

AN INTROSPECTIVE ON THE RETROSPECTIVE-APPROXIMATION PARADIGM

Raghu Pasupathy

Industrial and Systems Engineering, Virginia Tech
Blacksburg, VA 24061, USA

ABSTRACT

Retrospective Approximation (RA) is a solution paradigm introduced in the early 1990s by Chen and Schmeiser for solving one-dimensional stochastic root finding problems (SRFPs). The RA paradigm can be thought of as a refined and implementable version of sample average approximation, where a sequence of approximate problems are strategically generated and solved to identify iterates that progressively approach the desired solution. While originally aimed at one-dimensional SRFPs, the paradigm's broader utility, particularly within general simulation optimization algorithms, is becoming increasingly evident. We discuss the RA paradigm, demonstrate its usefulness, present the key results and papers on the topic over the last fifteen years, and speculate fruitful future directions.

1 INTRODUCTION

The earliest reference to *stochastic root finding* — the question of identifying the solution to a nonlinear system of equations with only a consistent estimator of the function involved — was in 1951 when Robbins and Monro wrote an article (Robbins and Monro 1951) outlining an iterative procedure called stochastic approximation (SA). This was soon followed by Kiefer and Wolfowitz's work (Kiefer and Wolfowitz 1952) presenting an analogous SA iteration for the context of *simulation optimization*, i.e., optimization with only a consistent estimator of the objective function. (The name “simulation optimization” was coined recently, and was not used by Kiefer and Wolfowitz in their original paper.) The six decades following the publication of these two influential articles have seen enormous development in the general area of SA, devoted primarily to providing insight on the theoretical properties of the SA iteration (Blum 1954, Dvoretzky 1956, Fabian 1968), and to devising variations aimed at accelerating SA's convergence primarily through insight on the choice of the gain sequence (Kesten 1958, Andradóttir 1991, Andradóttir 1996, Nemirovski and Shapiro 2004, Polyak and Juditsky 1992, Broadie, Cicek, and Zeevi 2010, Broadie, Cicek, and Zeevi 2009, Spall 1998, Spall 2000, Bhatnagar and Borkar 1997, Bhatnagar and Borkar 1998). For some overviews and entry points into this literature, see Lai (2003), Wasan (1969), Kushner and Yin (2003), Spall (2003) .

Interestingly, it was not until 1991 that a fundamentally different and competing solution paradigm was proposed for simulation optimization and stochastic root finding. Perhaps fueled by advances in deterministic nonlinear programming alongside simulation methodology, or motivated by the need for a more reliable and parameterless alternative to SA, Healy and Schruben (1991) proposed solving the simulation optimization problem by a technique called “retrospective optimization.” The technique is simple in concept: generate a sample-path realization (with a “large-enough” sample size) of the optimization problem at hand, and use deterministic nonlinear programming techniques to solve the realized sample-path problem. In other words, Healy and Schruben suggested using a (sampled) deterministic optimization problem as a surrogate to the problem at hand, the solution to which provides an estimator to the true solution. The sample-path approach proposed by Healy and Schruben had three crucial advantages: (i) it provided an opportunity to exploit recent strides in deterministic nonlinear programming; (ii) the approach was simple in that it could (in principle) be implemented by embedding and calling available deterministic solvers inside a simulation

package; and (iii) it lent structural efficiencies because it naturally allowed the use of common random numbers. Healy and Schruben gave several examples to demonstrate the power of the approach. Around the same time, numerous other authors introduced identical or slightly varying techniques (Shapiro 2004), seemingly in a quest for a Monte Carlo paradigm for solving the simulation optimization problem.

While all this interest in a sample-path paradigm for simulation optimization resulted in a much deeper understanding of the asymptotic properties of the resulting solution estimators, it became evident over a period of time that the “one shot sampling” implied by Healy and Schruben’s idea may frequently not be viable. Specifically, suppose a user stipulates that a certain simulation optimization problem be solved to identify a solution estimator with a stipulated quality, as measured by (say) its optimality gap. For particular problem contexts, theoretical results (Shapiro 2004) suggested that the minimum sample size (for generating the sample-path problem) needed for guaranteeing such a stipulated solution quality was often so large as to be impractical. Implementation was thus cumbersome, causing it to fall out of favor with many researchers, particularly in comparison with SA.

Then in 1994, the research landscape changed somewhat when Chen and Schmeiser (1994, 2001), in the context of one-dimensional stochastic root-finding problems, presented retrospective approximation (RA) as a certain refinement of Healy and Schruben’s proposal. Chen and Schmeiser’s key insight was that instead generating and solving a single sample-path problem, why not solve a sequence of sample-path problems generated with gradually increasing sample sizes? Each of the generated sample-path problems could be solved with a deterministic solver to a specified accuracy, and its solution could form the initial guess (“warm start”) to the subsequent problem. This strategy was attractive because it retained all the advantages (i), (ii), and (iii) above of the originally proposed sample-path paradigm while circumventing the sample sizing impasse faced by Healy and Schruben’s proposal. Efficiency was naturally built into the paradigm, at least in principle: solving the early sample-path problems was computationally inexpensive because small sample sizes were used, and solving the later sample-path problems was efficient because the warm starts from the previously solved sample-path problems ensured that not many points had to be visited before a solution was found.

This paper, intended as a tribute to RA, seeks to satisfy three objectives: (i) providing a deeper understanding of RA; (ii) providing an overview of the key papers that have been written on RA in the recent past; and (iii) reflect on current and future work surrounding RA. Towards (i), we present RA in algorithm form, discuss methodological and implementation issues surrounding the components comprising RA, and provide a listing of key theoretical results. Toward (ii) and (iii), we present a bibliography of some of the important and recent papers that have been written on the topic. We also reflect on future research directions within the RA context, and RA’s potential impact on problems lying outside simulation optimization and stochastic root finding.

1.1 Key Notation

The following is a list of key notation and definitions adopted in the paper: (i) π^* denotes the set of *true solutions* to the stochastic root finding or simulation optimization problem; (ii) X_k^* denotes a *true solution* to the k th sample-path problem; (iii) X_k denotes the k th *retrospective solution*, i.e., the estimated solution to the k th sample-path problem; (iv) $X_n \xrightarrow{P} X$ means that the sequence of random variables $\{X_n\}$ converges to the random variable X in probability; (v) $X_n \rightarrow X$ wp1 means that the sequence of random variables $\{X_n\}$ converges to the random variable X with probability one; (vi) $X_n \xrightarrow{d} X$ means that the sequence of random variables $\{X_n\}$ converges to the random variable X in distribution; (vii) $\text{dist}(x, \Omega) = \inf\{\|x - y\| : y \in \Omega\}$ denotes distance between a point $x \in \mathbb{R}$ and a set Ω ; (viii) $B(x, r)$ denotes a ball of radius r centered on x .

1.2 Organization

In Section 2, we formally present the stochastic root finding and simulation optimization problem statements. This is followed by Section 3 where we discuss the RA paradigm, present the RA algorithm, answer frequently

asked questions about RA, and list the most basic asymptotic properties. In Section 4, we provide an overview of recent important works within the context of RA. Section 5 includes some concluding remarks.

2 CONTEXTUAL PROBLEM STATEMENTS

While conceivably much more general, the RA paradigm has been proposed and used only within the contexts of the stochastic root finding problem (SRFP) and the simulation optimization problem (SOP). These are the simulation-based analogues of the deterministic root finding and deterministic optimization problems, respectively. Loosely speaking, an SRFP or an SOP results when the functions involved in a deterministic root finding or optimization problem are available only through a consistent estimator, e.g., a simulation model. Such a modeling framework is enormously attractive for “real-world” problem contexts since all functions within the formulation can be specified implicitly, allowing the embedding of any level of problem complexity. SRFPs and SOPs have found application in a wide variety of contexts such as vehicular transportation networks, quality control, telecommunication systems, and health care. See Andradóttir (2006), Spall (2003), Fu (2002), Barton and Meckesheimer (2006), Chen and Schmeiser (2001), Ólafsson (2006) for entry points into this literature and overviews on the subject.

Formally, the SRFP is stated as follows.

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

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

Similarly, the (global) version of SOP we will use in this paper is as follows.

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

Find: A global minimizer $x^* \in \mathcal{D}$ of g , i.e., find x^* such that every $x \in \mathcal{D}$ satisfies $g(x) \geq g(x^*)$, assuming that one such x^* exists.

As stated, the SRFP and SOP make no assumptions about the nature of $G_m(x)$ except that $G_m(x) \xrightarrow{d} g(x)$ as $m \rightarrow \infty$. Also, the feasible set \mathcal{D} is assumed to be known in the sense that the functions involved in the specification of \mathcal{D} are observed without error. Various slightly differing flavors of the SOP have appeared in the literature. See for example Nemirovski and Shapiro (2004).

3 RA SOLUTION PARADIGM

For concreteness, we present the RA solution paradigm within the context of SRFPs and SOPs. In understanding the RA solution paradigm, the notion of a sample-path problem is important. Loosely speaking, a sample-path problem is an approximation to the problem under consideration, and results when the (deterministic) functions involved within the SRFP or SOP are replaced by their consistent estimators. So, for SRFPs, since the problem is to identify $x = x^*$ satisfying $g(x) = 0$, the corresponding sample-path problem becomes:

$$S_m : \text{ find a zero } X_m^* \in \mathcal{D} \text{ of } G_m, \text{i.e., find } X_m^* \text{ such that } G_m(X_m^*) = 0.$$

Similarly, the sample-path problem for (global) SOPs is:

$$S_m : \text{ find } X_m^* \in \mathcal{D} \text{ such that } G_m(X_m^*) \leq G(x) \text{ for all } x \in \mathcal{D}.$$

Given the definition above of a sample-path problem, the RA paradigm essentially involves generating a sequence of sample-path problems with an increasing sequence of sample sizes, and solving them to progressively stricter error tolerances. Since the sample sizes used for generating the sample-path problems are chosen to be increasing, the resulting problems are better and better approximations (in a certain precise sense) of the underlying problem of interest. When the paradigm works as intended, solving the sample-path problem during the early iterations is efficient because of the small sample sizes and the large error tolerances

in use. The later iterations are efficient because solutions from the preceding sample-path problems are used as “warm starts” to the subsequent sample-path problems, and thus involve lesser exploration of the search space, at least in principle.

3.1 RA Listing

We now formally list the RA paradigm. Note that the listing given below is generic to SRFPs and SOPs — paradigms specific to SRFPs and SOPs are obtained by using the appropriate definition of the sample-path problem.

RA Components:

- (i) A procedure for solving the sample-path problem (S_m) to specified tolerance vector ε_k .
- (ii) A rule to compute the sample size sequence $\{m_k\}$.
- (iii) A rule to compute the error-tolerance sequence $\{\varepsilon_k\}$.
- (iv) A termination criterion, if any.
- (v) A rule to compute the weights $\{w_{kj} : j = 1, 2, \dots, k\}$ during each iteration.

RA Logic:

0. Initialize the retrospective iteration number $k = 1$. Set m_1, ε_1 .
1. Use RA component (i) to solve the sample-path problem (S_{m_k}) to within error-tolerance ε_k . Obtain a retrospective solution X_k .
3. Use component (v) to calculate the weighted solution \bar{X}_k as some weighted sum of retrospective solutions $\{X_i\}_{i=1}^k$:

$$\bar{X}_k = \sum_{j=1}^k w_{kj} X_j.$$

4. Use component (iv) to decide whether to terminate. Return \bar{X}_k as solution upon termination.
5. Use RA Components (ii) and (iii) to get $m_{k+1}, \varepsilon_{k+1}$. Set $k \leftarrow k + 1$ and go to 1.

The RA framework becomes an implementable algorithm when specific choices are made for the five algorithm components. We say more about choosing these algorithm parameters in the ensuing section and in Section 4.

3.2 RA Discussion

In what follows, we discuss and clarify various aspects of the RA framework. These range from rigorously defining specific notions (e.g., “error-tolerance”) used within the paradigm, to a broader discussion on why and under what circumstances we expect the algorithms resulting from the paradigm to function efficiently. Our discussion is in the form of answers to commonly raised questions/issues about particular aspects of RA.

- (a) *What does it mean to solve a sample-path problem to within error-tolerance ε_k (Step 1)?* To clarify this, we first define the function $q(\cdot)$ as a real-valued function such that an arbitrary sequence of “solutions” $\{y_k\}, y_k \in \mathcal{D}$ satisfies $q(y_k) \rightarrow 0$ if and only if $\text{dist}(y_k, \pi^*) \rightarrow 0$. The function $q(\cdot)$ is chosen by the user, and has the interpretation of the quality of any given solution $x \in \mathcal{D}$, with smaller values implying a better quality. Also, suppose that the function $q_k(\cdot)$ is a consistent estimator of $q(\cdot)$. Solving the k th sample-path function to within tolerance ε_k then simply means obtaining a solution X_k that satisfies $q_k(X_k) \leq \varepsilon_k$. As an example, when solving a stochastic root-finding problem, a

reasonable choice of $q(x)$ is $q(x) := g(x)$, in which case $q_k(x)$ can be chosen as $q_k(x) := G_{m_k}(x)$. In local SO problems, a choice such as $q(x) := \nabla g(x)$, $q_k(x) := \hat{\nabla}G_m(x)$ is reasonable.

- (b) *Generating a sample-path problem.* From the structure of the sample-path problem (S_m), it is clear that generating a sample-path problem essentially amounts to generating the sample-path function $G_m(\cdot)$. There are two predominant ways by which such generation happens. In a wide variety of “real-world” simulation contexts, the generation of $G_m(\cdot)$ is *implicit* by which we mean that obtaining the value of $G_m(x)$ involves executing a simulation at the design point x and expending m amount of computational effort. Executing the simulation at x thus provides only local function information, e.g., the function value $G_m(x)$ and possibly the gradient $\nabla G_m(x)$ of the function $G_m(\cdot)$ at x . An example of such implicit construction is a queueing network simulation of a manufacturing system, where $G_m(x)$ represents the iid sample average of throughput over a fixed time horizon, and for a particular routing policy x . By contrast, there are situations where construction of the function $G_m(\cdot)$ is *explicit*. A typical example is when $G_m(x) = m^{-1} \sum_{i=1}^m G(x, \xi_i)$ where $\xi_i, i = 1, 2, \dots, m$ are iid copies of a random variable ξ , and where the structural form of the function $G(\cdot, \cdot)$ is known. In such a setting, we see that upon generating the random variables $\xi_1, \xi_2, \dots, \xi_m$, the entire function $G_m(x)$ is explicitly constructed by “plugging” the obtained random variables $\xi_1, \xi_2, \dots, \xi_m$ into the expression for $G(x, \cdot)$ and then averaging.
- (c) *Why is the RA structure efficient?* The efficiency of the RA paradigm relies crucially on two elements. First, the RA structure is explicitly constructed so that solutions from the previous retrospective iterations can be used as initial solutions when solving the subsequent sample-path problems. If the paradigm works as intended, during the later iterations when the sample size m is large and the sample-path function $G_m(\cdot)$ closely approximates the true function $g(\cdot)$, the warm starts will prove to be extremely beneficial. The typical trajectory of RA algorithms will thus usually involve lots of exploration when solving the sample-path problems during the early iterations, and progressively less and less exploration during the later iterations when the sample size becomes larger. Second, the fact that the sample size is fixed across design points x allows for the use of common random random numbers within the RA framework, in the hope that any available structure within the sample-path functions are not destroyed. Such structure is typically very useful when using a generic numerical procedure to solve the generated sample-path problem.
- (d) *What if the sample-path problem does not have a solution?* This is usually a nuisance that is induced by the structure of the RA framework. This might be a problem particularly during the early iterations when the sample size is small and the sample-path problem is poorly behaved. Assuming there is no way to tell if a generated sample-path problem has a solution, some heuristic such as having an upper bound on the total computational effort expended during an iteration is a possible recourse. The only recourse when the sample-path problem is deemed unsolvable is to move to the next iteration. Under mild conditions on the sample-path function $G_m(\cdot)$ and its limit $g(\cdot)$, the sample-path problems will always have a solution for large enough sample size wp1.
- (e) *How are the sequence of sample sizes $\{m_k\}$ and the sequence of error-tolerances $\{\varepsilon_k\}$ chosen?* The sequence of error tolerances $\{\varepsilon_k\}$ and sample sizes $\{m_k\}$ explicitly control the total work done and the quality of the obtained solution within the RA framework. Specifically, small ε_k values demand more precision in solving the sample-path problems and correspondingly greater computational effort during each iteration. Larger ε_k values, by contrast, reduce the total amount of computational work during each iteration but correspondingly produce less precise sample-path solutions. Likewise, larger sample sizes, while associated with increased total computational effort, ensure better solutions owing to the fact that the corresponding sample-path functions approximate the underlying limit function better. These trade-offs suggest the existence of a relationship between the sequence of error tolerances $\{\varepsilon_k\}$ and the sequence of sample sizes $\{m_k\}$ that is associated with optimal algorithm evolution, however defined. It so happens that such an optimal relationship is indeed characterizable, and leads to specific choices during implementation, when the generalized

mean squared error is used as a measure of algorithm efficiency. We give a few more specifics on such choice in Section 4 — a complete treatment is provided in (Pasupathy 2010).

- (f) *How are the weights $\{w_{kj}\}$ chosen?* There is as yet no general theory on the choice of the weights $\{w_{kj}\}$. It can be shown that if the retrospective solutions X_k are unbiased, then the choice $w_{ki} = m_i^{-1/2}$ is optimal in the sense that it minimizes the mean squared error of the resulting solution \bar{X}_k . In practice, however, the retrospective solutions $\{X_k\}$ are often biased. In the SRFP context, it seems reasonable that the weights should be inversely proportional to the estimated absolute magnitude of the sample-path function at the retrospective solutions, i.e., $w_{kj} \propto \|G_m(X_j)\|$. Analogously, for the local SOP context, choosing the weights to be proportional to the estimated gradient (or the actual gradient, if available) of the sample-path function at the retrospective solutions seems reasonable, $w_{kj} \propto \|\nabla G_m(X_j)\|$. The development of a rigorous theory that justifies such choices is still lacking.

3.3 Asymptotic Behavior

In this section, we provide a sense of the kind of convergence and rate guarantees that can be provided for the sequence of solutions $\{\bar{X}_k\}$ obtained through the RA framework. For brevity, we state our results only for the context of SRFPs. Corresponding results for the SOP context follow analogously and can be found in Pasupathy (2010), Pasupathy and Kim (2011) and the references therein. An authoritative and comprehensive survey of results applicable to sample-path algorithms (see Section 1) for SOPs can be obtained through Shapiro (2004), Ruszcynski and Shapiro (2003). Extensions of these results to the corresponding RA context is in most cases pretty straightforward and adds little further insight.

We first state a result that provides a guarantee on the convergence (to zero) of the sequence $\{g(\bar{X}_k)\}$, i.e., the sequence of function values at the retrospective solutions returned by the RA paradigm. The proof of the theorem is a straightforward application of the definition of uniform convergence.

Theorem 1 Assume

- A₁. The set of zeros $\pi^* \subset \mathcal{D} \subset \mathbb{R}^q$ of the function g is nonempty;
- A₂. The sequence of sample-path functions $\{G_m(x)\}$ is such that the set of zeros Π_m^* of the function G_m is nonempty as $m \rightarrow \infty$ a.s.;
- A₃. The functional sequence $\{G_m(x)\} \rightarrow g(x)$ uniformly as $m \rightarrow \infty$ wp1.

Suppose the sequence of sample sizes $\{m_k\}$ and the sequence of error-tolerances $\{\varepsilon_k\}$ in the RA paradigm are chosen to satisfy $\{m_k\} \rightarrow \infty$ and $\varepsilon_k \rightarrow 0$ as $k \rightarrow \infty$. Furthermore, assume that the sample-path problems (S_{m_k}) are solved to obtain a retrospective solution X_k satisfying $\|G_{m_k}(X_k)\| \leq \varepsilon_k$. Then, $\Delta_{m_k} = \sup_{x \in \Pi_{m_k}^*} \{\|g(x)\|\} \rightarrow 0$ wp1. (Assume $\Delta_{m_k} = \infty$ if $\Pi_{m_k}^* = \emptyset$.)

We note that the result above talks about the quality of the solutions returned by the RA paradigm only in the function space. In other words, it says nothing about what happens to the distance of the retrospective solution X_k from the set of true solutions π^* as $k \rightarrow \infty$. To say something about such convergence, i.e., convergence in the design space, additional conditions need to be imposed. The following theorem shows convergence in the design space after assuming the continuity of the function $g(\cdot)$ and the compactness of the space \mathcal{D} . A proof follows in a somewhat straightforward manner from standard results in sample-path optimization.

Theorem 2 Assume

- A₁. The set of zeros $\pi^* \subset \mathcal{D} \subset \mathbb{R}^q$ of the function g is nonempty;
- A₂. The sequence of sample-path functions $\{G_m(x)\}$ is such that the set of zeros Π_m^* of the function G_m is nonempty as $m \rightarrow \infty$ wp1;
- A₃. The functional sequence $\{G_m(x)\} \rightarrow g(x)$ uniformly as $m \rightarrow \infty$ wp1;
- A₄. The function g is continuous on \mathcal{D} ; and

A5. the set \mathcal{D} is compact.

Suppose the sequence of sample sizes $\{m_k\}$ and the sequence of error-tolerances $\{\varepsilon_k\}$ in the RA paradigm are chosen to satisfy $\{m_k\} \rightarrow \infty$ and $\varepsilon_k \rightarrow 0$ as $k \rightarrow \infty$. Furthermore, assume that the sample-path problems (S_{m_k}) are solved to obtain a retrospective solution X_k satisfying $\|G_{m_k}(X_k)\| \leq \varepsilon_k$. Then, $\{\text{dist}(X_k, \pi^*)\} \rightarrow 0$ wp1 as $k \rightarrow \infty$.

To characterize the rate of convergence of RA's iterates, we now present a central limit theorem on the sequence of retrospective solutions $\{X_k\}$. This result appears with proof in Pasupathy (2010). A more general version of the same result is stated in Pasupathy and Kim (2011).

Theorem 3 Let the conditions of A.3 through A.5 of Theorem 2 hold. Furthermore, assume:

- A.6 X_m^* is the unique zero of the function $G_m(x)$ as $m \rightarrow \infty$ wp1, and x^* is the unique zero of $g(x)$;
- A.7 The functions $G_m(x)$ and $g(x)$ have non-singular derivatives $\nabla G_m(x), \nabla g(x)$ in some neighborhood around x^* ;
- A.8 The sequence $\{\nabla G_m(x)\}$ converges uniformly (elementwise) to $\nabla g(x)$ in some neighborhood around x^* wp1; and
- A.9 A central limit theorem holds for $G_m(x)$, i.e., $\sqrt{m}(G_m(x) - g(x)) \xrightarrow{d} N(0, \Sigma)$, where $N(0, \Sigma)$ is the Gaussian random variable with mean zero and covariance Σ .

Suppose the sequence of sample sizes $\{m_k\}$ and the sequence of error-tolerances $\{\varepsilon_k\}$ in the RA paradigm are chosen to satisfy $\{m_k\} \rightarrow \infty$ and $\varepsilon_k \rightarrow 0$ as $k \rightarrow \infty$. Furthermore, assume that the sample-path problems (S_{m_k}) are solved to obtain a retrospective solution X_k satisfying $\|G_{m_k}(X_k)\| \leq \varepsilon_k$. Then,

$$\sqrt{m_k}(X_k^* - x^*) \xrightarrow{d} N(0, \nabla g(x^*)^{-1}\Sigma(\nabla g(x^*)^{-1})^T).$$

4 A BIBLIOGRAPHY

In this section, we provide a brief account of the important work in the last fifteen years that relates to the RA paradigm in direct and tangential ways. The objective of this exercise is providing a sense of versatility of the RA idea, while also providing a glimpse of the issues that remain unresolved. This listing is by no means comprehensive — please see references listed in the papers cited to get an exhaustive account.

Chen and Schmeiser (2001), stemming from Huifen Chen's doctoral dissertation (Chen 1994), seems to be the amongst the earliest papers to discuss the RA paradigm in complete detail. The RA paradigm was presented in Chen and Schmeiser (2001) as a solution to the SRFP in one dimension. After presenting the RA paradigm, the paper constructs an RA algorithm called Bounding RA by making specific choices for the components listed in Section 3.1. The key idea in Bounding RA is the special structure of the numerical procedure that is used to solve the sample-path problems (S_{m_k}) — the procedure exploits the assumed structure of the underlying function $g(\cdot)$ to rapidly identify a retrospective solution X_k that satisfies $\text{dist}(X_k, \Pi_{m_k}) \leq \varepsilon_k$ during each iteration.

Jihong Jin (1998) designs RA algorithms, again by making specific choices for the components listed in Section 3.1, for solving SOPs. The important idea in this work is the adaptation and use of the Nelder-Mead algorithm for the numerical procedure within the RA paradigm.

Homem-de-Mello (2003) provides sufficient conditions for the schedule of sample sizes — essentially a lower bound on the rate of increase — to guarantee that the iterates obtained from an RA paradigm is consistent. (The paradigm treated in this paper is called a *variable sample size* method.)

Polak and Royset (2008) use a sequence of sample-path problems, akin to the RA paradigm, to solve a class of local SOPs. The main contribution of this work is the construction and estimation of a model of the computational effort involved in solving each of the sample-path problems. The model is then used within an optimization framework to determine the schedule of sample sizes that should be used within the proposed framework. Royset (2011) is a more recent paper along the same lines.

Pasupathy and Schmeiser (2009) generalize Bounding RA (Chen and Schmeiser 2001) to high dimensions. Like Chen and Schmeiser (2001), the key idea in this paper is the construction of a specific numerical procedure (component (i) in Section 3.1) for solving the sample-path problem (S_{m_k}), by and large through the introduction of an appropriate notion of bounding in higher dimensions.

Pasupathy (2010) deals with the question of how to trade-off the sequences $\{m_k\}$ and $\{\varepsilon_k\}$ within RA paradigms to achieve optimal algorithm evolution, defined in a certain rigorous sense. The central message in the paper is twofold. First, the rate at which sample sizes should be increased within RA paradigms depends explicitly on the quality of the numerical procedure in use for solving the generated sample-path problems, with higher increases permitted for faster numerical procedures. Second, an explicit relationship should hold between the error-tolerance sequence $\{\varepsilon_k\}$ and the sample-size sequence $\{m_k\}$ to ensure maximal algorithmic efficiency. Specific recommendations for the sample size and the error tolerance sequences are provided as a function of the convergence rate of the numerical procedure in use.

Wang and Schmeiser (2008) construct specific RA algorithms for local SOPs (i.e., SOPs where a local minimum is desired) when the domain \mathcal{D} is integer-ordered. The important idea in this work is the construction, and subsequent exploitation through the use of fictitious gradients, of a continuous extension of the sample-path functions $G_{m_k}(\cdot)$. The resulting iterates converge to a local minimum exponentially fast, and exhibit very reliable finite-time performance.

Ghosh and Pasupathy (2011) use a linearly converging interior-point method within an RA paradigm to solve general two-stage stochastic linear programs. In addition to proving consistency, Ghosh and Pasupathy provide a complexity analysis leading to guidance on the relative extent to which the sample-path problems should be solved vis-à-vis the prevailing sample-size.

Bayraksan and Morton (2009) develop methods for use as a termination criterion within paradigms such as RA when solving general SOPs. The methods construct finite-time termination rules, which when applied within a sequential sample-path setting, guarantee stopping with a solution having an optimality gap not exceeding a specified threshold, and with a specified confidence.

5 CONCLUDING REMARKS

RA is an attractive way to implement sample-path methods for simulation optimization and stochastic root finding problems. The RA paradigm is simple to implement, exploits the power of deterministic optimization, and naturally lends efficiency through its use of common random numbers and warm starts. It fully circumvents implementation issues faced by traditional sample-path methods through the use of a sequence of approximate deterministic problems generated with gradually increasing sample sizes. Furthermore, recent work seems to indicate that RA's iterates, like SA and unlike traditional sample-path methods, achieve the canonical asymptotic convergence rate $O(1/\sqrt{\text{simulation effort}})$.

RA has already had substantial impact on research directions within the context of root finding and optimization. Indications are that the coming years will increase such influence, as crucial parameter-choice related questions get resolved and commercial simulation languages become more sophisticated in their use of simulation optimization technology. Furthermore, RA will find use in areas outside stochastic root finding and simulation optimization. This is evident upon observing that RA's main idea is progressively generating (through simulation) better and better deterministic approximations of an underlying problem, and solving each of the generated problems to prescribed accuracy using solutions from previously generated problems as warm starts. This idea of progressive solution is not specific to stochastic root finding or simulation optimization, and seems like a viable strategy that should find use (albeit in various incarnations) across a broad swath of problem types.

ACKNOWLEDGEMENTS

The author was supported in part by Office of Naval Research contract N000141110419 and by IBM Research.

REFERENCES

- Andradóttir, S. 1991, December. "A projected stochastic approximation algorithm". In *Proceedings of the 1991 Winter Simulation Conference*, edited by B. L. Nelson, W. D. Kelton, and G. M. Clark, 854–957. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Andradóttir, S. 1996. "A Scaled Stochastic Approximation Algorithm". *Management Science* 42:475–498.
- Andradóttir, S. 2006. "An overview of simulation optimization via random search". In *Simulation*, edited by S. G. Henderson and B. L. Nelson, Handbooks in Operations Research and Management Science, 617–631. Elsevier.
- Barton, R. R., and M. Meckesheimer. 2006. "Metamodel-based simulation optimization". In *Simulation*, edited by S. G. Henderson and B. L. Nelson, Handbooks in Operations Research and Management Science, 535–574. Elsevier.
- Bayraksan, G., and D. P. Morton. 2009. "A sequential sampling procedure for stochastic programming". *Operations Research*. To Appear.
- Bhatnagar, S., and V. S. Borkar. 1997. "Multiscale stochastic approximation for parametric optimization of hidden Markov models". *Probability in the Engineering and Informational Sciences* 11:509–522.
- Bhatnagar, S., and V. S. Borkar. 1998. "A two time scale stochastic approximation scheme for simulation based parametric optimization". *Probability in the Engineering and Informational Sciences* 12:519–531.
- Blum, J. 1954. "Approximation Methods Which Converge with Probability One". *Annals of Mathematical Statistics* 25 (2): 382–386.
- Broadie, M., D. M. Cicek, and A. Zeevi. 2009, December. "An adaptive multidimensional version of the Kiefer-Wolfowitz Stochastic Approximation Algorithm". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 601–612. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Broadie, M., D. M. Cicek, and A. Zeevi. 2010. "General Bounds and Finite-Time Improvement for the Kiefer-Wolfowitz Stochastic Approximation Algorithm". *Operations Research*. To appear.
- Chen, H. 1994. *Stochastic Root Finding in System Design*. Ph. D. thesis, School of Industrial Engineering, Purdue University, West Lafayette, IN.
- Chen, H., and B. W. Schmeiser. 2001. "Stochastic root finding via retrospective approximation". *IIE Transactions* 33:259–275.
- Dvoretzky, A. 1956. "On Stochastic Approximation". In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, 39–56: University of California Press, Berkeley, CA.
- Fabian, V. 1968. "On asymptotic normality in stochastic approximation". *Annals of Mathematical Statistics* 39:1327–1332.
- Fu, M. C. 2002. "Optimization for simulation: Theory vs. practice". *INFORMS Journal on Computing* 14:192–215.
- Ghosh, S., and R. Pasupathy. 2011, December. "On Interior-Point Based Retrospective Approximation Methods for Solving Two-Stage Stochastic Linear Programs". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelsbach, K. P. White, and M. Fu. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Healy, K., and L. W. Schruben. 1991, December. "Retrospective simulation response optimization". In *Proceedings of the 1991 Winter Simulation Conference*, edited by B. L. Nelson, W. D. Kelton, and G. M. Clark, 954–957. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Homem-de-Mello, T. 2003. "Variable-sample methods for stochastic optimization". *ACM Transactions on Modeling and Computer Simulation* 13:108–133.
- Jin, J. 1998. *Retrospective Optimization of Stochastic Systems*. Ph. D. thesis, School of Industrial Engineering, Purdue University, West Lafayette, IN.
- Kesten, H. 1958. "Accelerated stochastic approximation". *Annals of Mathematical Statistics* 21:41–59.
- Kiefer, J., and J. Wolfowitz. 1952. "Stochastic estimation of the maximum of a regression function". *Annals of Mathematical Statistics* 23:462–466.

- Kushner, H. J., and G. G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. New York, NY.: Springer-Verlag.
- Lai, T. L. 2003. "Stochastic Approximation". *The Annals of Statistics* 31 (2): 391–406.
- Nemirovski, A., and A. Shapiro. 2004. "Convex approximations of chance constrained programs". *Optimization Online*. <http://www.optimization-online.org/>.
- Ólafsson, S. 2006. "Metaheuristics". In *Simulation*, edited by S. G. Henderson and B. L. Nelson, Handbooks in Operations Research and Management Science, 633–654. Elsevier.
- Pasupathy, R. 2010. "On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization". *Operations Research* 58:889–901.
- Pasupathy, R., and S. Kim. 2011. "The stochastic root-finding problem: overview, solutions, and open questions". *ACM TOMACS* 21 (3).
- Pasupathy, R., and B. W. Schmeiser. 2009. "Retrospective-approximation algorithms for multidimensional stochastic root-finding problems". *ACM TOMACS* 19 (2): 5:1–5:36.
- Polak, E., and J. O. Royset. 2008. "Efficient sample sizes in stochastic nonlinear programming". *Journal of Computational and Applied Mathematics* 217:301–310.
- Polyak, B. T., and A. B. Juditsky. 1992. "Acceleration of Stochastic Approximation by Averaging". *SIAM Journal on Control and Optimization* 30 (4): 838–855.
- Robbins, H., and S. Monro. 1951. "A stochastic approximation method". *Annals of Mathematical Statistics* 22:400–407.
- Royset, J. 2011. "On Sample Size Control in Sample Average Approximations for Solving Smooth Stochastic Programs". *Journal of Computational and Applied Mathematics*. Under Review.
- Ruszczynski, A., and A. Shapiro. (Eds.) 2003. *Stochastic Programming. Handbook in Operations Research and Management Science*. New York, NY.: Elsevier.
- Shapiro, A. 2004. "Monte Carlo sampling methods". In *Stochastic Programming*, edited by A. Ruszczynski and Shapiro, Handbooks in Operations Research and Management Science, 353–426. Elsevier.
- Spall, J. C. 1998. "Implementation of the simultaneous perturbation algorithm for stochastic optimization". *IEEE Transactions on Aerospace and Electronic Systems* 34:817–823.
- Spall, J. C. 2000. "Adaptive stochastic approximation by the simultaneous perturbation method". *IEEE Transactions on Automatic Control* 45:1839–1853.
- Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization*. Hoboken, NJ.: John Wiley & Sons, Inc.
- Wang, H., and B. Schmeiser. 2008, December. "Discrete Stochastic Optimization Using Linear Interpolation". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 502–508. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Wasan, M. T. 1969. *Stochastic Approximation*. Cambridge, UK: Cambridge University Press.

AUTHOR BIOGRAPHY

RAGHU PASUPATHY is an associate professor in the Industrial and Systems Engineering Department at Virginia Tech. His research interests lie broadly in Monte Carlo methods with a specific focus on simulation optimization and stochastic root finding. He is a member of INFORMS, IIE, and ASA, and serves as an Associate Editor for ACM TOMACS and INFORMS Journal on Computing. His e-mail address is pasupath@vt.edu and his web page is <https://filebox.vt.edu/users/pasupath/pasupath.htm>.