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

RAGHU PASUPATHY
Virginia Tech
and
SUJIN KIM
National University of Singapore

The stochastic root-finding problem (SRFP) is that of finding the zero(s) of a vector function, i.e., 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 various 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 [**Probability and Statistics**]: Probabilistic algorithms (including Monte Carlo); I.6 [**Simulation Modeling**]: Simulation Theory

General Terms: Algorithms, Design, Theory

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

1. INTRODUCTION

In this paper we consider the Stochastic Root-Finding Problem — 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 Sir Isaac Newton, in the mid-seventeenth century, first introduced a method to successively approximate polynomials. SRFPs, by contrast, have only recently gained attention — particularly, as the modeling of complex physical processes on a digital computer has progressively become easier with rapid advances in simulation methodology and related software.

Our aim is an exposition of the topic of SRFPs at the level of an "advanced tutorial," aimed at three groups: (i) practitioners, (ii) applied scientists, and (iii) researchers. The first group consists of people who have encountered or framed an SRFP, most probably in a quest for efficiency within their operations, and merely seek a solution to their *specific* problem. We expect that this group of people

have neither the time nor necessarily an interest in the actual mechanics of the solution. They are simply looking for a reliable tool to solve the particular SRFP at hand. Towards helping this target audience, we make an explicit attempt to reference implementations whenever readily available, subjectively discuss SRFP contexts where such implementations are likely to "work," and in a somewhat limited fashion recommend what can be done when they "fail." Accordingly, this group can simply skip to the implementation sections within Section 5 (Sections 5.1.3 and 5.2.3, particularly the Figures 1 and 2) after reading Sections 1 through 3. We expect the second group to comprise of scientists from other disciplines who have "stumbled" upon an SRFP formulation. Unlike practitioners in (i), they are probably not trying to solve one specific SRFP. Instead they view it as a puzzle that repeatedly appears as part of a larger and potentially unrelated problem class that they are trying to solve. This group will find value in both the commentary on implementations relevant to the group in (i), and the basic theory that we outline in Section 5. We see the third group as being comprised essentially of researchers (particularly graduate students) who are looking to advance the frontiers in SRFPs. We expect that they are mostly aware of the basic results that are currently available and would like to know what directions might be fruitful research pursuits. In an attempt to satisfy this group, we include a section that discusses, in a fair amount of detail, a number of open questions relating to SRFPs. Depending on how aware the "researcher" is of the current literature, he/she may want to skim Section 5 before proceeding to Section 6.

As Micheal Fu [1994] notes in his review article on simulation optimization, it is but 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 and 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 him/her to articles that delve deeper into specific aspects that may be of interest. Our guiding philosophy has been to focus on the basic results in detail, provide plenty of references to more nuanced results, and list algorithms in easily implementable form along with a subjective discussion of important practical issues.

2. MOTIVATION

There is hardly a need to motivate the deterministic analogue of SRFPs, i.e., the question of solving nonlinear systems of equations (henceforth abbreviated as DRFPs for Deterministic Root Finding Problems). As is pointed out in the classic references by Rheinboldt and Ortega [1987; 1970], 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 provide a detailed motivation or a review 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 Traub [1964], Householder [1970] for solving one-dimensional DRFPs, Young [1971], Stewart [1973] for solving linear DRFPs, and Rheinboldt [1987], Rheinboldt and Ortega [1970], Todd [1976], Kelly [2006; 1995] for solving general DRFPs. (Also see Ehrlichman and Henderson [2007] and a few references therein for some very specific structured DRFPs.) There here has been a clear resurgence in interest on this topic in the last three years with the identification of decompositions that have made fourth-order convergence algorithms possible. See Abbasbandy [2003], Noor et al. [2007], Noor and Noor [2007b; 2006; 2007a], and Noor [2007] for more details.

It may be convincingly argued that the relevance of SRFPs is implied by that of DRFPs. This is because, 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, while he/she 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. Most importantly, 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 rigorous measures that quantify the deterioration caused due to the replacement of the original problem with a surrogate.

Toward further motivating such contexts, and in helping the reader identify SRFPs as they appear, we present three concrete examples of settings where the natural formulation turns out to be an SRFP. Each of these examples has been taken and suitably modified from recent references.

Example 1 (Submarine Stocking Problem): The question of how much inventory to hold is asked in a variety of settings, and is often posed as an SRFP. As an example, consider a naval vessel which, among other things, carries a large inventory of spare parts to support aircraft and shipboard machinery. The spare parts carried aboard these vessels are used to satisfy demand arising from failed equipment or scheduled maintenance activities. Since the nature of such demand is random, it is often difficult to predict in advance as to how much inventory a vessel should be carrying. In order to pose the above problem rigorously, assume that a vessel is supposed to carry q types of spare parts. Let $x_i, i = 1, 2, \ldots, q$ be a positive-valued continuous-variable (assumed for simplicity) that represents the quantity of the spare-part type i. Assume that the demand for spare parts arrives according to some general q-dimensional stochastic process, and the cumulative demand by

time t is $D(t) = (D_1(t), D_2(t), \ldots, D_q(t))$. The individual demands will often be correlated depending on the nature of the failure processes involved, and the level of substitutability across the spare parts. Suppose that the vector $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_q)$ represents the desired stockout probabilities. Assuming a planning horizon $[0, h]$, and denoting $g_i(x_1, x_2, \ldots, x_q) = \Pr\{D_i(h) > x_i\}$, the problem is that of identifying $x_1, x_2, \ldots, x_q$ such that $g(x_1, x_2, \ldots, x_q) = \gamma$, where $g \equiv (g_1, g_2, \ldots, g_q)$ and $\gamma \equiv (\gamma_1, \gamma_2 \ldots, \gamma_q)$. Under mild conditions, the solution to the above problem identifies the "least onboard inventory" that will satisfy the stipulated stockout constraints. In the presence of a stochastic simulation capable of generating the process $D(t)$, the above becomes a stochastic root-finding problem.

Example 2 (Measuring Financial Risk): A commonly used measure of risk associated with a financial position X, e.g., bond or stock, is what is called the *Value-at-Risk* (VaR). VaR (at level λ) is defined as

$$\mathrm{VaR}_\lambda(X) = \inf\{m \in \mathbb{R} : \Pr\{X + m < 0\} \leq \lambda\}. \tag{1}$$

VaR essentially connotes the loss quantile, i.e., the smallest monetary amount which when added to X ensures that the probability of a loss is kept below λ. (See Glasserman [1991] and Giesecke et al. [2008] for additional details and examples.) As noted in Giesecke et al. [2008], an alternative measure of risk called the *Utility-based Shortfall Risk* (USR) has recently been gaining in popularity. The USR for a position X is defined as

$$\mathrm{USR}_\lambda(X) = \inf\{m \in \mathbb{R} : \mathrm{E}[l(-m - X)] \leq \lambda\}. \tag{2}$$

for an appropriately defined loss function l. Dunkel and Weber [2009] observe that under certain conditions this is the same as m satisfying the equation

$$\mathrm{E}[l(-X - m)] = \lambda. \tag{3}$$

Now consider the question of estimating the financial risk associated with a position X either through VaR or through USR. In both cases, the resulting problem turns out to be an SRFP if the distribution of loss is not known analytically (as is often the case), but instead estimated through a simulation. Formally, suppose $g(x) = \Pr\{X + x < 0\}$. Then the problem of estimating VaR for X is essentially identifying a root of the equation $g(x) = \lambda$. A similar root finding problem holds when using USR as a measure and noting (3). In both these cases, if the function g can only be estimated (e.g., through Monte Carlo), a one-dimensional SRFP results.

Example 3 (Vehicular-Traffic Equilibrium): In managing urban traffic networks, planners routinely attempt "nudging" the system toward desirable equilibria through external controls, e.g., traffic signals, traffic information, tolls. (See Sheffi [1985] for some examples.) Consider, for instance, a simple network consisting of a single origin-destination pair in a highway network connected by q paths. Suppose that a planner has the ability to influence driver's behavior on the network through the imposition of tolls $x = (x_1, x_2, \ldots, x_q)$ on the q paths. Specifically, assume that the planner, through some means, has determined that he would like set the toll prices so that the resulting fraction of drivers choosing the q paths are $\gamma_1, \gamma_2, \ldots, \gamma_q$, respectively, where $0 \leq \gamma_i \leq 1, \sum_{i=1}^{q} \gamma_i = 1$. Denoting the total number of drivers who make the trip by the random variable D, and the number choosing path i by

the random variable $Y_i(x)$, a reasonable question to ask might be the following.

$$\text{Find } x \text{ such that } \mathrm{E}\left(\frac{Y_1(x)}{D}, \frac{Y_2(x)}{D}, \ldots, \frac{Y_q(x)}{D}\right) = (\gamma_1, \gamma_2, \ldots, \gamma_q),$$

given $(\gamma_1, \gamma_2, \ldots, \gamma_q)$ and a simulation capable of generating random variates from the distributions of D and $(Y_1(x), Y_2(x), \ldots, Y_q(x))$. In other words, the problem is that of identifying the toll prices $x_1, x_2, \ldots, x_q$ that ensure that the proportions of drivers choosing the various routes match pre-specified target proportions on average. While the problem presented is a gross simplification, realistic variations of the toll pricing problem are very routine and form an active area of enquiry within the transportation systems community (see Lawphongpanich et al. [2006]).

For additional examples of SRFPs, see Kushner [2003], Chen [1994], and Pasupathy [2009]. Also, most sample problems provided in Ortega and Rheinboldt [1970, Chapter 1] and Kelly [2006] automatically become SRFPs if the constituent functions can only be estimated.

Two other facts are in order when considering the relevance of SRFPs.

(i) All fixed-point problems — problems that demand solutions to the equation $f(x) = x, f : \mathbb{R}^q \to \mathbb{R}^q, x \in D \subset \mathbb{R}^q$ — where the function f is known only through a stochastic simulation can be equivalently formulated as SRFPs by simply defining $g(x) = f(x) - x$. This fact is significant because fixed-point problems arise routinely in disciplines such as economics, game theory, chemical engineering, physics, biology, and transportation systems (e.g., several hundred articles of such can be obtained through the *Journal of Fixed Point Theory and Applications*).

(ii) The first-order conditions for identifying the minimum of a real-valued function f defined in $\mathbb{R}^q$, when a consistent estimator of the gradient $g(x) = \nabla f(x)$ is available, is an SRFP. This fact is again significant because the problem of identifying the minimizer of a real-valued function f, given only an estimator of f, is popularly called the simulation-optimization problem (SOP). Such problems are currently of great interest. (See Andradóttir [1998; 2006], Azadivar [1999], Swisher et al. [2000], April et al. [2001], Ólafsson and Kim [2002], Fu [1994; 2005], and Henderson and Nelson [2006] for reviews.)

3. THE PROBLEM SETTING AND SCOPE

Rheinboldt [1987; 1970], 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 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 paper 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 paper is stated as follows.

Given: A simulation capable of generating, 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, i.e., find x^* such that $g(x^*) = 0$, assuming that one such 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 $m \to \infty$. This is a standing assumption and particular methodologies that we will discuss make further assumptions about G_m as needed. For the purposes of this paper, the feasible set D is assumed to be known, i.e., 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. Furthermore, we assume that the set D is *connected* by which we mean that any two points lying in the set D can be connected by a curve that completely lies in D. Most contexts in which SRFPs arise seem to have even more structure, e.g., D is typically convex, and so limiting our scope to connected sets D seems to be reasonable.

We conclude this section by noting that all SRFPs can indeed be posed as simulation optimization problems where a local minimum is desired. One specific formulation is as follows.

Given: A simulation capable of generating, 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 local minimizer $x^* \in D$ of $h(x) = g(x)^T g(x)$, i.e., find x^* having a neighborhood $\mathcal{N}(x^*)$ such that every $x \in \mathcal{N}(x^*)$ satisfies $h(x) \geq h(x^*)$, assuming that one such x^* exists.

To the extent that SRFPs can be posed as above, the entire literature on simulation-optimization problems is directly relevant. However, owing to a need to "draw a line" on the scope of the article, we have chosen not review this part of the SOP literature. The interested reader might consult the review sources listed earlier for entry points into the literature.

4. NOTATION AND TERMINOLOGY

The following is a list of key notation and definitions adopted in the paper: (i) x^* denotes a *true solution* to the SRFP; (ii) $X_n \xrightarrow{p} X$ means that the sequence of random variables $\{X_n\}$ converges to the random variable X in probability; (iii) $X_n \to X$ a.s. means that the sequence of random variables $\{X_n\}$ converges to the random variable X almost surely, i.e., with probability one; (iv) $X_n \xrightarrow{d} X$ means that the sequence of random variables $\{X_n\}$ converges to the random variable X in distribution; (v)

$\text{dist}(x, B) = \inf\{\|x - y\| : y \in B\}$ denotes the Euclidean distance between a point $x \in \mathbb{R}^q$ and a set $B \subset \mathbb{R}^q$; (vi) $\text{dist}(A, B) = \sup\{\text{dist}(x, B) : x \in A\}$ denotes the distance between two sets $A, B \subset \mathbb{R}^q$; (vii) $\mathbb{H}(A, B) = \max(\text{dist}(A, B), \text{dist}(B, A))$ refers to the Hausdorff distance between two sets A and B; (viii) 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; (ix) 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; (x) 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 $f(x) = 0$.

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/semi-parametric methods. Each of these groups is discussed in detail in the sections that follow. In addition to the core methodology, basic theoretical results, and key assumptions, we also include separate implementation sections aimed at the practitioner. Live links to computer code are listed whenever available.

5.1 STOCHASTIC APPROXIMATION

Recall Newton's method for DRFPs on the function $g : \mathbb{R}^q \to \mathbb{R}^q$:

$$x_{n+1} = x_n - (\nabla g(x_n))^{-1} g(x_n).$$

In the above recursion, the function g and its gradient ∇g are assumed to be known — a major difference from SRFPs where only a consistent estimator of g is available. In 1951, Herbert Robbins and Sutton Monro introduced a "Newton-like" iteration for solving SRFPs, by substituting the estimator G for g, and including a carefully chosen sequence $\{a_n\}$ aimed at nullifying the effects of randomness:

$$X_{n+1} = X_n - a_n G(X_n). \tag{4}$$

Note that in (4), the iterates are random variables and the sequence $\{a_n\}$ is specified by the user as part of the recursion.

While the recursion in (4) is simple, nearly five decades of research have gone into analyzing it after its original introduction by Herbert Robbins and Sutton Monro [1951] in a seminal paper. Interestingly, and in a simplistic sense, much of such analysis has centered around just two questions:

1. 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?
2. At what "rate" do the iterates converge to a zero when they do, and what choice of the sequence $\{a_n\}$ 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 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 (4). 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 two of these variations in a form that is readily implementable. Excellent survey articles on the theoretical properties of SA type methods are widely available. 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 (4), it is useful to write it in a slightly different form

$$X_{n+1} = \overbrace{X_n - a_n g(X_n)}^{\text{"deterministic" part}} - \underbrace{a_n \left(G(X_n) - g(X_n)\right)}_{\text{"stochatsic" part}}. \tag{5}$$

As noted in the very readable book by Wasan [1969], the decomposition in (5) 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 variable and so this component is really not deterministic.) Accordingly, in order for the recursion in (5) 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 (4) 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 $G(x)$, e.g., $\mathrm{E}[(G(x) - g(x))^2] \leq \sigma^2 < \infty$ for all x, the stochastic part $a_n(G(X_n) - g(X_n))$ will disappear in the limit. (Although, this is not the way Robbins and Monro actually establish convergence.)

Blum [1954a] establishes a.s. convergence of the iterates in (4) under the same conditions on the sequence $\{a_n\}$ as Robbins and Monro [1951], and slightly weaker conditions on the structure of the root-finding function g. We now formally state this result.

Theorem 5.1 (Convergence of SA Methods in One Dimension). *Assume the following.*

A_1. $\sum_{n=1}^{\infty} a_n = \infty$.
A_2. $\sum_{n=1}^{\infty} a_n^2 < \infty$.
A_3. $|g(x)| \leq c(|x - x^*| + 1)$ *for all* x *and some* $c > 0$.

A_4. *Var*$(G(x)) \leq \sigma^2 < \infty$, *for all* x.

A_5. *the function* $g(x) < 0$ *for* $x < x^*$ *and* $g(x) > 0$ *for* $x > x^*$.

A_6. $\inf_{\delta_1 \leq |x-x^*| \leq \delta_2} |g(x)| > 0$ *for every pair of numbers* δ_1, δ_2 *satisfying* $0 < \delta_1 < \delta_2 < \infty$.

Then $\{X_n\} \to x^*$ *a.s. as* $n \to \infty$.

The assumptions A_1 through A_6 above are very similar in spirit to those assumed by Robbins and Monro. The conditions A_3, A_5 and A_6 impose structure on the underlying function g — A_3 imposes a maximum growth rate, A_5 ensures that there is at most one root, and A_6 ensures that the iterates do not accumulate at any finite point. These along with A_1 are directed at the deterministic part of the recursion in (5). The other conditions A_2 and A_4 are meant to drive the stochastic component in (5) to zero.

We emphasize that the results in Robbins and Monro [1951] and Blum [1954a] assume that $\{a_n\}$ is a sequence of positive real numbers. In other words, the sequence $\{a_n\}$ is fixed ahead of time and cannot be a function of the iterates $\{X_n\}$. This somewhat stringent assumption was relaxed by another seminal paper in 1956. Dvoretzky [1956] introduced a general class of iterations that includes (4), and demonstrated that it converges both in L_2 and a.s. Perhaps more importantly, he showed that such convergence is possible even when the sequence $\{a_n\}$ is chosen as a function of the iterates $\{X_n\}$, as long as certain mild conditions akin to A_1 and A_2 above are satisfied. Dvoretzky's result, commonly known as "Dvoretzky's approximation theorem," has been the precursor to SA type algorithms where the sequence $\{a_n\}$ is chosen adaptively. (We discuss such adaptive versions in a little more detail in Section 5.1.3, but one such early notable adaptive variation is that by Kesten [1958].) The interested reader is referred to the accounts in Lai [2003] and Wasan [1969].

All theory discussed thus far pertains to the one dimensional case. Blum [1954b] was the first to prove the a.s. convergence of the iteration in (4) 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.

Theorem 5.2 (Convergence of SA Methods in Multiple Dimensions). *Assume the following.*

A_1. $\sum_{n=1}^{\infty} a_n = \infty$.

A_2. $\sum_{n=1}^{\infty} a_n^2 < \infty$.

A_3. *There exists a positive-valued function* $f(x) : D \to I\!R$ *with unique minimum at* x^*, *and having continuous first and second partial derivatives such that*

(*a*) $\sup_{\epsilon \leq \|x-x^*\|} \nabla f(x)^T g(x) < 0$ *for all* $\epsilon > 0$, *where* $\nabla f(x)$ *is the column vector of first partial derivatives of the function* f *at* x, *and*

(*b*) $E[G(x)^T H(x) G(x)] < \infty$ *where* $H(x)$ *is the matrix of second partial derivatives of the function* f *at* x.

Then $\{\|X_n - x^*\|\} \to 0$ *a.s. as* $n \to \infty$.

The above conditions for convergence are much more intuitive than they appear at first sight. The first two assumptions A_1, A_2 are as before. The assumption $A_3(a)$ is essentially a multidimensional analogue of monotonicity meant to guarantee that SA's iterates are always moving in "roughly" the correct direction "on average." Specifically, since $\nabla f(x)$ points in the direction of steepest ascent, the condition $A_3(a)$ stipulates that $g(x)$ makes an acute angle with $\nabla f(x)$, and hence the observed vector $G(x)$ provides roughly the correct direction on average. The assumption $A_3(b)$ is a joint stipulation on the growth of $f(x)$ and the extent of the noise in $G(x)$. This is clear if we write $\mathrm{E}[G(x)^T H(x) G(x)] = g(x)^T H(x) g(x) + \mathrm{E}[(G(x) - g(x))^T H(x)(G(x) - g(x))] < \infty$, assuming of course that $\mathrm{E}[G(x)] = g(x)$. It should be seen as a unified condition that replaces conditions A_3 and A_4 in the one dimensional context.

As noted, much of the work between 1951 and the 1970s on proving the convergence of SA variants adopted a fairly predictable approach — decompose the SA iteration into "deterministic" and "stochastic" parts as demonstrated, prove that the stochastic part vanishes through assumptions on the growth/noise implicit in $G(x)$ combined with the convergence of $\sum_{n=1}^{\infty} a_n^2$, and prove that the deterministic part converges to a zero due to structural assumptions on g and the divergence of $\sum_n^{\infty} a_n$. This changed in 1977 when Ljung [1977a; 1977b] initiated a markedly different route to analyze SA type methods. Specifically, the modern method of analyzing SA type iterations usually follows four steps — (i) decompose the SA iteration into a certain "noise term" and a certain "mean term," and assume that the "noise term" is a martingale difference (see Doob [1953]); (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, pp. 102]) to show that the behavior of the mean term tends to a certain ordinary differential equation (ODE) as $n \to \infty$; and (iv) analyze the behavior of the ODE using traditional methods of Lyapunov stability (see Kushner and Yin [2003, Chapter 4]) to show that the iterates converge to a subset of the limit or invariant set for the ODE.

In detailing the above recipe, consider the following projected form of (4):

$$X_{n+1} = \Pi_H[X_n + a_n G(X_n)] = X_n + a_n G(X_n) + a_n Z_n, \tag{6}$$

where H is a compact set appropriately defined, $\Pi_H[x]$ is the projection of x into the set H, and Z_n is the correction term $\Pi_H[X_n + \epsilon_n G(X_n)] - X_n + a_n G(X_n)$. We now formally list a result on the convergence of SA type methods within the ODE context. (See Kushner and Yin [2003, pp. 132].)

Theorem 5.3 (Convergence of SA Methods Through ODE Method). *Assume the following.*

A_1. $\sum_{n=1}^{\infty} a_n = \infty, a_n \geq 0$.

A_2. $\sum_{n=1}^{\infty} a_n^2 < \infty$.

A_3. $\sup_n E[\|G(X_n)\|^2] < \infty$.

A_4. *There exist functions $g_n(\cdot)$ continuously uniform in n and random variables β_n such that*

(a) $E[G(X_n)|X_0, G(X_i), i < n] = g_n(X_n) + \beta_n$, *and*
(b) for each $x \in H$, $\lim_n \left\| \sum_{i=n}^{m_n(t)} a_i(g_i(x) - g(x)) \right\| = 0$ *for each* $t > 0$.

In (b) above, $m_n(t)$ *is the unique value of* m *satisfying* $\sum_{i=n}^{m} a_i < t < \sum_{i=n}^{m+1} a_i$.

A_5. $\beta_n \to 0$ *a.s.*

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

$$\dot{x} = g(x) + z, \quad z \in -C(x), \tag{7}$$

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

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_4(b) is the famous martingale noise assumption which implies that $G(X_n)$ can be written as

$$G(X_n) = g_n(X_n) + \delta M_n + \beta_n, \tag{8}$$

where the term δM_n is a martingale difference. In physical terms, 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 variable on the history of observations made thus far is embedded entirely within the current iterate X_n.

Plugging (8) in (6), the SA iteration becomes

$$X_{n+1} = X_n + a_n g_n(X_n) + a_n Z_n + \epsilon_n \delta M_n + a_n \beta_n. \tag{9}$$

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, in combination with assumption A_5, shows that the iterates in (9) reduce to that of a "mean" iteration involving just the fist three terms on the right-hand side of (9). 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 5.3. Specifically, under the assumptions $A_1, \ldots, A_5$, the iterates $\{X_{n+1}\}$ in (9) are such that the distance between X_n and a limit set of the ODE defined in Theorem 5.3 tends to zero as $n \to \infty$, a.s.

The assumptions on the sequence $\{a_n\}$ can be further relaxed. For example, the assertions regarding convergence remain the same if we require that A_1 and A_2 hold only a.s. More importantly, the stipulations A_2 and A_5 can be dispensed with and replaced by more direct and weaker assumptions on the last two terms appearing on the right-hand side of (9). It is shown in Kushner and Clark [1978], for instance, that A_1, A_2 and A_5 are not necessary if we assume that the rate of change of $\sum_{i=0}^{m_0(t)} a_i \delta M_i$ and $\sum_{i=0}^{m_0(t)} a_i \beta_i$ are zero as $t \to \infty$. (The definition of $m_0(t)$ is as explained in assumption A_4.) We do not go into further detail on this point but see Section 5.3 and Chapter 6 in Kushner and Yin [2003] for a precise statement, some treatment on verifiable sufficient conditions for the rate of change assumption to hold, and extensions to contexts where the noise in the estimator is correlated, e.g., due to a steady-state simulation.

5.1.2 *Convergence Rates.* The a.s. convergence of the SA algorithm ensures that the distance between the iterate X_n and the solution x^* converges to zero. However, this convergence gives no information about how fast the iterates approach the solution. The convergence rate can be obtained by establishing the asymptotic distribution for the iterate X_n. Since Chung [1954] and Sacks [1958] developed the first asymptotic normality results for SA methods, different types of convergence rate results have been introduced. 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 first start with the one-dimensional SA algorithm.

THEOREM 5.4 (CONVERGENCE RATES FOR SA IN ONE DIMENSION). *Consider the unconstrained SA algorithm* (4) *with* $q = 1$. *Assume that the assumptions for a.s. convergence hold. Furthermore, assume*

A_1. *g has a positive derivative* $g'(x^*)$ *at* x^*;

A_2. α *is in* $(1/2, 1]$, $n^\alpha a_n \to a$, *for some* $a > 0$, *and if* $\alpha = 1$, $a > \frac{1}{2g'(x^*)}$;

A_3. *the function* $\sigma^2(x) = Var(G(x))$ *is continuous at* x^*;

A_4. $\sup_{|x-x^*|\leq\epsilon} E[(G(x) - g(x))^2 I\{|G(x) - g(x)| > K\}] \to 0$ *as* $K \to \infty$, *for some* $\epsilon > 0$.

Then, $n^{\alpha/2}(X_n - x^*) \xrightarrow{d} N(0, s^2)$, *where* $N(0, s^2)$ *is a Gaussian random variable with mean zero and variance* s^2. *The asymptotic variance* $s^2 = a\sigma^2(x^*)/2g'(x^*) - 1$ *if* $\alpha = 1$, *and* $a\sigma^2(x^*)/2g'(x^*)$ *otherwise.*

The above theorem is a straightforward application of Theorem 2.2 in Fabian [1968]. The assumption A_1 means that g is strictly increasing in a neighborhood of x^*. 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_4 is to ensure that $\sup_{|x-x^*|\leq\epsilon} E[(G(x) - g(x))^{2+v}] < \infty$ for some $\epsilon > 0$ and some $v > 0$.

The above results show that the 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 the same result in a typical central limit theorem. The form of s^2 suggests that the asymptotic variance s^2 can be minimized with $a = 1/g'(x^*)$ and the corresponding minimum asymptotic variance is $\sigma^2(x^*)/(g'(x^*))^2$. 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. The asymptotic variance increases as the value of $g'(x^*)$ decreases. A small value of $g'(x^*)$ implies that the function g is flat in a neighborhood of x^*, which gives rise to a slow convergence of SA around the solution x^*. The sequence $\{a_n\}$ thus plays a crucial role in both asymptotic convergence and finite sample performance. The above results and insights easily extend to multidimensional problems through the following result.

THEOREM 5.5 (CONVERGENCE RATES FOR SA IN HIGH DIMENSIONS). *Consider the unconstrained SA algorithm* (4) *in multiple dimensions* ($q > 1$). *Assume that the assumptions for almost sure convergence hold. Furthermore, assume*

A_1. *g has a positive definite gradient matrix $\nabla g(x^*)$ 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 $\nabla g(x^*)$;*

A_3. *$\sup_{\|x-x^*\|\leq\epsilon} E[\|G(x) - g(x)\|^2 I\{\|G(x) - g(x)\| > K\}] \to 0$ as $K \to \infty$, for some $\epsilon > 0$;*

A_4. *the matrix function $\Sigma_x = Var(G(x))$ is continuous at x^*.*

Then, $n^{\alpha/2}(X_n - x^*) \xrightarrow{d} N(0, V(x^*))$, where $N(0, V(x^*))$ is a Gaussian random vector with mean zero and covariance matrix $V(x^*)$. The covariance matrix $V(x^*)$ depends on a_n, α, Σ_{x^*} and $\nabla g(x^*)$.

The proof of Theorem 5.5 and the explicit form of $V(x^*)$ can be found in Fabian [1968]. For a corresponding analysis through the ODE method, see Kushner and Yin [2003, Ch. 10].

Theorem 5.5 is significant because it provides insight into the choice of parameters within SA type methods. Specifically, it can be easily seen that the optimal rate of convergence is achieved by choosing $\alpha = 1$. Also, the norm of the covariance matrix $V(x^*)$ is minimized when $a_n = (n\nabla g(x^*))^{-1}$, thereby resulting in the optimal covariance $(\nabla g(x^*)^{-1})^T \Sigma_{x^*} \nabla g(x^*)^{-1}$.

In practice, however, since the gradient $\nabla g(x^*)$ 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 $\nabla g(x^*)$. 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 least squares estimation problems. The procedure is an analog of Newton's method applied to $\|g\|^2$. Unlike the classical SA method, this procedure does not necessarily attempt to move in a direction that decreases $\|X_n - x^*\|$, at least not during the initial stage of the procedure. It moves in a direction which decreases $\|g\|^2$, but the iterate eventually approaches the solution x^* as the algorithm evolves. At each iteration, this approach requires $O(q)$ function evaluations, which may not be feasible in a high-dimensional problem.

Adaptive SA methods specifically tailored for high dimensional problems became popular in the 1990s, particularly in the context of SOPs. In Spall [2000], the Hessian and gradients in SOPs are estimated using the now famous simultaneous perturbation methodology. In high-dimensional problems, such simultaneous changes admit an efficient implementation by greatly reducing the number of objective function evaluations or gradient evaluations. At each iteration, the Hessian estimates are projected onto the set of positive definite and symmetric matrices so that the algorithm converges to a minimum. The resulting estimators are strongly consistent under some regularity conditions. Bhatnagar [2005] extended this into

three-timescale simultaneous perturbation stochastic approximation (SPSA) algorithms. At each iteration, these algorithms update the Hessian and the gradients simultaneously, using no more than four samples. The resulting Hessian estimates can be easily modified to ensure symmetry.

One concern with adaptive SA methods in general is that the performance of the algorithms heavily depends on the choice of sequence $\{a_n\}$. In countering this, at least partially, Ruppert [1991] and Polyak and Juditsky [1992] showed that the averaged iterate $\bar{X}_n = \sum_{i=1}^{n} X_i/n$ is more robust than X_n. For certain types of problems, the finite-time performance of SA algorithms can be dramatically improved by such averaging. Juditsky et al. [2009] impressively demonstrate this on a class of convex SOPs, even with a constant step size. Similar results can be reasonably expected on SRFPs. From a theoretical standpoint, if $a_n = O(n^{\alpha})$ and $\alpha \in (1/2, 1)$, under some assumptions that are similar to those for a non-averaging estimator, the iterate average $\bar{X}_n$ achieves the optimal asymptotic convergence rate $O(1/\sqrt{n})$ without needing to know or estimate the gradient matrix. The limiting covariance matrix is identical to the covariance matrix based on the optimal choice of a_n in non-averaging SA algorithms. This result is quite impressive, in the sense that, by simply averaging the iterates, the algorithm can achieve the optimal convergence rate with the smallest possible covariance matrix.

5.1.3 *Implementation.* In principle, SA methods are easy to implement — choose the sequence $\{a_n\}$, and then use the recursion (4) 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. The asymptotic distribution results in Theorem 5.4 and Theorem 5.5 give us some insight into the choice of $\{a_n\}$. For instance, they suggest setting $a_n = a/(n + n_0)^{\alpha}$ with $a = 1, \alpha = 1$, resulting in the following SA iteration:

$$X_{n+1} = X_n - \frac{a}{n + n_0} U(X_n)^{-1} G(X_n), \tag{10}$$

where $U(X_n)$ is an estimator of the gradient $\nabla g(X_n)$. Interestingly, however, these asymptotically optimal choices do not result in uniformly good finite-time performance. For instance, 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^*. Considering this, Spall [2000] suggests using $a_n = a/(n + n_0)^{\alpha}$, where $\alpha \in (1/2, 1]$.

For the estimator $U(X_n)$ of the gradient matrix $\nabla g(X_n)$ appearing in (10), we suggest using Venter's scheme for mid to low-dimensional problems, and Spall's adaptive SPSA method for high dimensional problems. In illustrating this scheme, 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. 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

$$U(X_n) = \begin{cases} W_n, & \text{if } det(W_n) \neq 0; \\ I, & \text{if } det(W_n) = 0. \end{cases} \tag{11}$$

The step (11) is for coping with a possibility of the matrix W_n being nonsingular. (For adaptive SPSA methods, see Spall [2003, Ch 7].)

Wei [1987] proves that under a set of regularity conditions, X_n and $U(X_n)$ are consistent estimators and $\{X_n\}$ achieves the optimal convergence rate $O(1/\sqrt{n})$. This result shows that the asymptotic behavior of X_n is not affected by $U(X_n)$ as long as $U(X_n)$ is consistent. However, Free and Ruppert [1990] show 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 $\{c_n\}$ as well, 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 (Free and Ruppert [1990]). At each iteration, it might also be desirable to use the average of several, say m, replications of $U(X_n)$ to diminish the effect of noise.

With a view toward implementation, Figure 2 gathers the above recommendations and provides two 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 m; it returns the next (vector) seed and the estimator $G(x)$ realized using the chosen seed. In terminating simulations, the computational burden m is simply the implicit sample size chosen by the user when observing the estimator of g at a given x. We note that in some cases (see for instance Glasserman [1991]) the gradient estimator $U(x)$ is directly provided by the simulation oracle, in which case component P2 becomes redundant.

The following are three typical questions that a user might ask in implementing algorithms such as those listed in Figure 1.

Q.1 How should we choose the initial solution X_0?

Q.2 When should we terminate the algorithm?

Q.3 What if there is no root for the underlying system, i.e., the root-finding function $g(x)$ has no zero?

A simple answer to the question Q.1 is to choose a value of X_0 which is believed to be near the solution x^*. However, without much information about the underlying system, it may not be possible to make a reasonably good guess for the solution x^*. While there exists no rigorous answer for this question, one viable option might be running a standard first order SA or other rough optimization algorithm to obtain a solution close to x^*.

The SA algorithm listed in Figure 1 is a "non-terminating" version, i.e., the only guarantee that the algorithm provides is that X_n converges to a zero as $n \to \infty$ (in a certain precise sense). Given that during implementation the algorithm needs

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 $\nabla g(x)$ 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 computational burden 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 $X_{n+1} = \Pi_H(X_n - a_n U(X_n)^{-1} G(X_n))$.
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 might be reasonable.
– First order SA (first variation):

(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.

– Second order SA (second variation):

(i) estimate $U(x)$ using finite-difference like method 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 $\gamma = 1/4$.

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.

to be terminated in finite time, Q.2 is a very reasonable question to ask, and again for which there currently exists no rigorous answer. A heuristic that works well in practice is provided in Pasupathy and Schmeiser [2009]. To elaborate, suppose the algorithm has successfully completed k iterations, and the implementer is considering termination. Furthermore, for the opportunity to terminate, suppose the implementer is satisfied with a solution that is at most ϵ within a zero in the function-value space, with probability at least $1-\alpha$. More precisely, the solution X_k is acceptable if $\Pr\{\|g(X_k)\| \leq \epsilon\} \geq 1-\alpha$. The heuristic in Pasupathy and Schmeiser [2009] then assumes that $g(X_k)$ is the normal random variable $N(\mu, \Sigma)$, where μ and Σ are estimated appropriately. (When $G(X_k)$ is obtained as a sample average, the obvious estimators for μ and Σ are the sample mean and covariance respectively. As we detail in Section 6.1, the construction of an estimator for Σ in the context of SA type problems is frequently a nontrivial issue.) The algorithm is then terminated if $\Pr\{\|N(\mu, \Sigma)\| \leq \epsilon\} \geq 1-\alpha$, after calculating the probability $\Pr\{\|N(\mu, \Sigma)\| \leq \epsilon\}$ either numerically or using Monte Carlo.

We have much less to say on the last question Q.3. This question of detecting the feasibility of a system is difficult because it is impossible (in general) to differentiate a diverging sequence of iterates from a slowly converging sequence of iterates, merely through the observation of a finite number of iterates. The implementer is thus forced to resort to a heuristic to conclude infeasibility. For example, SA type variations such as those presented in Figure 1, when implemented on an infeasible problem, will repeatedly tend toward the boundary of the feasible region. Such behavior is a cue to the user to either expand the feasible region or conclude that the system under consideration is infeasible. A few other recommendations provided in Kelly [2006; 1995] for the context of DRFPs might be useful in practice.

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 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 root-finding function g by its consistent estimator G_m. The sample-path problem S_m thus has the following simple form:

$$\begin{array}{ll} \text{Find } x \text{ such that:} & G_m(x, \underline{\omega}(m)) = 0 \quad (S_m) \\ \text{subject to:} & x \in D \subset \mathbb{R}^q, \end{array}$$

where the notation $G_m(x, \underline{\omega}(m))$ is used to convey the fact (through an abuse of notation) that the root-finding function in (S_m) is a deterministic "$\underline{\omega}(m)$ realization" of the random function $G_m(x)$. For instance, if the function $g(x) = \mathrm{E}[G(x, \xi)]$, and $\xi_1, \xi_2, \ldots, \xi_m$ are independent and identically distributed (i.i.d.) realizations of the random variable ξ, then $\underline{\omega}(m) \equiv (\xi_1, \xi_2, \ldots, \xi_m)$ and $G_m(x, \underline{\omega}(m)) = \frac{1}{m}\sum_{i=1}^{m} G(x, \xi_i)$.

Also, we emphasize that like much of numerical optimization, the root-finding function in (S_m) is not "constructed" ahead of time. Instead, the value of the function $G_m(x, \underline{\omega}(m))$ at any x is "revealed" (only) when the search routine visits x.

The appeal of SP methods is two fold. First, the method is conceptually very simple — substitute the underlying root-finding 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. This arguably allows SP methods to be relatively free from parameter tuning problems that SA type methods are widely known to face.

5.2.1 *Basic Convergence Results.* Unlike SA type methods where convergence has generally been in the design space (i.e., distance between the iterate and a zero of the root-finding 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 variable X_m^* is a zero of the random function $G_m(x)$, and assume that X_m^* is well-defined and measurable. We might then ask under what conditions do the sequences $\{\mathbb{H}(X_m^*, \pi^*)\}$ and $\{\mathbb{H}(g(X_m^*), 0)\}$ go to zero a.s.? The answers to both 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 [2009].

Theorem 5.6 (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 non-empty;*

A_2. *the functional sequence $\{G_m(x)\}$ is such that the set of zeros Π_m^* of the function G_m is non-zero 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. (Assume $\Delta_m = \infty$ if $\Pi_m^* = \emptyset$.)

The proof of the above theorem is a straightforward application of the definition of uniform convergence. Note that no assumptions about the continuity of g have been made. 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 $G_m(x) = -(2m + x)I_{(-\infty,-m]}(x) + xI_{(-m,\infty)}(x) - 1$ 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 5.6 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 root-finding function g, specifically that D is compact and g is continuous.

Theorem 5.7 (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 non-empty;*

A_2. *the functional sequence* $\{G_m(x)\}$ *is such that the set of zeros* Π_m^* *of the function* G_m *is non-zero 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.,* $\mathbb{H}(\Pi_m^*, \pi^*) \to 0$ *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, pp. 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 5.6, 5.7 match the corresponding theorems in the SOP context, e.g., Shapiro [2004].

5.2.2 *Convergence Rates.* In discussing the convergence rates that are obtainable using SP methods, we first list a basic result that ties the convergence rate of the available estimator $G_m(x)$ to the rate at which solutions (or their function values) converge to their limit. Specifically, Theorem 5.8 asserts that in SP methods, under the conditions laid down in Theorems 5.6 and 5.7 respectively, convergence in the function-value space and the design space simply inherit the rate exhibited by the estimator $G_m(x)$ in use within the algorithm. The result is canonical in the sense that it assumes nothing other than the continuity of the function g, and the uniform convergence of its estimator. A corresponding theorem for the SOP context can be found in Pflug [1996].

Theorem 5.8 (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 non-empty as* $m \to \infty$ *a.s.*

A_2. $\beta(m)$ *is a (deterministic) function satisfying* $\lim_{m\to\infty} \beta(m) = \infty$ *such that the supremum-error sequence* $\{\sup_x \|G_m(x) - g(x)\|\}$ *is* $O_p\left(\beta(m)^{-1}\right)$*, i.e., for* $\epsilon > 0$*, there exists* K_ϵ *such that* $\sup_m \Pr\{\beta(m) \sup_x \|G_m(x) - g(x)\| \geq K_\epsilon\} \leq \epsilon$.

A_3. *There exists* $s > 0$ *such that* $\|g(x)\| \geq s\mathbb{H}(x, \pi^*)$ *for all* $x \in D$.

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

The proof for $\{\sup_{x\in\Pi_m^*} \|g(x)\|\}$ being $O_p\left(\beta(m)^{-1}\right)$ follows upon observing that $\sup_{x\in\Pi_m^*}\{\|g(x)\|\} = \sup_{x\in\Pi_m^*}\{\|G_m(x) - g(x)\|\} \leq \sup_{x\in D} \|G_m(x) - g(x)\|$. The assumption A_3 imposes a "minimum growth rate" on the norm of the root-finding function g, which in combination with A_1 and A_2 guarantees the convergence rate in the function-value space.

Something more specific than Theorem 5.8 can be said if we make further assumptions about the estimator G_m, the root-finding function g, and the numerical

procedure in use within the SP framework. For instance, suppose a chosen numerical procedure is used within the SP framework to obtain a sample-path solution X_m^*. Suppose further that the sequence $\{X_m^*\}$ is such that it converges weakly to a random variable X_∞. Intuitively, the random variable X_∞ describes the limiting rate at which the different zeros within π^* are attained by the numerical procedure in use. If we also assume that a central limit theorem (CLT) holds for the estimator G_m, then the solutions obtained $\{X_m^*\}$ and their function values follow a certain general form of a CLT.

THEOREM 5.9 (A CLT FOR SP METHODS). *Assume that the following conditions hold.*

A_1. *A CLT holds for $G_m(x)$, i.e., $\sqrt{m}\,(G_m(x) - g(x)) \xrightarrow{d} N(0, \Sigma_x)$, where $N(0, \Sigma_x)$ is the Gaussian random variable with mean zero and covariance Σ_x.*

A_2. *The functions $G_m(x)$ and $g(x)$ have nonzero gradients $\nabla G_m(x), \nabla g(x)$ in some neighborhood around x^* a.s., for each $x^* \in \pi^*$.*

A_3. *The sequence $\{\nabla G_m(x)\}$ converges uniformly to $\nabla g(x)$ in some neighborhood around x^* a.s., for each $x^* \in \pi^*$.*

A_4. $X_m^* \xrightarrow{d} X_\infty$.

Let μ_m, ν_m, μ_∞ denote the probability measures of $\sqrt{m}\,(X_m^* - E[X_\infty])$, $\sqrt{m}\,g(X_m^*)$, and X_∞ respectively, and let $\phi(z_1, z_2)$ denote the density function of a Gaussian random variable with mean z_1 and covariance z_2. Then

(i) $\mu_m(A) \to \int_A \phi(x, \nabla G_m(x)^{-1} \Sigma_x (\nabla G_m(x)^{-1})^T) \mu_\infty(dx)$;
(ii) $\nu_m(A) \to \int_A \phi(0, \Sigma_x) \mu_\infty(dx)$.

Theorem 5.9 essentially states that, under mild conditions, the limiting distribution of X_m^* is an X_∞-mixture of normal random variables. Since X_∞ satisfies $g(X_\infty) = 0$ by definition, $g(X_m^*)$ turns out to be a mixture of multivariate normals, each of which has mean 0. In other words, Theorem 5.9 states that as m becomes larger, X_m^* looks like a random draw from a multivariate normal distribution that is centered on a point in π^*, chosen according to the random variable X_∞. Similarly, $g(X_m^*)$ looks like a random draw from a multivariate normal distribution that is centered on 0, but with covariance that is dependent on the structure of g at a point (in π^*) that is chosen according to the random variable X_∞. Theorem 5.9 is the *general central limit theorem* for stochastic root finding. It parallels Theorem 10 in Shapiro [2004], where the limiting optimal value of the SOP turns out to be the *infimum of a certain set of multivariate normals*, each of which is centered on the optimal value of the limiting problem. A proof for Theorem 5.9 follows along the lines of the more restrictive Theorem 4 in Pasupathy [2009].

From an implementation standpoint, a natural question to ask 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 obtained X_m^* satisfies $\Pr\{\|g(X_m^*)\| \geq \epsilon\} \leq \alpha$ for given ϵ, α? Theorem 5.10 answers this question. Similar to Shapiro [2004], the proof follows from simple probability arguments based on the Bonferroni's inequality [1998], the assumption of a large-deviation law, and a Taylor's series expansion of the moment-generating function of the estimator $G_m(x)$.

THEOREM 5.10 (MINIMUM SAMPLE SIZE IN SP METHODS). *Let $D \subset \mathbb{R}^q$ be a finite set. Define the sets $\pi^*(\epsilon) = \{x \in D : \|g(x)\| \leq \epsilon\}, \Pi^*_m(\delta) = \{x \in D : \|G_m(x)\| \leq \delta\}$, and $c(\delta) = \{x \in \mathbb{R}^q : |x^1| \leq \delta_1, |x^2| \leq \delta_2, \ldots, |x^q| \leq \delta_q\}$. Then*

(i) $\Pr\{\Pi^*_m(\delta) \nsubseteq \pi^*(\epsilon)\} \leq |D| \exp(-n\tau(\delta, \epsilon))$ *where*

$$\tau(\delta, \epsilon) = \mathrm{Min}_{x \in \mathfrak{D} \setminus \pi^*(\epsilon)} \inf_{z \in \partial c(\delta)} I(z);$$

(ii) Let Y_i, $i = 1, 2, \ldots, m$ be i.i.d. random variables and $G_m(x) = m^{-1} \sum_{i=1}^m Y_i(x)$. Assume that there exist a vector of positive constants $\sigma = [\sigma^1, \sigma^2, \ldots, \sigma^q]^T$, and a $q \times q$ correlation matrix ρ, satisfying $M(t) \leq (1/2)\sigma\rho\sigma^T \quad \forall t \in \mathbb{R}^q$, where $M(t)$ is the moment-generating function of $Y_i - \mathrm{E}[Y_i]$. Then,

$$m \geq \frac{2\left(\mathrm{Max}(\sigma^1, \sigma^2, \ldots, \sigma^q)\right)^2 \left(1 + \sum_{i \neq j} \rho_{ij}\right)}{(\epsilon - \delta)^T (\epsilon - \delta)} \log \frac{|D|}{\alpha}$$

implies $\Pr\{\Pi^*_m(\delta) \nsubseteq \pi^*(\epsilon)\} \leq \alpha$.

The minimum sample size expression in Theorem 5.10 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 5.10 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 [2004, pp. 375]. We emphasize that a limitation of Theorem 5.10 is that it applies only to estimators $G_m(x)$ that are constructed as the sample mean of i.i.d random variables.

5.2.3 *Implementation.* In practice, SP methods are only infrequently implemented in their generic form, i.e., where a *single* sample-path problem with a "large enough" sample size is generated, and then solved to a prescribed error tolerance. This is because the sample size required to obtain a stipulated quality of solution (e.g., through Theorem 5.10) often tends to be impractically large. Furthermore, even if the stipulated sample size is manageable, the idea of solving a single deterministic approximation, while simple, tends to be suboptimal from the standpoint of computational efficiency.

The need for implementability and efficiency has led to the construction of refinements of generic SP. 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, and then solve these to progressively decreasing error tolerances. This elaborate structure in the SP refinements is explicitly constructed to gain overall efficiency. The early iterations are efficient, in principle, because the small sample sizes ensure that not much computing effort is expended in generating sample-path problems. The later iterations are efficient, again in principle, because the starting solution for the sample-path problem is probably close to the true solution, and not much effort is expended in solving sample-path problems. The solving of the individual sample-path problems, as in generic SP, is accomplished by choosing any numerical procedure from amongst the powerful DRFP techniques that are widely available.

The most notable amongst the said SP refinements is called Retrospective Approximation (RA). It was first designed in the context of one-dimensional SRFPs by Chen and Schmeiser [2001], and subsequently extended to multiple dimensions by Pasupathy and Schmeiser [2009]. A similar refinement for SOPs called *variable sample methods* is introduced and analyzed in detail by Homem-de-Mello [2003].

With a view toward implementation, Figure 2 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 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" during Step 1 of the algorithm logic, thereby preserving any structural properties inherent in the sample-paths $G_m(x)$. For more on common random numbers, see Law [2007] or Bratley et al. [1987]. As Figure 2 notes, readily implementable RA code (that uses the Newton-Krylov method) is available through `<https://filebox.vt.edu/users/pasupath/pasupath.htm>`. When implementing an RA algorithm of the sort provided in Figure 2, the complications encountered are almost identical to those during the implementation of SA type algorithms. Accordingly, our comments from Section 5.1.3 apply here as well, and we will not go into further detail.

5.3 PARAMETRIC AND SEMI-PARAMETRIC METHODS

The SA and SP methods are nonparametric procedures, in the sense that they do not assume a parametric form for the root-finding function g or its estimator G. By contrast, parametric and semi-parametric (PSP) methods assume a parametric form for g and/or G to varying levels of stringency. For instance, a simple linear regression model or other type of parametric model based on the assumed distribution of $G(x)$ may first be assumed to approximate $g(x)$. The model parameters appearing in the assumed form are then estimated using observed data $(X_1, G(X_1)), ..., (X_n, G(X_n))$. The constructed approximation is then used to identify an approximation X_{n+1} of a zero of g. At this stage, either the procedure is terminated and the current solution X_{n+1} returned, or additional data is collected to further update the approximation. PSP methods are relatively very new, and accordingly have seen much less development. Nevertheless, and unsurprisingly, they have been demonstrated to be very efficient in settings where the assumed parametric form is reasonable.

To illustrate PSP methods in further detail, consider a simple linear regression model to approximate g. For notational convenience, let $Y = G(X)$. Assume that X and Y are linearly related, that is, $Y = B(X - x^*) + \epsilon$, where B is a $q \times q$ known matrix, and the error ϵ has mean zero. Let $\hat{B}_n$ be the least squares estimator of B based on n observations. Then we can estimate the solution x^* by using $\bar{X}_n - \hat{B}_n^{-1}\bar{Y}_n$, where $\bar{X}_n$ and $\bar{Y}_n$ are, respectively, the sample means of X_i and Y_i,

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 $\{w_k\}$ 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 $G_m(x, \omega_0)$ 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 $G_m(x, \omega_0) = 0$ 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., $\overline{X}_k = (\sum_{i=1}^k w_k)^{-1} \sum_{i=1}^k w_k X_k$.
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 $G_{m_k}(x, \omega_0) \leq \epsilon_k$
– After k iterations, set $w_i = m_i$ for all $i \leq k$; or set $w_i = 0$ for $i < k$ and $w_k = 1$ for $i = k$.

Implementable Code Location:

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

Fig. 2. A self-contained illustration of RA algorithms for solving SRFPs, along with parameters required from the user, RA logic, and recommended parameter values.

$i = 1, 2, ..., n$. Then it follows that

$$X_{n+1} = \bar{X}_n - \hat{B}_n^{-1}\bar{Y}_n = X_n - \frac{\hat{B}_n^{-1}}{n}Y_n,$$

which is identical to the adaptive SA recursion form given by (10). Wei [1987] shows that under the condition given in Theorem 5.5, this estimator enjoys the optimal convergence rate, and its asymptotic covariance matrix is $(B^{-1})^T\Sigma_{x^*}B^{-1}$.

Wu [1985] proposes a similar parametric sequential procedure based on the maximum likelihood estimation (MLE) method. The essential idea is to approximate the function $g(x)$ using a linear parametric function $F(x|\gamma)$, where the parameter γ is estimated via the MLE method. If X is normally distributed, this MLE-based procedure is identical to the adaptive SA method that is based on least squares estimation. When the choice of F is correct, such a model exhibits the same asymptotic behavior as the SA algorithms with the optimal sequence $\{a_n\}$. Wu [1986] extends the MLE approach to generalized linear models (GLM), resulting in more flexible parametric models for the function F. Ying and Wu [1997] showed that such MLE-based sequential approaches generate a sequence of solutions $\{X_n\}$ that converge to the true solution x^*, regardless of the selection of the parametric function F. However, this asymptotic result often cannot be realized in practice if F is not a good approximation to g. The procedure assumes a linear relationship between X and Y (or through a link function in GLM), and therefore it assigns equal weights to all the observations. If X and Y are nonlinearly related, the convergence rate of the MLE estimator can deteriorate. The simulation study by Free and Ruppert [1990] shows that the finite sample performance of the procedure can be heavily affected by an improper choice of F.

To overcome the model uncertainties in F, Joseph et al. [2007] propose a Bayesian approach by imputing a prior on Y. Particularly, they assume that the conditional distribution of Y given X is Gaussian with mean $\beta(X - \theta)$, and use a Gaussian process to introduce randomness into the model. Formally, the form of the function F for one-dimensional SRFPs is

$$Y = F(X|\gamma = (\beta, \sigma, \tau, \theta, \lambda)) = (\beta + e(X))(X - \theta) + \epsilon,$$

where the error ϵ is a Gaussian random variable with mean zero and variance σ^2, and the random correlated error process $e(X)$ is a mean zero Gaussian process with $cov(e(X_i), e(X_j)) = \tau^2 R_{ij}$. The correlation function is given by $R_{ij} = exp(-\lambda|X_i - X_j|^p)$, where $\lambda > 0$ and $0 < p \leq 2$. It turns out that estimating γ using the MLE method can be difficult or infeasible due to the complexity of the MLE objective function. Joseph et al. [2007] address this issue using a Bayesian estimation method to estimate the solution x^*. The model used in Joseph et al. [2007] is very general in the sense that both SA and Wu's procedures can be obtained as special cases by setting particular values of the parameters in the model. Furthermore, Joseph et al. [2007] demonstrate that as the iteration number grows, the model tends to give more weightage to the observations closer to the root. This results in a better local fit around the root, and consequently faster convergence than other parametric and nonparametric schemes.

Although the Bayesian approach by Joseph et al. [2007] has merit, much more research is needed to understand various issues. First, the procedure applies only

to one-dimensional problems, and the asymptotic behavior of the procedure is not yet fully understood. Second, compared to the MLE method, the implementation of the Bayesian estimation method hinges on the choosing of a proper prior distribution and the introduction of model uncertainty through a random correlated error process $e(X)$. Third, as mentioned in Joseph et al. [2007], extension to non-normal prior distributions is a challenging problem. By contrast, the theory for MLE-based models has been exhaustively studied within a frequentist setting, and also as a special case of the adaptive SA method.

6. FUTURE RESEARCH AGENDA

In this section, we discuss three areas within SRFPs where we believe advances in the current state of knowledge are much needed. The reader will notice that we have favored a fairly-detailed and nontrivial discussion of a small number of open and pressing problems, over a longer list that potentially sacrifices such detail. The questions appear in no particular order.

6.1 Finite-Time Stopping for Implementation

A key question that currently remains largely unresolved is that of the finite stopping of algorithms for solving SRFPs. Specifically, let us suppose that X_m is the solution returned by an algorithm (e.g., SA or RA) after expending a total amount of effort m in attempting to solve an SRFP. Virtually all of the algorithms we have discussed in this paper are concerned with whether, and the manner in which, X_m converges to a zero of the root-finding function g. For example, in the context of SA type algorithms, we have presented various results that lay down the conditions that guarantee the a.s. convergence of $\{X_m\}$ to the solution set π^*, along with the rate at which such convergence happens. Similar conditions have also been presented for the context of SP methods, in both the domain space and in the function-value space. While these results are of use in obtaining a sense of the asymptotic efficiency, they provide no directives on when to stop an algorithm. Such finite-stopping directives are critical from the standpoint of implementation, since a user cannot wait "until the algorithm converges."

Let us now pose such a finite-time stopping 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 $\Pr\{|g(X_m)| > \epsilon\} \leq \alpha$. (See Section 4 for the meaning of $|x|$ when x is a vector.) 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.

In the context of SP methods, the above question was addressed in a "static" manner through Theorem 5.10 where an expression for the minimum sample size m^* was derived so that X_{m^*} is an (ϵ, α) solution to the SRFP. While mathematically correct, the minimum sample size derived using Theorem 5.10 tends to be so conservative as to be impractical even in simple cases. This is understandable considering that the expression for m^* was obtained using Bonferroni's inequality applied not just to the region around a zero, but over the entire domain D. The

minimum sample size result in Theorem 5.10 then begs the question: "is there a practically implementable sequential stopping rule for SP methods, akin to those that exist for constructing confidence intervals, which can guarantee the attainment of an (ϵ, α) solution?" This question has been tackled with limited success for SOPs by Bayraksan and Morton [2009]. The corresponding problem in our context actually seems easier, due to the fact that the key quantity $|g(X_m)|$ is actually "estimated" by the observable quantity $|G_m(X_m)|$!

Let us delve a little further into the above point — the finite-time stopping in the context of SP methods. Under certain conditions, we have seen that $g(X_m)$ asymptotically tends to a mixture of normal random variables, each of which is centered on 0. This fact implies that a reasonable "guess" of the "location" of $g(X_m)$ is the hyperrectangle $G_m(X_m) \pm a_m\hat{\sigma}_m$ (provided the constants $\{a_m\}$ are appropriately chosen), where $\hat{\sigma}_m$ is the estimated standard error (matrix) of $G_m(X_m)$. This suggests stopping the algorithm when the constructed hyperrectangle lies completely within the hyperrectangle centered at the origin and having sides given by the vector 2ϵ. So, terminate the algorithm when

$$|G_m(X_m) \pm a_m I\hat{\sigma}_m| \leq \epsilon, \tag{12}$$

where $\{a_m\}$ is an appropriately chosen sequence of constants, and I is the $q \times q$ identity matrix. When using an implementable variation of SP such as RA, the stopping rule in (12) becomes

$$|G_{m_k}(X_{m_k}) \pm a_k I\hat{\sigma}_{m_k}| \leq \epsilon. \tag{13}$$

While intuitive, the behavior of the stopping rule in (12) is still unclear. For instance,

(a) is the expected time at stopping finite?

(b) under what conditions is the solution obtained upon stopping X_{k^*} *consistent*, i.e., an (ϵ, α) solution? and

(c) can the deterioration in coverage be characterized rigorously?

All of the above 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 those available for sequential confidence intervals can be derived.

While the discussion of the specifics in the last three paragraphs pertain to SP methods, a similar analysis seems possible for SA type algorithms as well. The first (and only such) seems to be the work by Hsieh and Glynn [2002] in the context of constructing confidence regions for a (unique) zero in SRFPs (or a unique global minimizer in SOPs), when using an SA type algorithm. The generic $100(1-\alpha)$ percent confidence region is based on the well-known CLT for SA type methods (see for example Nevel'son and Khas'minskii [1973]), and has the form

$$\{x : m(X_m - x)C_m^{-1}(X_m - x)^T \leq z\}, \tag{14}$$

where C_m is the assumed estimator of the asymptotic covariance matrix C, and z satisfies $\Pr\{\chi_q^2 > z\} = \alpha$. (Hsieh and Glynn [2002] actually propose an alternative to the above generic form in order to circumvent the difficult problem of estimating C.)

Various important questions pertaining to (14) arise in the current context, where a (ϵ, α) solution is desired.

(a) How can the confidence interval procedure proposed by Hsieh and Glynn [2002] be modified to provide a procedure that guarantees a fixed width (or fixed volume) region such as that stipulated by a (ϵ, α) solution?

(b) What if the root-finding function g has multiple zeros, as is often the case?

(c) How much deterioration in coverage does the procedure resulting from answering (a) and/or (b) suffer?

The answer to (a) seems direct and will simply involve a sequential procedure that progressively estimates the confidence region until the volume falls below the stipulation. Such procedures are shown to be consistent in great generality by Glynn and Whitt [1992]. The questions in (b) and (c) will pose much greater challenges. For instance, when multiple zeros are present, Hsieh and Glynn's revised procedure no longer holds because it relies on random restarts of the SA type algorithm converging to the same zero. In such a case, since all zeros $x^* \in \pi^*$ satisfy $g(x^*) = 0$ by definition, perhaps the solution is to simply construct the required sets in the function-value space rather than the domain space. This will however re-introduce the difficult problem of having to estimate the constant C.

Answering the question in (c) is key for practical implementation. A fine second-order analysis akin to that by Woodroofe [1976] for sequential confidence intervals seems like a viable route. Such an analysis will establish the rate at which the degradation in coverage disappears as a function of effort m. At the minimum, such an analysis will provide a sense of whether the proposed sequential procedures are actually implementable.

6.2 Super-Quadratic SA Type Iterative Schemes?

In the last decade or so, there has been a tremendous revival of attention given to methods for solving DRFPs. Such attention can be traced back to the paper by Weerakoon and Fernando [2000] which presented a very simple modification of the generic Newton's method to achieve cubic convergence. In illustrating their simple (but revolutionary) idea, suppose we seek a zero of the real-valued function $g : \mathbb{R} \to \mathbb{R}$. Assuming that g is differentiable everywhere in $\mathbb{R}$, and using the fundamental theorem of calculus (see for example Royden [1988]), we write

$$g(x) = g(x_n) + \int_{x_n}^{x} g'(y)\, dy, \tag{15}$$

where $x, x_n \in \mathbb{R}$. Notice that the second term on the right-hand side of (15) is the area $\int_{x_n}^{x} g'(y)\, dy$ under the derivative $g'(y)$ between x_n and x. With the aim of obtaining an iterative procedure, suppose we choose $x = x^*$, where x^* is a zero of the function g, and approximate the area under $g'(y)$ between x_n and x^* by the rectangular area $A_n(x^*) = g'(x_n)(x^* - x_n)$. Plugging $x = x^*$ in (15), approximating $\int_{x_n}^{x} g'(y)\, dy$ by A_n, and then "solving" for x^* gives $x^* \approx x_n - (g'(x_n))^{-1} g(x_n)$. This prompts the recursion

$$x_{n+1} = x_n - \frac{1}{g'(x_n)} g(x_n). \tag{16}$$

The reader would recognize the recursion in (16) as the well-known Newton's recursion but arrived at through a very different route.

Noting that key to the above method is the approximation of $\int_{x_n}^{x} g'(y)\,dy$ by the rectangular area A_n, one is naturally led to ask if other more accurate approximations of $\int_{x_n}^{x} g'(y)\,dy$ can be used to obtain recursions that are more efficient than the generic Newton's recursion. For instance, using the midpoint rule and approximating the area $\int_{x_n}^{x} g'(y)\,dy$ as $A_n(x) = g((x+x_n)/2)(x-x_n)$ gives the modified recursion

$$y_n = x_n - \frac{g(x_n)}{2g'(x_n)}; \quad x_{n+1} = x_n - \frac{1}{g'(y_n)} g(x_n). \tag{17}$$

Using the trapezoidal rule to approximate the needed area results in the recursion

$$y_n = x_n - \frac{g(x_n)}{g'(x_n)}; \quad x_{n+1} = x_n - \frac{2}{g'(y_n) + g(x_n)} g(x_n). \tag{18}$$

Quite remarkably, both recursions (17) and (18) exhibit cubic convergence! (See for example, Weerakoon and Fernando [2000], Homeier [2003], and Frontini and Sormani [2003].) Recall that the generic Newton's method exhibits only quadratic convergence, although it uses one less observation on the gradient of g.

The above development has led to a flurry of recent activity in this area, resulting in a large number of variations of Newton's iteration, and having orders of convergence much faster than quadratic. The references in this respect are too numerous to list — see Abbasbandy [2003], Noor et al. [2007], Noor and Noor [2007b; 2006; 2007a], Homeier [2003], and Darvishi and Barati [2007] for more details and references. While a number of these constitute a variation in the way the integral $\int_{x_n}^{x} g'(y)\,dy$ is approximated, several of these references also involve entirely new methods that are based on novel decompositions and quadrature. Important, however, is the fact that a substantial fraction of these new methods require the computation of the second derivative of the root-finding function g. This turns out to be computationally expensive even when g is directly observable ($O(q^2)$ evaluations of g). For this reason, we focus our attention only on methods in the genre of that presented by Weerakoon and Fernando [2000].

The natural extension of iteration (17) to SRFPs (in one dimension) is

$$Y_n = X_n - \frac{a_n}{2G'(X_n)} G(X_n); \quad X_{n+1} = X_n - \frac{b_n}{G'(Y_n)} G(X_n), \tag{19}$$

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 as

$$X_{n+1} = X_n - b_n h_G \left(X_n - \frac{a_n}{2} h_G(X_n) G(X_n) \right) G(X_n), \tag{20}$$

where $h_G(x) = (\nabla G(x))^{-1}$. It is very likely that the conditions needed for the convergence of the iteration in (20) to a valid zero of g will be very similar to those required by corresponding iterations such as SPSA (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. The more interesting question is whether any gains in convergence rates, over corresponding SA type iterations are obtainable using the modified iteration.

If the massive gains (second order convergence to third order convergence) realized in the deterministic context are transferrable at least in part, the modification outlined in (20) is surely worthy of further study.

A key drawback of the iteration proposed in (20) is that it assumes that the gradient $\nabla G(x)$ is "realizable" along with the estimator itself. While this is true in many instances [Glasserman 1991], it is not always the case. In cases where $\nabla G(x)$ is not directly available, we are forced to further modify (20) by estimating the gradient-inverse of g using the estimator G. The current knowledge on this topic is vast — see Fu [2006] for an entry point into the literature.

The iteration in (20) requires observation of the estimator of g at a single location, and the observation of the gradient-estimator of g at two locations. Of course, whether or not these observations are computationally equivalent will depend on the nature of the gradient estimator in use. To this extent, an analysis of iteration (20) that explicitly takes into account the total computational burden will be insightful. (This is in contrast to generic convergence rate results, where the iteration number n is the sole measure of computational burden.) For example, suppose $c(q, n)$ measured appropriate units is the total computational burden of a single iteration in (20), i.e., of making an observation of the estimator of g at one location and it's gradient estimator at two locations. Then, analyzing the behavior of the sequences of random variables $\{(\sum_{i=1}^{n} c(q,i))^r \|g(X_n)\|\}$ and $\{(\sum_{i=1}^{n} c(q,i))^r \mathbb{H}(\pi^*, \{X_n\}\}$ will lend useful insight into whether such modifications as (20) are actually more efficient than those already available. A simple analysis of this sort for the deterministic case appears in Frontini and Sormani [Frontini and Sormani 2003], using the efficiency index suggested in Gautschi [1997]. Similar analyses in the SP context appear in Pasupathy [Pasupathy and Schmeiser 2009].

6.3 Approximating All Roots of a "Stochastic" Function

By "approximating all roots of a 'stochastic' function," we mean an SRFP where all roots of the function g are sought. This situation is frequently encountered in the context of fixed-point problems representing physical equilibria. For example, suppose $g(x) = h(x) - x$, where $x \in \mathbb{R}^q$ represents the "conditions" in a system and $h(x) : \mathbb{R}^q \to \mathbb{R}^q$ represents the conditions that result from x. To provide a concrete example, x might be the flows on a traffic network, and $h(x)$ might be the flows in the traffic network that result from drivers learning from their past experience and other drivers' choices. So, the fixed-point problem "find x such that $h(x) = x$" represents the identification of the conditions at traffic equilibrium, i.e., characterizing steady-state conditions where no further "evolution" of traffic flows need to occur. In such a scenario, a planner would like to know all solutions to the fixed-point problem $h(x) = x$ because each of these is an equilibrium condition to which the system might evolve. Similar situations are common in the study of the long-term evolution of economies, populations, or chemical processes.

For clarity, let us pose the above question rigorously.

Given: A simulation capable of generating, 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: All zeros π^* of g, i.e., find the set $\pi^* \subset D$ satisfying $g(x) = 0$ if $x \in \pi^*$, and

$g(x) \neq 0$ if $x \notin \pi^*, x \in D$, assuming that π^* is non-empty and finite.

A method to solve the above problem would thus return a random set Π_t of vectors lying in D, after expending a total amount of effort t on the problem. We seek methods that identify the true set π^* as the expended effort increases to infinity, i.e., methods that ensure that the distance $\mathbb{H}(\Pi_t, \pi^*) \to 0$ as $t \to \infty$. Furthermore, good methods would ensure that such convergence happens at a fast rate.

Constructing an SP type algorithm to solve the above problem is conceptually simple:

1. randomly generate a sample-path of g using a sample-size m;
2. randomly generate an initial guess using a sampling distribution on D;
3. using the generated initial guess, obtain an approximation of a root by executing a locally convergent algorithm;
4. accumulate the solution set Π_t with the solution obtained in Step 3, and go back to Step 2.

It seems likely that under very mild assumptions on the structure of the function g, and the sampling distribution in use in Step 2, the "algorithm" outlined above will ensure that $\mathbb{H}(\Pi_t, \pi^*) \to 0$ as $t \to \infty$. The more interesting questions thus pertain to (i) the tradeoff between the sample-size used in Step 1 and the number of restarts executed using Step 2; and (ii) the strategic choice of the sampling distribution in Step 3.

Towards further exposition of the key questions that arise in the context of the above procedure, let us introduce some notation: (i) $X_0 \in D$ is a random vector denoting the initial guess and having the distribution $F_{X_0}(\cdot)$; (ii) r denotes the number of restarts of the numerical procedure in Step 2; (iii) $X^*_m(X_0)$ denotes the root obtained using the locally convergent DRFP algorithm, expressed explicitly as a function of X_0; (iv) $X_0^1, X_0^2, \ldots, X_0^r$ are r i.i.d copies of the random variable X_0, and $\mathbf{X_0}(\mathbf{r}) = (X_0^1, X_0^2, \ldots, X_0^r)$; (v) the function $a(x) : D \to \pi^*$ denotes the local minimum that will be attained by the algorithm in use when executed on the function g; (vi) $N(x, m)$ is a random variable denoting the number of steps taken by the algorithm when executed on the random function G_m with the initial guess $x \in D$. When $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ with $x_i \in D$, through an abuse of notation we write $a(\mathbf{x})$ to mean the set $\{a(x_1), a(x_2), \ldots, a(x_k)\}$. We note that the notation introduced in (v) makes implicit assumptions about the nature of the algorithm in use in Step 3. Specifically, the algorithm in use is "deterministic" in the sense that for a given initial guess and a given root-finding function, the evolution of the algorithm remains the same. Second, the function g and the algorithm in use are such that every initial guess results in the algorithm evolving to some solution in π^*.

Given the above, we can obtain an upper bound on the expected Hausdorff distance between the true solution set π^* and the solution set Π_t obtained after expending t units of computing effort.

$$\mathrm{E}[\mathbb{H}(\pi^*, \Pi_t)] \leq \mathrm{E}[\mathbb{H}(\pi^*, \{a(\mathbf{X_0}(\mathbf{r}))\}] + \mathrm{E}[\mathbb{H}(\{a(\mathbf{X_0}(\mathbf{r}))\}, \Pi_t)]. \tag{21}$$

The inequality in (21) follows from the fact that $\mathbb{H}(\cdot, \cdot)$ is a true distance measure and hence follows the triangle inequality. Furthermore, it is like a "bias-variance"

decomposition in the sense that the first term on the right-hand side of (21) can be coerced to zero only by increasing r to ∞, and the second term only by increasing the sample size m to ∞. The broad question is identifying a qualitative relationship between the rates at which r and m should be sent to ∞, while recognizing that $r \times m \times \sum_{j=1}^{r} N(X_0^j, m) = t$.

Using the law of total probability, we can write

$$\mathrm{E}[\mathbb{H}(\pi^*, \{a(\mathbf{X_0}(\mathbf{r}))\}] = \sum_{s \in 2^{\pi^*}} \Pr\{a(\mathbf{X_0}(\mathbf{r})) = s\} \mathbb{H}(\pi^*, s), \tag{22}$$

where 2^{π^*} is the collection of all possible subsets of the finite set π^*. Now every term $p(s) = \Pr\{a(\mathbf{X_0}(\mathbf{r})) = s\} \to 0$ as $r \to \infty$ as long as $s \neq \pi^*$ ($p(\pi^*) \to 1$ as $r \to \infty$). Therefore, the rate at which $\mathrm{E}[\mathbb{H}(\pi^*, \{a(\mathbf{X_0}(\mathbf{r}))\}]$ tends to zero is solely governed by the slowest sequence $\{p(s)\}_r, s \in 2^{\pi^*}$. We thus conclude that $\mathrm{E}[\mathbb{H}(\pi^*, \{a(\mathbf{X_0}(\mathbf{r}))\}]$ is $O\left((1 - p_*)^r\right)$ where $p_* = \min_{j=1,2,\ldots,k} \Pr\{a(X_0) = \{j\}\}$. In other words, the rate at which the first term on the right-hand side of (21) tends to zero is an exponential in r. Furthermore, such convergence can be accelerated by ensuring that p_* is as large as possible, i.e., our sampling distribution F_{x_0} should be such that the likelihood of the algorithm evolving to any of the k zeros of g, when executed on g directly, should be equalized. This is in a sense the inverse of stratified sampling since we seek to sample not in proportion to the "influence region" of a zero, but in such a way that all zeros are "reached" with equal probability. A somewhat similar analysis can be performed on the second term on the right-hand side of (21). Using results from Section 5.2, and making mild assumptions on the random variable $N(x, m)$, it can be shown that $\mathrm{E}[\mathbb{H}(\{a(\mathbf{X_0}(\mathbf{r}))\}, \Pi_t)] = O_p(1/\sqrt{m})$.

These two results on the respective rates at which the terms $\mathrm{E}[\mathbb{H}(\pi^*, \{a(\mathbf{X_0}(\mathbf{r}))\}]$ and $\mathrm{E}[\mathbb{H}(\{a(\mathbf{X_0}(\mathbf{r}))\}, \Pi_t)]$ tend to zero give us a clear qualitative insight on the trade-off between the number of restarts r and sample size m. It seems clear that, in general, the number of restarts should be "very small" in relation to the sample size in the sense of the above arguments. Specifically, it seems to be true that r and m should be chosen so that $r = O(\ln m)$. While elegant, this result is still qualitative, and numerous questions remain unanswered from the standpoint of implementation.

- What fixed sample-size m and fixed restarts r guarantees that the Haudorff distance between the obtained solution set and the true solution set is greater than $\epsilon > 0$ with probability not greater than $\alpha > 0$?
- Can a fixed (m, r) result obtained through answering the first question be extended into a more efficient sequential sampling rule?
- Can the sampling distribution used to generate the initial guess be progressively tilted to ensure that in the limit, each root in the true solution set is reached with equal probability?

REFERENCES

Abbasbandy, S. 2003. Improving Newton–Raphson method for nonlinear equations by modified Adomian decomposition method. *Applied Mathematics and Computation 145*, 887–893.

Andradóttir, S. 1998. A review of simulation optimization techniques. In *Proceedings of the 1998 Winter Simulation Conference*, D. Medeiros, E. Watson, J. S. Carson, and M. S. Manivannan, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 151–158.

ANDRADÓTTIR, S. 2006. An overview of simulation optimization via random search. In *Simulation*, S. G. Henderson and B. L. Nelson, Eds. Handbooks in Operations Research and Management Science. Elsevier, 617–631.

APRIL, J., GLOVER, J., KELLY, J., AND LAGUNA, M. 2001. Simulation/optimization using "real-world" applications. In *Proceedings of the 2001 Winter Simulation Conference*, B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 134–138.

ATLASON, J., EPELMAN, M. A., AND HENDERSON, S. G. 2004. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research 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. *Management Science*. To appear.

AZADIVAR, F. 1999. Simulation optimization methodologies. In *Proceedings of the 1999 Winter Simulation Conference*, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 93–100.

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

BAYRAKSAN, G. AND MORTON, D. P. 2009. A sequential sampling procedure for stochastic programming. *Operations Research*. To Appear.

BHATNAGAR, S. 2005. Adaptive three-timescale stochastic approximation. *ACM Transactions on Modeling and Computer Simulation 15,* 1, 74–107.

BLUM, J. 1954a. Approximation methods which converge with probability one. *Annals of Mathematical Statistics 25,* 2, 382–386.

BLUM, J. 1954b. Multidimensional stochastic approximation. *Annals of Mathematical Statistics 25,* 4, 737–744.

BRATLEY, P., FOX, B. L., AND SCHRAGE, L. E. 1987. *A Guide to Simulation*. Springer-Verlag, New York.

CHEN, H. 1994. Stochastic root finding in system design. Ph.D. thesis, School of Industrial Engineering, Purdue University, West Lafayette, IN.

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

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

CHUNG, K. L. 1954. On a stochastic approximation method. *Annals of Mathematical Statistics 25*, 463–483.

DARVISHI, M. T. AND BARATI, A. 2007. A third-order Newton type method to solve systems of nonlinear equations. *Applied Mathematics and Computation 187*, 630–635.

DOOB, J. L. 1953. *Stochastic Processes*. John Wiley & Sons, New York, NY.

DUNKEL, J. AND WEBER, S. 2009. Stochastic root finding and efficient estimation of convex risk measures. *Operations Research*. In Press.

DVORETZKY, A. 1956. On stochastic approximation. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley, CA., 39–56.

EHRLICHMAN, S. M. T. AND HENDERSON, S. G. 2007. Finite-sample performance guarantees for one-dimensional stochastic root finding. In *Proceedings of the 2007 Winter Simulation Conference*, S. G. Henderson, B. Biller, M. H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 313–321.

FABIAN, V. 1968. On asymptotic normality in stochastic approximation. *Annals of Mathematical Statistics 39*, 1327–1332.

FREES, E. W. AND RUPPERT, D. 1990. Estimation following a Robbins-Monro designed experiment. *Journal of American Statistical Association 85*, 1123–1129.

FRONTINI, M. AND SORMANI, E. 2003. Some variant of Newton's method with third-order convergence. *Applied Mathematics and Computation 140*, 419–426.

FU, M. C. 1994. Optimization via simulation: A review. *Annals of Operations Research 53*, 199–247.

Fu, M. C. 2006. Gradient estimation. In *Simulation*, S. G. Henderson and B. L. Nelson, Eds. Handbooks in Operations Research and Management Science. Elsevier, Amsterdam, Chapter 19, 575–616.

Fu, M. C., Glover, F., and April, J. 2005. Simulation optimization: a review, new developments, and applications. In *Proceedings of the 2005 Winter Simulation Conference*, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey.

Gautschi, W. 1997. *Numerical Analysis: An Introduction*. Birkhäuser, Boston, MA.

Giesecke, K., Schmidt, T., and Weber, S. 2008. Measuring the risk of large losses. *Journal of Investment Management 6,* 4, 1–15.

Glasserman, P. 1991. *Gradient Estimation Via Perturbation Analysis*. Kluwer, Netherlands.

Glynn, P. W. and Whitt, W. 1992. The asymptotic validity of sequential stopping rules for stochastic simulations. *The Annals of Applied Probability 2,* 1, 180–197.

Healy, K. and Schruben, L. W. 1991. Retrospective simulation response optimization. In *Proceedings of the 1991 Winter Simulation Conference*, B. L. Nelson, D. W. Kelton, and G. M. Clark, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 954–957.

Henderson, S. G. and Nelson, B. L., Eds. 2006. Handbooks in Operations Research and Management Science: Simulation, vol. 13. Elsevier.

Herer, Y. T., , Tzur, M., and Yucesan, E. 2006. The multilocation transshipment problem. *IIE Transactions 38*, 185–200.

Homeier, H. H. H. 2003. A modified Newton method for rootfinding with cubic convergence. *Journal of Computational and Applied Mathematics 157*, 227–230.

Homem-de-Mello, T. 2003. Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation 13*, 108–133.

Householder, A. S. 1970. *The numerical treatment of a single nonlinear equation*. McGraw-Hill, New York, NY.

Hsieh, M. and Glynn, P. W. 2002. Confidence regions for stochastic approximation algorithms. In *Proceedings of the 2002 Winter Simulation Conference*, E. Yucesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 370–375.

Joseph, V. R., Tian, Y., and Wu, C. F. J. 2007. Adaptive designs for stochastic root-finding. *Statistica Sinica 17*, 1549–1565.

Juditsky, A., Lan, G., Nemirovski, A., and Shapiro, A. 2009. Stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* 4.

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.

Kesten, H. 1958. Accelerated stochastic approximation. *Annals of Mathematical Statistics 21*, 41–59.

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

Kushner, H. J. and Yin, G. G. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY.

Lai, T. L. 2003. Stochastic approximation. *The Annals of Statistics 31,* 2, 391–406.

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

Lawphongpanich, S., Hearn, D. W., and Smith, M. J., Eds. 2006. Applied Optimization: Mathematical and Computational Models for Congestion Charging, vol. 101. Springer.

Ljung, L. 1977a. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control AC-22*, 551–575.

Ljung, L. 1977b. On positive real transfer functions and the convergence of some recursive schemes. *IEEE Transactions on Automatic Control AC-22*, 539–551.

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. 2007a. Iterative methods with fourth-order convergence for nonlinear equations. *Applied Mathematics and Computation 189,* 1, 221–227.

Noor, M. A. 2007. Some iterative methods free from second derivatives for nonlinear equations. *Applied Mathematics and Computation 192,* 1, 101–106.

Noor, M. A., Khan, W. A., and Hussain, A. 2007. A new modified Halley method without second derivatives for nonlinear equation. *Applied Mathematics and Computation 189,* 2, 1268–1273.

Noor, M. A. and Noor, K. I. 2006. Improved iterative methods for solving nonlinear equations. *Applied Mathematics and Computation 183,* 2. 774–779.

Noor, M. A. and Noor, K. I. 2007b. Improved iterative methods for solving nonlinear equations. *Applied Mathematics and Computation 184,* 2. 270–275.

Ólafsson, S. and Kim, J. 2002. Simulation optimization. In *Proceedings of the 2002 Winter Simulation Conference*, E. Yucesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 79–84.

Ortega, J. M. and Rheinboldt, W. C. 1970. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, NY.

Pasupathy, R. 2009. On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. To Appear in *Operations Research*.

Pasupathy, R. and Schmeiser, B. W. 2009. Retrospective-approximation algorithms for multi-dimensional stochastic root-finding problems. *ACM TOMACS 19,* 2, 5:1–5:36.

Pflug, G. C. 1996. *Optimization of Stochastic Models: The Interface Between Simulation and Optimization*. Kluwer, Boston, MA.

Plambeck, E. L., Fu, B. R., Robinson, S. M., and Suri, R. 1996. Sample-path optimization of convex stochastic performance functions. *Mathematical Programming 75*, 137–176.

Polyak, B. T. and Juditsky, A. B. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization 30,* 4, 838–855.

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

Robbins, H. and Monro, S. 1951. A stochastic approximation method. *Annals of Mathematical Statistics 22*, 400–407.

Ross, S. 1998. *A First Course in Probability*. Prentice Hall, New York, NY.

Royden, H. 1988. *Real Analysis*. Prentice Hall, New York, NY.

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

Ruppert, D. 1985. A Newton-Raphson version of the multivariate Robbins-Monro procedure. *Annals of Statistics 13*, 236–245.

Ruppert, D. 1991. Stochastic approximation. Handbook in Sequential Analysis. Dekker, New York, NY, 503–529.

Sacks, J. 1958. Asymptotic distribution of stochastic approximation procedure. *Annals of Mathematical Statistics 29*, 373–405.

Shapiro, A. 1991. Asymptotic analysis of stochastic programs. *Annals of Operations Research 30*, 169–186.

Shapiro, A. 2004. Monte Carlo sampling methods. In *Stochastic Programming*, A. Ruszczynski and Shapiro, Eds. Handbooks in Operations Research and Management Science. Elsevier, 353–426.

Sheffi, Y. 1985. *Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods*. Prentice Hall, New York, NY.

Spall, J. C. 2000. Adaptive stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control 45*, 1839–1853.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. Wiley, Hoboken, NJ.

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

Swisher, J., Hyden, P. D., Jacobson, S. H., and Schruben, L. W. 2000. A survey of simulation optimization techniques and procedures. In *Proceedings of the 2000 Winter Simulation Conference*, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, Eds. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 119–128.

Todd, M. J. 1976. *The computation of fixed points and applications*. Springer-Verlag, Heidelberg, NY.

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. *Annals of Statistics 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. *Computational and Applied Optimization 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. *Applied Mathematics Letters 13*, 87–93.

Wei, C. Z. 1987. Multivariate adaptive stochastic approximation. *Annals of Statistics 15*, 1115–1130.

Woodroofe, M. 1976. A renewal theoem for curved boundaries and first passage times. *The Annals of Probability 4,* 1, 67–80.

Wu, C. F. J. 1985. Efficient sequential designs with binary data. *Annals of Statistics 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 Monograph Series 8. Hayward, CA, 298–313.

Ying, Z. and Wu, C. F. J. 1997. An asymptotic theory of sequential designs based on maximum likelihood recursions. *Statistica Sinica 7*, 75–918.

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