An Algorithm for Gradient-Free Simulation Optimization Using Sampling Control[1]

RAHUL SIMHA
Department of Computer Science
The George Washington University
Washington, DC 20052
simha@seas.gwu.edu

Abstract

This paper considers a class of continuous-variable optimization problems in which the objective function cannot be computed exactly but must instead be *estimated* by using, typically, a simulation. Since the simulation output is stochastic, iterative optimization algorithms for these problems are often augmented with noise-removal features to ensure convergence to an optimal solution. Two broad noise-removal approaches are considered: *stepsize-control*, which involves a decreasing stepsize or *sampling-control*, which involves an increasing sample size. Of these two, stepsize-control predates sampling-control by several decades and is the most commonly used approach, although, as we show, sampling-control has some advantages over stepsize-control. Our particular focus is on optimization problems for which direct gradient estimates are not available and instead, must be approximated using estimates of the objective function. The classic Kiefer-Wolfowitz algorithm [20], using stepsize-control, is one such algorithm that estimates a divided difference approximation of the gradient. This paper presents a sampling-controlled version of this algorithm that also uses divided difference estimates and has the benefit of being easily parallelizable. A convergence proof and some simulation results are also included.

[1]This work was partially supported by NSF grant ANI-9973106 and Darpa grant N66001-00-1-8949.

1 Introduction

Most nonlinear continuous-variable optimization problems are solved by *iterative* algorithms that start with an initial solution and gradually iterate towards the optimum [26, 28]. The key inputs to each iterative step of these algorithms usually involve evaluating gradients or the objective function. In many applications, however, the gradient or objective function cannot be computed exactly but instead must be *estimated* or measured, subject to random errors. This sub-class of problems is typified by objective functions whose evaluation consists of a simulation driven by a random number generator [6, 14, 23, 24, 27, 30]. When such random estimates are used as inputs to an iterative optimization algorithm, the resulting iteration is *stochastic*. Ever since the introduction of the now-classic Robbins-Munro algorithm [32], such stochastic iterative algorithms have been studied in various application areas under different names: simulation optimization (simulation) [27, 30], stochastic approximation (control, statistics and bio-assay) [4, 20, 21, 32, 43] and stochastic learning (learning theory) [22, 29].

If the random "noise" introduced by estimation is unchecked, a stochastic iterative algorithm that uses gradient or function estimates can end up not converging. The vast majority of stochastic iterative algorithms remove this noise by augmenting the corresponding deterministic algorithm with a sequence of *decreasing stepsizes*, one for each computational step of the algorithm. We refer to this method of augmenting a deterministic iterative algorithm as *stepsize-controlled iteration*. In prior work [12, 36, 38], we introduced an alternative noise removal technique, termed *sampling-controlled iteration*, that removes noise through repeated sampling. This procedure was shown to have advantages over stepsize-control such as requiring less computation [36], handling inherent bias in measurements and, in some cases, better performance [36, 38]. For both types of noise-removal approaches, it can be shown that, for a wide class of algorithms, convergence to the desired optimum is obtained with probability one with appropriate regularity and distribution assumptions [4, 11, 12, 21, 25].

In this paper, we take up the problem of stochastic optimization when gradients cannot be estimated directly. In this case, the gradient in a deterministic iterative algorithm is replaced by an estimate of a divided-difference approximation of the gradient. The stepsize-controlled version of this algorithm is the well-known Kiefer-Wolfowitz algorithm [20]. We provide a sampling-controlled version in this paper that has the advantages of sampling-control mentioned above. In addition, we highlight an important new computational advantage here: the sampling-controlled algorithm is easily parallelizable. With some regularity assumptions, we prove convergence with probability one and, just as Kiefer-Wolfowitz derived an exact relation between the decreasing stepsize and decreasing divided-difference parameter, so do we derive an exact relation between the divided-difference parameter and increasing sample size. The proof uses some elementary upper bound results from large deviation theory [42, 35] and are presented for the single-variable case for ease of exposition; we believe that they are easily generalizable to the multivariate case because the corresponding large deviations results also generalize.

At this point, we note that both stepsize-control and sampling-control may be used simultaneously with appropriate rates of stepsize decrease and sampling increase. Indeed, a few deterministic algorithms already use decreasing stepsizes and thus, a sampling-controlled stochastic version would only add estimation with increasing sample sizes. However, we are particularly interested in the class of deterministic iterative algorithms for which a *fixed* stepsize is sufficient, a class that includes many gradient-based iterations [9, 13, 26, 28, 36, 40].

As mentioned above, prior work in the general area of stochastic iterative algorithms is spread across various areas. For a recent survey of simulation optimization see [2, 6] and for a comprehensive introduction see [30]. Stochastic approximation, introduced in [32], is covered in detail in several books [4, 21, 43]. The large deviations approach to stepsize-control was introduced in [11] and to sampling-control in [12]. (See [35] for an introduction to large deviations.) Derivative estimation, in particular, the likelihood ratio and perturbation analysis methods, is covered in [7, 34]. Examples of deterministic iterative optimization algorithms together with applications can be found in [5, 9, 13, 16, 19, 26, 28, 40]. Stochastic iterative optimization algorithms are described in [3, 8, 10, 12, 15, 22, 23, 24, 29, 36, 38, 39]. An on-line bibliography of resources in the area of simulation optimization is maintained at [1]. The starting point for this paper is the simple, but effective (stepsize-controlled) Kiefer-Wolfowitz algorithm [20], for which we provide a sampling-controlled version.

The paper is structured as follows. In the next section, we review the key ideas in the sampling-controlled approach. In Section 3 we present our sampling-controlled optimization algorithm and prove convergence in Section 4. Next, we present some simulation results in Section 5 before summarizing in Section 6.

2 Sampling-controlled Stochastic Approximation: An Overview

As mentioned earlier, a stochastic iterative algorithm in which a deterministic algorithm is augmented with decreasing stepsizes is referred to as a *stepsize-controlled* algorithm whereas an augmentation with increasing sampling sizes is referred to as a *sampling-controlled* algorithm [12]. We now present a summary of some of the key ideas as background for our new algorithm. We begin with a generic deterministic iterative algorithm:

$$x^{(n+1)} = x^{(n)} + ah(x^{(n)}) \tag{1}$$

where n denotes the iteration number, h is a real-valued function and a is a real-valued stepsize. Here, the sequence of iterates $x^{(n)} \in \mathbb{R}$ is assumed to converge to some desired optimum, $\lim_{n\to\infty} x^{(n)} = x^*$. For example, if h is the gradient of a twice-differentiable unimodal function f, $x^{(n)}$ converges to the unique (unconstrained) optimum x^*. We refer to the above iteration as the *deterministic version* of the algorithm h. In this case, the gradient h is computed analytically.

In many applications, however, neither the function h nor even the objective function f is known; instead, a simulation is performed to *estimate* h or f using random samples. First, let us consider the case in which h can be estimated directly. We note that gradient estimation for simulations, especially Markovian simulations, has been the subject of much interest in the past 15 years [7, 17, 24, 30, 31, 41]. For each $x \in \mathbb{R}$ let $\left\{Y^k(x), k = 1, 2, ...\right\}$ be a family of random variables and define

$$\bar{Y}^L(x) = \frac{1}{L}\sum_{k=1}^{L} Y^k(x), \qquad L = 1, 2, ...$$

to be a point estimate [33] of a function $Y(x)$ based on L independent, identically distributed (iid) random samples, $Y^1(x), ..., Y^L(x)$. In most applications, $Y(x)$ does not directly correspond to h, but instead, typically, $Y(x)$ represents a system quantity that is relatively easy to estimate whereas $h(x)$ is a more complex system property that is expressed as a function of simpler quantities such as $Y(x)$. (For example, in a queueing system, h may be a delay gradient with respect to the arrival rate whereas $Y(x)$ is the steady-state mean delay. Then, in order to estimate h, it becomes necessary to use an estimate, $g\left(\bar{Y}^L(x)\right)$, constructed from the estimate $\bar{Y}^L(x)$ of $Y(x)$, where g is a function on the range of $\bar{Y}^L(x)$.

Now suppose $E\left[g\left(\bar{Y}^L(x)\right)\right] = h(x)$ is an unbiased estimator. Define the *stochastic iteration*

$$X^{(n+1)} = X^{(n)} + a_n g\left(\bar{Y}^{L_n}(X^{(n)})\right) \tag{2}$$

where a_n is a sequence of positive real numbers and L_n is a sequence of positive integers. Observe that since $g\left(\bar{Y}^{L_n}(X^{(n)})\right)$ is a random variable, the sequence $X^{(n)}$ is a sequence of random variables. We now distinguish between two types of stochastic iterations:

- *Stepsize-controlled iteration.* In this case, the fixed stepsize in equation (1) is replaced with decreasing stepsizes that satisfy the standard stepsize-control conditions [32]: $a_n > 0, a_n \to 0$ and $\sum_{i=1}^{\infty} a_i = \infty$. Here, the sample size is typically unity: $L_n = 1$. The intuition is that the divergent sum prevents the algorithm from getting "stuck" at a non-optimal point, while the decreasing stepsizes gradually remove estimation noise.

- *Sampling-controlled iteration.* In this form of iteration, the fixed stepsize is retained ($a_n = a$) from equation (1) but the number of samples per iteration is gradually increased: $L_n > \gamma_n \log_n$, where γ_n is any sequence that diverges to infinity (e.g., $\gamma_n = n^\epsilon$ or $\gamma_n = \log n$). The intuition behind sampling-control is relatively easier to follow: initially, the algorithm uses high-noise estimates assuming the initial value is far from optimal, and later makes cautious (large-sample) estimates assuming it is closer to the optimum. Clearly, for very large L_n, the iteration is almost deterministic (at a sufficiently large distance away from the optimum).

It is possible to show that, under certain regularity conditions [4, 11, 12, 21, 25, 30], these stochastic iterations converge to x^*, i.e., $\lim_{n\to\infty} X^{(n)} = x^*$ *a.s.* (the notation *a.s.* is used to denote *almost surely* or convergence with probability one [33]).

The two approaches may be compared as follows As shown in [12], sampling-control is theoretically better at removing the effects of estimator bias and transient behavior. Note that some experimentation [38] with optimization of queueing systems indicates that bias is often not an insurmountable concern. As a practical matter, however, the stepsize in stepsize-control can get too small to allow for effective iteration. These observations have been pointed out elsewhere [12, 36, 38]. We now identify one additional advantage in using sampling-control: the L_n samples at each iteration can be obtained in parallel and since, in most simulations, the L_n samples are iid samples, the parallelization effort is minimal. As we will see, the simulation results in this paper support this view.

As mentioned earlier, in most optimization problems of interest, the function h is either the gradient (typically, in an unconstrained problem [26]) or some function of the gradient (in a constrained problem [13, 38]) and thus, $g\left(\bar{Y}^L(x^{(n)})\right)$ is, loosely speaking, an estimate of the gradient. While direct gradient estimation is possible in some systems [7], in many cases it is not. In these cases, it is customary to continue to use the same deterministic form but to approximate the gradient using divided-differences. Thus, in a stochastic iteration, the actual value used is both an approximation and an estimate. In the next section, we present a sampling-controlled version of one such iteration.

3 A Sampling-Controlled Kiefer-Wolfowitz Algorithm

When gradients cannot be computed, a divided-difference approximation to the gradient can be used even in deterministic problems. For example, consider the following modification of equation (1):

$$x^{(n+1)} = x^{(n)} + a\left(\frac{f(x^{(n)} + c_n) - f(x^{(n)})}{c_n}\right) \tag{3}$$

where f is the function being optimized and c_n is a real sequence converging to zero. Here, the gradient h in equation (1) is replaced by a divided-difference approximation $\frac{f(x^{(n)}+c_n)-f(x^{(n)})}{c_n}$ with an iteration-dependent parameter c_n. If the original algorithm converged, the condition $c_n \to 0$ is usually sufficient for convergence of the above algorithm. As mentioned earlier, in general, h might be a function of the gradient rather than the gradient itself; for example, in several constrained optimization algorithms h is not even a continuous function of the gradient [13, 36]. Note that divided difference estimates (also called finite-difference estimates) can be computed in several different ways in the multidimensional case – see [10] for further details.

Now, consider a stochastic version of equation (3). First, we define a point estimator for $f(x)$:

$$\bar{Z}^L(x) = \frac{1}{L}\sum_{k=1}^{L} Z^k(x), \qquad L = 1, 2, ...$$

where $Z^k(x), k = 1, 2, ...$ are iid random variables such that $E\left[Z^k(x)\right] = f(x)$. Then, a stochastic iteration can be described as follows:

$$X^{(n+1)} = X^{(n)} + a_n\left(\frac{\bar{Z}^{L_n}(X^{(n)} + c_n) - \bar{Z}^{L_n}(X^{(n)})}{c_n}\right) \tag{4}$$

Thus, simulation estimates are taken at parameter values $X^{(n)}$ and $X^{(n)} + c_n$ and their difference divided by c_n, thereby giving an estimated approximation to the gradient. It is important to note that different random values are used in the two estimates (even though the notation used is the same).

The stochastic iteration in equation (4), like that of equation (2), also has stepsize-control and sampling-control versions:

- *Stepsize-controlled iteration.* The stepsize-controlled version is the well-known Kiefer-Wolfowitz algorithm [20], in which $a_n \to 0$ and $L_n = 1$. In this case, the following conditions are used to prove convergence [20]: $c_n \to 0$, $a_n \to 0$, $\sum a_n = \infty$, $\sum a_n c_n < \infty$ and $\sum a_n^2 c_n^{-2} < \infty$. Intuitively, the difference parameter c_n needs to converge much slower than the stepsize. For example, the above conditions are satisfied when $a_n = n^{-1}$ and $c_n = n^{-1/3}$.

- *Sampling-controlled iteration.* The sampling-controlled version is the focus of this paper. Essentially, we show that $\frac{L_n c_n^2}{\log n} \to \infty$ is the equivalent condition for sampling-control. Analogous to the stepsize case, the intuition here is that the the sampling rate must increase faster than the difference parameter decreases.

In the next section, we use large deviations theory to prove convergence and derive the above relation between the sampling rate L_n and the sequence of difference parameters c_n.

4 Convergence

We will use the following assumptions in our proof:

- *A*1. *Compactness of domain.* We will assume throughout the paper that the iterates $X^{(n)}$ remain in a compact set $D \subset \mathbb{R}$. This assumption is typically necessary to satisfy assumption $A2$ described below and, practically, may have to be accomplished through some type of constraint enforcement mechanism [13, 36], which we will assume to be incorporated into the function h. It is worth mentioning that in the stochastic case, additional care may be needed to ensure that $X^{(n)}$ remains in D [36].

- *A*2. *Stability of convergence under perturbations.* As in [12], we assume that the deterministic version given by equation (1) satisfies the following stability property. Given $\delta > 0$ there exists $N = N(\delta) < \infty$ and $\epsilon = \epsilon(\delta) > 0$ such that if the real-valued iterates $z^{(n)}$ are returned through the perturbed recursion

$$z^{(n+1)} = z^{(n)} + af'(z^{(n)} + \epsilon^n), \qquad n = 0, 1, ..., 2N - 1 \tag{5}$$

and if $\sup_{0 \leq n \leq 2N} |\epsilon^n| < \epsilon$ (using any suitable norm, the sup norm, for example) then, for all $z^{(0)} \in D$,

$$\left| z^{(n)} - x^* \right| \leq \delta, \qquad N \leq n \leq 2N$$

Here, f' denotes the derivative of f. Intuitively, this assumption characterizes the convergence behavior of the deterministic version under infinitesimally small perturbations by requiring the trajectory of the perturbed version, equation (5), to behave nearly as well as the unperturbed version, equation (1), for a certain fixed number of iterations (N to $2N$), for small enough perturbations (smaller than ϵ). As discussed in [12], the assumption allows for the possibility of the function h being discontinuous.

- *A*3. *Twice-differentiability of objective function.* We assume that f is twice-differentiable at all points in D and that f'' is bounded on D.

- *A*4. *Differentiability of large deviations rate function.* For each x, define

$$M(\theta, x) = E\left[e^{\theta Z^k(x)}\right]$$

and assume the large deviations rate function

$$I(z, x) = \sup_{\theta}[\theta z - \log M(\theta, x)]$$

exists in a neighborhood of $z = 0$ and is twice-differentiable at $z = 0$. Note that $I(0, x) = E\left[Z^k(x)\right]$ [35].

A discussion of some of these assumptions and their reasonableness may be found in [12]. Note that the process $X^{(n)}$ may visit some point in D more than once, i.e., $X^{(n_1)} = X^{(n_2)} = x$ for some x and $n_1 < n_2$. In this case, it should be understood that, at iteration n_2, different samples, independent of those taken at n_1 are used in the construction of the estimate $\bar{Z}^{L(n_2)}(X^{(n_2)})$.

We next present the key steps in the proof. Consider $f'(x)$, the derivative of $f(x)$ and let c denote a slight change (increase) in x. Suppose we let ϵ_1 and ϵ_2 be errors in estimation of f at $x+c$ and x respectively. Then, for the error of our estimate of f' to be less than ϵ we get the condition:

$$\frac{f(x+c) + \epsilon_1 - (f(x) + \epsilon_2)}{c} - f'(x) \quad < \quad \epsilon.$$

Using a Taylor approximation, this can be restated as

$$\frac{f(x) + f'(x)c + 1/2f''(x)c^2 + \epsilon_1 - (f(x) + \epsilon_2)}{c} - f'(x) \quad < \quad \epsilon.$$

Upon simplification,

$$\frac{\epsilon_1 + \epsilon_2}{c} + 1/2cf''(x) \quad < \quad \epsilon$$

which holds if both

$$c < \frac{\epsilon}{f''(x)}$$

and

$$\epsilon_1 + \epsilon_2 < \frac{\epsilon c}{2}. \tag{6}$$

For fixed ϵ, c can be chosen small enough to satisfy the former inequality using assumption $A3$.

Now consider the second inequality and let

$$\epsilon_3 = \min(\epsilon_1, \epsilon_2). \tag{7}$$

The key idea in the proof is to show that for large n, the probability of incurring the ϵ errors due to estimation noise is exponentially small. Indeed, the upper bound of Cramer's theorem [35, 42] gives

$$P\left[\left|\bar{Z}^L - f(x)\right| \geq \epsilon_3\right] \leq \exp\left(-I(\epsilon_3, x)L\right).$$

In other words, if L is large enough (in a single estimate), the estimator $\bar{Z}^L$ differs from the mean (by an amount ϵ_3) with exponentially low probability.

We will next follow the outline of theorems 1-3 in [12] and prove that $P\left[\left|X^{(n)} - x^*\right| > \delta \ i.o.\right] = 0$ for all $\delta > 0$ (here, the notation $i.o.$ refers to 'infinitely often' [33]), thereby showing that $X^{(n)} \to x^*$ with probability one. We begin with the parameter N in assumption $A2$ and group the set of all iterations into blocks of size N. Fix $\delta > 0$ and define

$$E_i = \left\{\left|X^{(n)} - x^*\right| > \delta \text{ for some } n,\ iN \leq n \leq iN + N\right\}$$

where N is from assumption $A3$ for our chosen value of δ. Then,

$$\left\{\left|X^{(n)} - x^*\right| > \delta \; i.o.\right\} = \{E_i \; i.o.\} \quad a.s. \tag{8}$$

where the first event in equation (8) is $i.o.$ in n whereas the second is $i.o.$ in i. Next, we use the stability property in assumption $A2$ to find an equivalent "event" in terms of the behavior of the estimator. With the above value of δ, pick ϵ as in assumption $A2$ and, following that, pick ϵ_3 from equation (7).

Now, Theorem 3 in [12] shows that

$$P[E_i] \leq 2N \exp\left(-\beta\left[\inf_{iN-N\leq n\leq iN+N} L(n)\right]\right)$$

where

$$\beta = \inf_{x\in D, u>\epsilon_3} I(u,x)$$

Then,

$$\begin{aligned}
\sum_{i=0}^{\infty} P[E_i] &\leq 2N \sum_{n=0}^{\infty} \exp\left(-\beta\left[L(n)\right]\right) \\
&= 2N \sum_{n=0}^{\infty} n^{-\beta[\gamma_n]} \\
&\leq 2N \sum_{n=0}^{\infty} n^{-B(c_n)[\gamma_n]}
\end{aligned}$$

where, using inequality (6),

$$B(c_n) = \inf_{x\in D, u>\epsilon c_n/2} I(u,x)$$

and

$$L_n = \gamma_n \log n.$$

Now, if γ_n is any sequence of positive real numbers such that $B(c_n)\gamma_n \to \infty$, then the sum above is finite, and therefore, by the Borel-Cantelli lemma [33] $P[E_i \; i.o.] = 0$.

The final argument required is to determine the rate at which $B(c_n)$ goes to zero. We can infer from Chapter 9 of [35] that the large deviations rate function is bounded by a quadratic near zero, i.e., $B(x) \leq \nu x^2$ for some constant $\nu > 0$ in a sufficiently small neighborhood of $x = 0$. Thus, the condition $c_n^2\gamma_n \to \infty$ is enough to guarantee convergence. We now state the result formally.

Theorem: *Consider a deterministic iteration of the type in equation (1) that converges to a desired point x^* satisfying assumptions $A1 - A3$. Then, the divided-difference stochastic version, equation (4), converges to x^* if assumption $A4$ is satisfied and the sequences c_n and L_n satisfy $\frac{L_n c_n^2}{\log n} \to \infty$.*

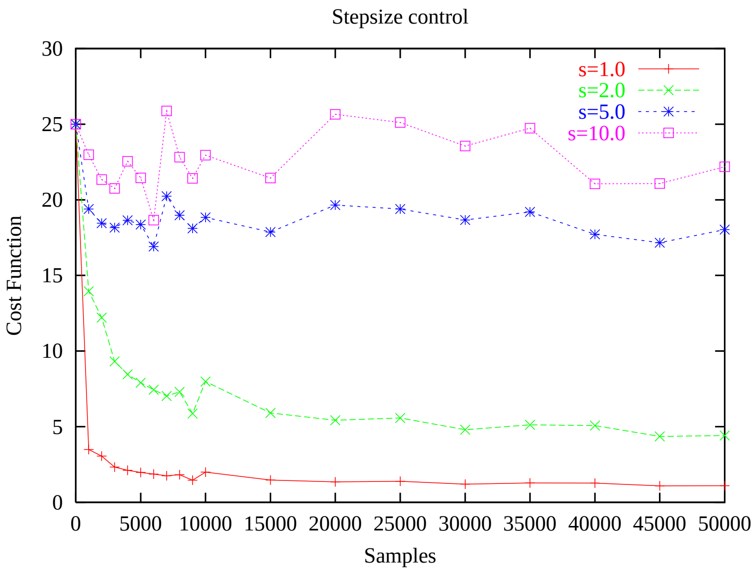

Figure 1: Stepsize-controlled iteration

5 Simulation Results

We now present some simulation results comparing the stepsize and sampling-controlled approaches. In addition, we assess the potential of parallelism in the sampling-controlled approach.

For the first set of results, we consider the simple function $f(x) = x^2$ and add Gaussian noise [33] to create stochastic output. More precisely, let W_σ be a Normal random variable with mean zero and standard deviation σ. Then, define the random variables $Z^k(x) = x^2 + W_\sigma$ so that $E\left[\bar{Z}^L(x)\right] = f(x)$ in equation (4). Thus, a "simulation" estimates $f(x)$ with the estimator $Z^L(x)$. By varying the standard deviation σ, one obtains different levels of noise; for example $\sigma = 0$ results in a deterministic iteration whereas $\sigma = 10$ results in considerable noise.

Figure 1 shows the performance of the stepsize-controlled approach for various parameters. Because we use this type of graph throughout the paper, we now explain the output in some detail. Loosely speaking, we wish to evaluate the "progress" of an algorithm over time. To compare both approaches fairly, the x-axis is marked in numbers of samples rather than iterations; this is because a sampling-controlled algorithm will use far fewer iterations and many more samples between iterations than a stepsize algorithm. Yet the real "cost" in terms of execution time is the time spent in "simulation" (extracting a measurement). Accordingly, we believe the total number of samples taken is a reasonable approximation of the actual run-time. Next, we explain how a y-value is obtained. Consider x-axis value of, say, 5000. We run the algorithm together with the simulation and halt after 5000 samples are taken (totally). At this point, the iteration will have reached some value of the parameter x, say, x_0. Then, $f(x_0)$ is, roughly, the progress made by the algorithm after 5000 samples. An average of 100 independent such evaluations is used as each y-value. Thus, to some degree of confidence, the curve that joins these points is a profile of the

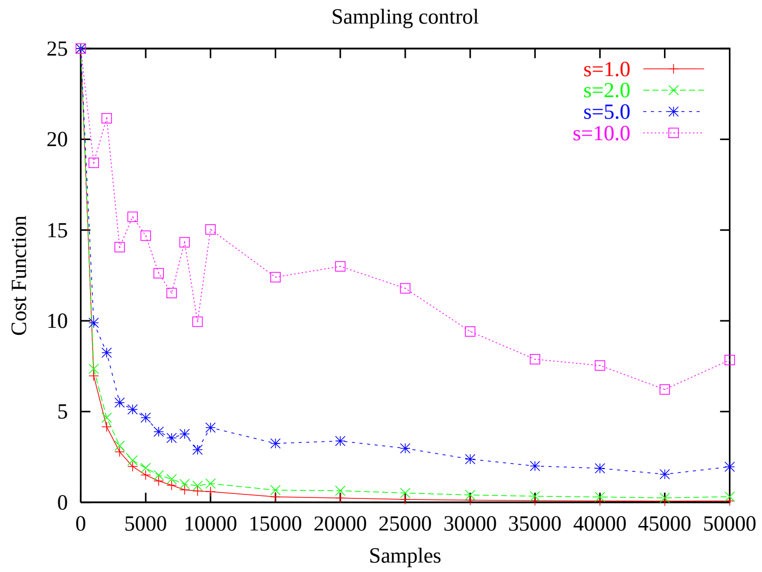

Figure 2: Sampling-controlled iteration

convergence behavior of the algorithm: the sharper the descent and the closer to the x-axis, the better.

Now, the algorithms are somewhat sensitive to the choice of the sequences a_n, c_n and L_n. After some experimentation with several sequences to select the best for each of the two algorithms, we obtained the following choices: $a_n = \frac{0.5}{n}$ for the stepsize method when $\sigma \in \{1, 2\}$ and $a_n = \frac{0.01}{n}$ $\sigma \in \{5, 10\}$; $L_n = n, a = 0.01$ for the sampling method; finally, $c_n = 0.1^{-1/3}$ for both methods.

Figure 1 shows the convergence profiles for the stepsize algorithm for various values of $\sigma \in \{1, 2, 5, 10\}$ (s is the figures) starting from an initial value $X^{(0)} = 5$. Observe that high values of noise result in poor performance.

Figure 2 shows equivalent convergence profiles for the sampling-controlled method. For comparison with the stepsize method, two curves, one for each method, using the same problem parameters are plotted in Figure 3. Here, the value of σ is 5.0. Clearly, for low amounts of noise ($\sigma = 1$) there is very little difference between the two approaches, but when the noise is significant ($\sigma > 2$), sampling-control appears to perform better than stepsize-control.

Figure 4 evaluates the impact of parallelism on the sampling-controlled method. To assess this impact, we estimate the time-taken on an m-processor parallel machine by dividing the number of samples by the number of processors and adding some overhead, as would be incurred in an actual parallel implementation. In the experiment, we take the overhead to be about 30%, which we believe is more than reasonable for our easily parallelizable application (All L_n samples per iteration are independent and so there are no synchronization costs). The figure shows convergence profiles for different values of m, the number of processors: $m \in \{1, 8, 16, 32\}$. As the figure shows, parallelism clearly speeds up convergence, even with only 8 processors.

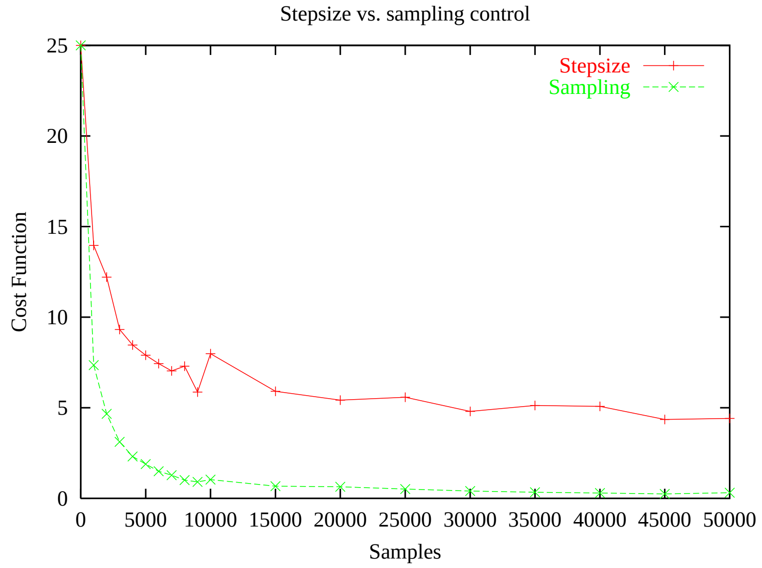

Figure 3: Comparison of stepsize- and sampling-control

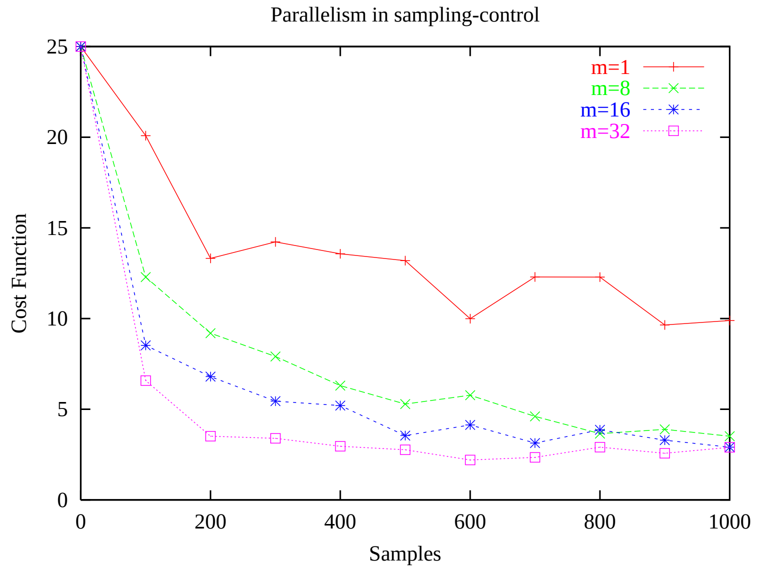

Figure 4: Assessing parallelism in sampling-control

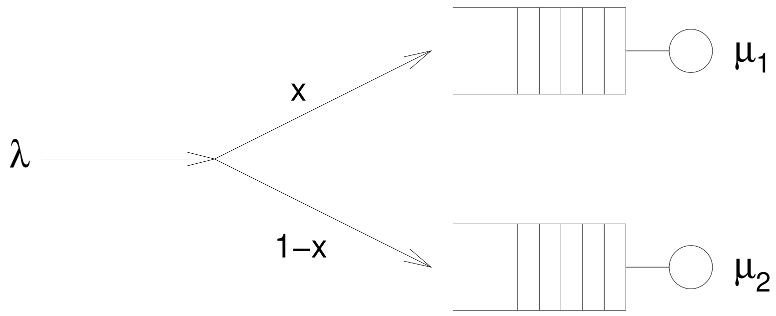

Figure 5: A routing problem

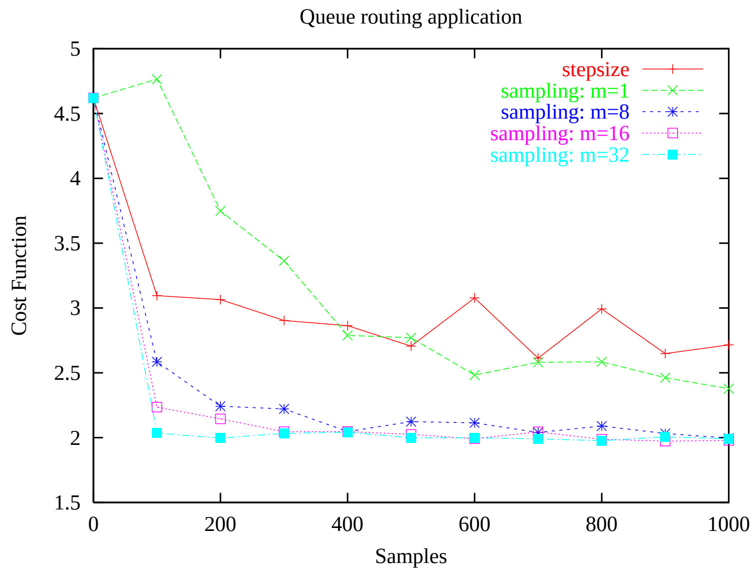

Figure 6: Queue routing application: a comparison

We now apply both the stepsize-control and sampling-control approaches to the routing problem shown in Figure 5. In this problem, jobs or packets arrive at rate λ to a 2-server or 2-processor system in which each server has a difference service rate. A fraction x of jobs are routed to the first server, and the remainder, $1 - x$, are routed to the second server; for a particular job, the routing is decided using a Bernoulli trial with probability x. The goal is to minimize the average sojourn time through the system over all possible values of the routing parameter x. If the arrival stream is Poisson and the service distributions are exponential, then the sojourn time can be analytically expressed and the optimal value of x can be easily determined. By using a simulation of this system, and estimates of sojourn time, we can compare the performance of the two stochastic approaches, using our theoretical solution as the benchmark.

Figure 6 plots the convergence profile of stepsize-control and four cases sampling-control corresponding to different numbers of parallel processors. The optimal solution is approximately 1.91. Only the first 1000 samples are shown because the convergence profiles are the most significant in this range; both algorithms eventually converge if iterated sufficiently. The pattern in this application is similar to that of the quadratic function: there is somewhat better performance from sampling-control when it is parallelized, and comparable performance is obtained otherwise.

6 Summary

In this paper, we considered continuous-variable optimization problems in which the objective function is estimated, typically, using a (stochastic) simulation. Since random noise is reflected in function estimates, iterative optimization techniques that use such estimates must necessarily be modified to ensure convergence. For this purpose, we have presented a new derivative-free stochastic approximation algorithm: a sampling-controlled version of the the classic Kiefer-Wolfowitz algorithm [20]. In an earlier paper, we showed that the general approach was useful in removing estimator bias. In this paper, we show that the sampling-control approach lends itself easily to parallelization. Simulation results show that this approach is competitive with the equivalent stepsize-controlled method.

Acknowledgements: The author would like to thank Prof. Paul Dupuis, Division of Applied Mathematics, Brown University for valuable comments regarding this work and Dr. Alan Weiss, AT&T Labs, for helping identify results in his book [35] that are used in the proof of convergence.

References

[1] H.Arsham. Simulation Optimization and Sensitivity Analysis Resources, `http://ubmail.ubalt.edu/~harsham/ref/Refsim.htm`.

[2] F.Azadivar. Simulation Optimization Methodologies. *Prof. Winter Simulation Conf.*, 1999.

[3] Barto, A.G., and Anandan, P., "Pattern Recognizing Stochastic Learning Automata," *IEEE Transactions on Man, Systems & Cybernetics*, Vol. 15, pp. 360-374, 1985.

[4] A.Benveniste, M.Metivier and P.Priouret. Adaptive Algorithms and Stochastic Approximation, *Springer-Verlag*, Berlin, 1990.

[5] D.Bertsekas and J.Tsitsiklis. Parallel and Distributed Computation: Numerical Methods, *Athena Scientific*, 1996.

[6] Y.Carson and A.Maria, "Simulation Optimization: Methods and Applications", *Proc. Winter Simulation Conf.*, 1997, pp.118-126.

[7] C.Cassandras. Discrete-Event Systems: Modeling and Performance Analysis, *Irwin Inc.*, Homewood, IL, 1993.

[8] C.Cassandras, M.V.Abidi and D.Towsley, "Distributed Routing with Online Marginal Delay Estimation", *IEEE Trans. Comm.*, Vol. 38, No.3, 1990, pp.348-359.

[9] F.Chang and L.Wu, "An Optimal Adaptive Routing Algorithm", *IEEE Trans. on Automatic Control*, pp. 690-700, Vol. AC-31, No. 8, Aug 1986.

[10] D.C.Chin, "Comparative Study of Stochastic Algorithms for System Optimization Based on Gradient Approximations," *IEEE Transactions on SMC - Part B: Cybernetics*, vol. 27, no. 2, pp. 244 - 249 April 1997

[11] P.Dupuis and H.J.Kushner, "Stochastic Approximation and Large Deviations: Upper Bounds and w.p.1 Convergence", *SIAM J. Control and Opt.*, Vol. 27, No. 5, pp.1108-1135.

[12] P.Dupuis and R.Simha, "On Sampling-Controlled Stochastic Approximation", *IEEE Trans. on Automatic Control*, Vol. 36, No.8, August 1991, pp.915-924.

[13] R.G.Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," pp. 73-85, *IEEE Trans. on Communications*, Vol COM-25, 1977.

[14] P.W.Glynn, "Optimization of Stochastic Systems", *Proc. Winter Simulation Conference*, 1986, pp. 52-58.

[15] B.Hajek, "Stochastic Approximation Methods for Decentralized Control of Multiaccess Communications", *IEEE Trans. on Communications*, pp. 176-184, Vol. IT-31, No.2, March 1985.

[16] Heal, G., "Planning Without Prices," *Review of Economic Studies*, Vol. 36, pp. 346-362, 1969.

[17] P.Heidelberger and D.Towsley, "Sensitivity Analysis from Sample Paths Using Likelihoods", *Management Science*, **35**, 1475-1488, December 1989.

[18] P.Heidelberger, X-R.Cao, M.A.Zazanis and R.Suri, "Convergence Properties of Infinitesimal Perturbation Analysis Estimates", *Managment Science*, Vol. 34, No. 11, Nov 1988, pp. 1281-1302.

[19] Ho, Y.C., Servi, L., and Suri, R., "A Class of Center-Free Resource Allocation Algorithms," *Large Scale Systems*, Vol. 1, pp. 51-62, 1980.

[20] J.Kiefer and J.Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function", *Ann. Math. Stat.*, Vol. 23, 1952, pp.462-466.

[21] H.J.Kushner and D.S.Clark. Stochastic Approximation Methods for Constrained and Unconstrained Systems, *Springer Verlag*, New York, 1978.

[22] S.Lakshmivarahan. Learning Algorithms: Theory and Applications, *Springer-Verlag*, New York.

[23] Y.H.Lee, Y.B.Kim and K.J.Park, "Single Run Optimization Using the Reverse Simulation Method", *Proc. Winter Simulation Conf.*, 1997, pp.187-193.

[24] Y.T.Leung and R.Suri, "Convergence of a Single Run Simulation Optimization Algorithm", *Proc. American Control Conf.*, 1988,

[25] L.Ljung, "Analysis of Recursive Stochastic Algorithms," *IEEE Trans. on Automatic Control*, Vol AC-20, 1977, pp. 551-575.

[26] D.G.Luenberger. Linear and Nonlinear Programming, *Addison-Wesley*, Reading, MA, 1984.

[27] M.S.Meketon, "Optimization in Simulation: A Survey of Recent Results", *Proc. Winter Simulation Conf.*, 1987.

[28] M.Minoux. Mathematical Programming: Theory and Algorithms, *Wiley*, 1986, pp. 113-116.

[29] K.Narendra and M.A.L.Thathachar. Learning Automata: An Introduction, *Prentice-Hall*, NJ, 1989.

[30] G.Pflug. Optimization of Stochastic Models: The Interface Between Simulation and Optimization, *Kluwer Acad. Pub.*, Norwell, MA, 1996.

[31] M.I.Reiman and A.Weiss, "Sensitivity Analysis for Simulations Via Likelihood Ratios", *Operations Research*, Vol. 37, pp. 830-844, September 1989.

[32] H.Robbins and S.Munro, "A Stochastic Approximation Method," *Ann. Math. Stat.*, Vol 22, 1951, pp. 400-407

[33] V.K.Rohatgi. An Introduction to Probability Theory and Mathematical Statistics, *Wiley*, 1976, chapter 6, pp.263-275 and chapter 8, pp.333-349.

[34] R.Y.Rubenstein. Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks, *Krieger Pub.*, 1992.

[35] A.Shwartz and A.Weiss. Large Deviations for Performance Analysis: Queues, Communications and Computing, *CRC Press*, 1995.

[36] R.Simha and J.F.Kurose, "Stochastic Approximation Schemes for a Load Balancing Problem", *Proc. 27th Annual Allerton Conference*, Allerton, Illinois, 1989.

[37] R.Simha and J.F.Kurose, "Relative Reward Strength Algorithms for Learning Automata", *IEEE Trans. on Systems, Man & Cybernetics*, Vol. 19, No. 2, March 1989, pp. 388-398.

[38] R.Simha and J.F.Kurose, "On-line Minimization of Call Setup Times Via Load Balancing: A Stochastic Approximation Approach", *IEEE Trans. Communications*, Vol. 42, No. 2, 1994, pp.1141-1152.

[39] J.C.Spall. Stochastic Optimization and the Simultaneous Perturbation Method, *Prof. Winter Simulation Conf.*, 1999.

[40] T.E.Stern, "A Class of Decentralized Routing Algorithms Using Relaxation", *IEEE Trans. on Communications*, pp. 1092-1102, Vol. COM-25, No. 10, Oct 1977.

[41] R.Suri, "Perturbation Analysis: The State of the Art and Research Issues Explained Via the GI/G/1 Queue", *Proc. IEEE*, Vol. 77, No. 1, Jan 1989, pp. 114-137.

[42] S.R.S.Varadhan. Large Deviations and Applications, *SIAM*, Philadelphia, 1984.

[43] M.T.Wasan. Stochastic Approximation, *Cambridge Univ. Press*, 1969.