude. This, we think, is because of the way we have reported results for this example. When constructing the plots in Figures 2 and 4, we searched for a single set of algorithm parameters for use across all dimensions. SA, being sensitive to the choice of algorithm parameters, seems to have been disadvantaged by this policy. It is thus likely that the results would have been more comparable if different algorithm parameters were allowed and tuned across dimensions. We should note, however, that parameter tuning within SA is itself a nontrivial problem.

The performance of the RA and adaptive SA algorithms on the demonstrated example are typical of their more general performance. To summarize, RA can be used reliably, provided a good numerical procedure is chosen when solving the generated sample-path problems. Adaptive SA algorithms of the type displayed in Figure 1 can be reliably used provided a "good" initial solution is identified, and some parameter tuning is allowed. Both algorithms typically exhibit $O(1/\sqrt{N})$ decay in the number of simulation calls, and a roughly sublinear increase in computational effort with dimension. (For a more elaborate but general discussion of SP versus SA methods, see Section D of the online appendix.)

6. FUTURE RESEARCH AGENDA

In this section we briefly discuss two areas within SRFPs, where we believe advances in the current state of knowledge are much needed. For brevity, we limit ourselves to a surface-level description of these questions. We include a somewhat deeper discussion, and an additional topic of interest, in Section F of the online appendix.

6.1. Finite-Time Stopping for Implementation

A question of critical import that currently remains largely unresolved is that of the finite stopping of algorithms for solving SRFPs. In posing this problem precisely, suppose we force the user to specify a tolerance vector ϵ ($q \times 1$ vector of strictly positive numbers), and a confidence level $1 - \alpha$, in the sense that he/she is assumed to be satisfied with any solution X_m that is at most ϵ away from a zero of g (either in function-value space or in design space) with probability at least $1 - \alpha$. Formally, X_m is considered an (ϵ, α) solution to the SRFP if $\Pr\{|g(X_m)| > \epsilon\} \leq \alpha$. Then, we seek a stopping rule that can be employed within algorithms for SRFPs that will guarantee that the solution X_m obtained upon stopping is an (ϵ, α) solution.

For SP methods, under certain conditions, it can be shown that $g(X_m)$ asymptotically tends to a mixture of normal random variables, each of which is centered on 0. This implies that a reasonable "guess" of the "location" of $g(X_m)$ is the hyperrectangle $G_m(X_m) \pm a_m\hat{\sigma}_m$, where $\hat{\sigma}_m$ is the q-dimensional vector containing the diagonal elements of the estimated standard error of $G_m(X_m)$. This suggests stopping the algorithm when the constructed hyperrectangle lies within the hyperrectangle centered at the origin and having sides given by the vector 2ϵ. So terminate the algorithm when $|G_m(X_m) \pm a_m\hat{\sigma}_m| \leq \epsilon$, where $\{a_m\}$ is appropriately chosen. While intuitive, the behavior of this rule is still unclear. For instance: (a) Is the expected time at stopping finite? (b) When is the solution obtained upon stopping consistent? and (c) Can the

deterioration in coverage be characterized rigorously? All of these questions have direct analogues in the context of constructing sequential confidence intervals (see for instance Chow and Robbins [1965]). The question in (c) is particularly useful from the standpoint of implementation if results akin to those available for sequential confidence intervals can be derived.

6.2. Super-Quadratic SA-Type Iterative Schemes?

In the last decade or so, there has been a tremendous revival of attention given to variations of the generic Newton recursion for solving DRFPs on the target function $g(x)$. A notable example producing the iterates $\{x_n\}$ is

$$y_n = x_n - \frac{g(x_n)}{2g'(x_n)}; \quad x_{n+1} = x_n - \frac{1}{g'(y_n)} g(x_n). \tag{9}$$

Quite remarkably, the above recursion exhibits cubic convergence to the correct solution! (See for example, Weerakoon and Fernando [2000] for a very readable account.) Recall that the generic Newton method exhibits only quadratic convergence, although it uses one less observation on the Jacobian of g. The above development has led to a flurry of recent activity in this area, resulting in a large number of variations of the Newton iteration, and having orders of convergence much faster than quadratic. The references in this respect are too numerous to list (see Abbasbandy [2003] and Noor and Noor [2007] for more details and references).

The crucial question is whether such advances in DRFPs can be exploited to the context of SRFPs. The natural extension of iteration (9) to SRFPs (in one dimension) is $Y_n = X_n - 0.5a_n G'(X_n)^{-1} G(X_n); \quad X_{n+1} = X_n - b_n G'(Y_n)^{-1} G(X_n)$, where G is the estimator of g, and the sequences $\{a_n\}$, $\{b_n\}$ are user-defined. The generalized iteration in multiple dimensions can be written analogously. It is very likely that the conditions needed for the convergence of the proposed stochastic iteration to a valid zero of g will be very similar to those required by corresponding iterations such as adaptive SA (see Section 5.1.3). Furthermore, such convergence should be provable using the standard techniques outlined in Section 5.1, or even through the direct invocation of Dvoretsky's approximation theorem [Lai 2003]. The more interesting question is whether any gains in convergence rates, over corresponding SA-type iterations, are obtainable using the modified iteration.

ACKNOWLEDGMENT

We would like to thank the anonymous referees and the area editor for their thorough reviews and suggestions.

REFERENCES

ABBASBANDY, S. 2003. Improving the Newton–Raphson method for nonlinear equations by modified Adomian decomposition method. *Appl. Math. Comput. 145*, 887–893.

ATLASON, J., EPELMAN, M. A., AND HENDERSON, S. G. 2004. Call center staffing with simulation and cutting plane methods. *Ann. Oper. Res. 127*, 333–358.

ATLASON, J., EPELMAN, M. A., AND HENDERSON, S. G. 2005. Optimizing call center staffing using simulation and analytic center cutting plane methods. *Manage. Sci*. To appear.

BARTLE, R. G. 1976. *The Elements of Real Analysis*. Wiley, New York.

BARTON, R. R. AND MECKESHEIMER, M. 2006. Metamodel-based simulation optimization. In *Simulation*, S.G. Henderson and B.L. Nelson Eds. Elsevier, Amsterdam, 535–574.

BAYRAKSAN, G. AND MORTON, D. P. 2009. A sequential sampling procedure for stochastic programming. *Oper. Res*. To appear.

BEN-AKIVA, M. AND LERMAN, S. R. 1985. *Discrete Choice Analysis: Theory and Application to Travel Demand*. The MIT Press, Cambridge, MA.

BHATNAGAR, S. 2005. Adaptive three-timescale stochastic approximation. *ACM Trans. Model. Comput. Simul. 15, 1*, 74–107.

BLUM, J. 1954a. Approximation methods which converge with probability one. *Ann. Math. Stat. 25,* 2, 382–386.

BLUM, J. 1954b. Multidimensional stochastic approximation. *Ann. Math. Stat. 25*, 4, 737–744.

BRATLEY, P., FOX, B. L., AND SCHRAGE, L. E. 1987. *A Guide to Simulation*. Springer, Berlin.

BROADIE, M., CICEK, D. M., AND ZEEVI, A. 2009. An adaptive multidimensional version of the Kiefer-Wolfowitz stochastic approximation algorithm. In *Proceedings of the Winter Simulation Conference*. M. D. Rossetti et al. Eds. IEEE, Los Alamitos, CA, 601–612.

CHEN, H. AND SCHMEISER, B. W. 2001. Stochastic root finding via retrospective approximation. *IIE Trans. 33*, 259–275.

CHOW, Y. S. AND ROBBINS, H. E. 1965. On the asymptotic theory of fixed-width confidence intervals for the mean. *Ann. Math. Stat. 36*, 457–462.

CHUNG, K.L. 1954. On a stochastic approximation method. *Ann. Math. Stat. 25*, 463–483.

DUNKEL, J. AND WEBER, S. 2009. Stochastic root finding and efficient estimation of convex risk measures. *Oper. Res*. To appear.

FABIAN, V. 1968. On asymptotic normality in stochastic approximation. *Ann. Math. Stat. 39*, 1327–1332.

FREES, E. W. AND RUPPERT, D. 1990. Estimation following a Robbins–Monro designed experiment . *J. Amer. Stat. Asoc. 85*, 1123–1129.

FU, M. C. 1994. Optimization via simulation: A review. *Ann. Oper. Res. 53*, 199–247.

GIESECKE, K., SCHMIDT, T., AND WEBER, S. 2008. Measuring the risk of large losses. *J. Investment Manage. 6*, 4, 1–15.

GLASSERMAN, P. 1991. *Gradient Estimation Via Perturbation Analysis*. Kluwer, Amsterdam.

HEALY, K. AND SCHRUBEN, L. W. 1991. Retrospective simulation response optimization. In *Proceedings of the Winter Simulation Conference*. B. L. Nelson et al. Eds. IEEE, Los Alamitos, CA, 954–957.

HENDERSON, S. G. AND NELSON, B. L. (eds.). 2006. *Handbooks in Operations Research and Management Science: Simulation*. Vol. 13. Elsevier, Amsterdam.

HERER, Y. T., TZUR, M., AND YUCESAN, E. 2006. The multilocation transshipment problem. *IIE Trans. 38*, 185–200.

HIGLE, J. AND SEN, S. 1991. Stochastic decomposition: An algorithm for two-stage stochastic linear programs with recourse. *Math. Oper. Res. 16*, 650–669.

HOMEM-DE-MELLO, T. 2003. Variable-sample methods for stochastic optimization. *ACM Trans. Model. Comput. Simul. 13*, 108–133.

HOUSEHOLDER, A. S. 1970. *The Numerical Treatment of a Single Nonlinear Equation*. McGraw-Hill, New York.

HSIEH, M. AND GLYNN, P. W. 2002. Confidence regions for stochastic approximation algorithms. In *Proceedings of the Winter Simulation Conference*. E. Yucesan et al. Eds. IEEE, Los Alamitos, CA, 370–375.

JOSEPH, V. R., TIAN, Y., AND WU, C. F J. 2007. Adaptive designs for stochastic root-finding. *Statistica Sinica 17*, 1549–1565.

KELLY, C. T. 1995. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, PA.

KELLY, C. T. 2006. *Solving Nonlinear Equations with Newton's Method*. SIAM, Philadelphia, PA.

KIM, S. AND HENDERSON, S. G. 2008. The mathematics of continuous-variable simulation optimization. In *Proceedings of the Winter Simulation Conference*. S. J. Mason et al. Eds. IEEE, Los Alamitos, CA, 159–167.

KLEYWEGT, A. J., SHAPIRO, A., AND HOMEM-DE-MELLO, T. 2001. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim. 12*, 479–502.

KUSHNER, H. J. AND YIN, G. G. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, Berlin.

LAI, T. L. 2003. Invited paper: Stochastic approximation. *Ann. Stat. 31*, 2, 391–406.

LAW, A. M. 2007. *Simulation Modeling and Analysis*. McGraw-Hill, New York.

LAWPHONGPANICH, S., HEARN, D. W., AND SMITH, M. J. (eds.). 2006. *Applied Optimization: Mathematical and Computational Models for Congestion Charging*, Vol. 101. Springer, Berlin.

LJUNG, L. 1977a. Analysis of recursive stochastic algorithms. *IEEE Trans. Autom. Control AC-22*, 551–575.

LJUNG, L. 1977b. On positive real transfer functions and the convergence of some recursive schemes. *IEEE Trans. Autom. Control AC-22*, 539–551.

LUEDTKE, J. AND AHMED, S. 2008. A sample approximation approach for optimization with probabilistic constraints. *SIAM J. Optim*. To appear.

NEMIROVSKI, A., JUDITSKY, A., LAN, G., AND SHAPIRO, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim. 4.*

NEVEL'SON, M. B. AND KHAS'MINSKII, R. Z. 1973. *Stochastic Approximation and Recursive Estimation*. American Mathematical Society, Providence, RI.

NOOR, K. I. AND NOOR, M. A. 2007. Iterative methods with fourth-order convergence for nonlinear equations. *Appl. Math. Computation 189*, 1, 221–227.

ORTEGA, J. M. AND RHEINBOLDT, W. C. 1970. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York.

PASUPATHY, R. 2010. On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. *Oper. Res. 58*, 4, 889–901.

PASUPATHY, R. AND SCHMEISER, B. W. 2009. Retrospective-approximation algorithms for multidimensional stochastic root-finding problems. *ACM Trans. Model. Comput. Simul. 19*, 2, 5:1–5:36.

PFLUG, G. C. 1996. *Optimization of Stochastic Models: The Interface Between Simulation and Optimization*. Kluwer, Amsterdam.

PLAMBECK, E. L., FU, B. R., ROBINSON, S. M., AND SURI, R. 1996. Sample-path optimization of convex stochastic performance functions. *Math. Program. 75*, 137–176.

POLYAK, B. T. AND JUDITSKY, A. B. 1992. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim. 30*, 4, 838–855.

RHEINBOLDT, W. C. 1987. *Methods for Solving Systems of Nonlinear Equations in Several Variables*. Society for Industrial Mathematics.

ROBBINS, H. AND MONRO, S. 1951. A stochastic approximation method. *Ann. Math. Stat. 22*, 400–407.

RUBINSTEIN, R. Y. AND SHAPIRO, A. 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. Wiley, New York.

RUPPERT, D. 1985. A Newton–Raphson version of the multivariate Robbins–Monro procedure. *Ann. Stat. 13*, 236–245.

RUPPERT, D. 1991. Stochastic approximation. In *Handbook in Sequential Analysis.* Dekker, New York, 503–529.

SACKS, J. 1958. Asymptotic distribution of stochastic approximation procedure. *Ann. Math. Stat. 29*, 373–405.

SHAPIRO, A. 1991. Asymptotic analysis of stochastic programs. *Ann. Oper. Res. 30*, 169–186.

SHAPIRO, A. 2004. Monte Carlo sampling methods. In *Stochastic Programming*, A. Ruszczynski and Shapiro Eds. Elsevier, Amsterdam, 353–426.

SHEFFI, Y. 1985. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice Hall, New York.

SPALL, J. C. 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. Aerospace Electron. Syst. 34*, 817–823.

SPALL, J. C. 2000. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Trans. Autom. Control 45*, 1839–1853.

SPALL, J. C. 2003. *Introduction to Stochastic Search and Optimization*. Wiley, New York.

STEWART, G. W. 1973. *Introduction to Matrix Computations*. Academic Press, New York.

TODD, M. J. 1976. The *Computation of Fixed Points and Applications*. Springer, Berlin.

TRAUB, J. F. 1964. *Iterative Methods for the Solution of Equations*. Prentice-Hall, Englewood Cliffs, NJ.

VENTER, J. H. 1967. An extension of the Robbins–Monro procedure. *Ann. Stat. 38*, 181–190.

VERWEIJ, B., AHMED, S., KLEYWEGT, A., NEMHAUSER, G., AND SHAPIRO, A. 2003. The sample average approximation method applied to stochastic vehicle routing problems: a computational study. *Comput. Appl. Optim. 24*, 289–333.

WASAN, M. T. 1969. *Stochastic Approximation*. Cambridge University Press, Cambridge, UK.

WEERAKOON, S. AND FERNANDO, T. G. I. 2000. A variant of Newton's method with accelerated third-order convergence. *Appl. Math. Lett. 13*, 87–93.

WEI, C. Z. 1987. Multivariate adaptive stochastic approximation. *Ann. Stat. 15*, 1115–1130.

WU, C. F. J. 1985. Efficient sequential designs with binary data. *Ann. Stat. 13*, 1498–1508.

WU, C. F. J. 1986. Maximum likelihood recursion and stochastic approximation in sequential designs. In *Statistical Procedures and Related Topics*, J.V. Ryzin Ed. IMS, Hayward, CA, 298–313.

YAKOWITZ, S., L'ECUYER, P., AND VAZQUEZ-ABAD, F. 2000. Global stochastic optimization with low-dispersion point sets. *Oper. Res. 48*, 939–950.

YING, Z. AND WU, C. F. J. 1997. An asymptotic theory of sequential designs based on maximum likelihood recursions. *Stat. Sinica 7*, 75–918.

YOUNG, D. M. 1971. *Iterative Solution of Large Linear Systems*. Academic Press, New York.

Received September 2009; revised August 2010; accepted September 2010