(2)

STOCHASTIC OPTIMIZATION BY SIMULATION: SOME EXPERIMENTS WITH A SIMPLE STEADY-STATE QUEUE

by

Pierre L'Ecuyer, Nataly Giroux, and Peter W. Glynn

TECHNICAL REPORT No. 31

February 1989

Prepared under the Auspices
of
U.S. Army Research Contract
DAAL-88-K-0063
-03-

DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

Stochastic Optimization by Simulation: Some Experiments With a Simple Steady-state Queue

Pierre L'Ecuyer[1], Nataly Giroux[2] and Peter W. Glynn[3]

ABSTRACT

New approaches like perturbation analysis and the likelihood ratio method have been proposed recently to estimate the gradient of a performance measure with respect to some continuous parameters in a dynamic stochastic system. In this paper, we experiment the use of these estimators in stochastic approximation algorithms, to perform so-called "single-run optimizations". We also compare them to finite difference estimators, with and without common random numbers. The experiments are done on a simple M/M/1 queue. The performance measure involves the average system time per customer, and the optimal solution is easy to compute analytically, which facilitates the evaluation of the algorithms. We also demonstrate some properties of the algorithms. In particular, we show that using perturbation analysis, the single-run optimization converges to the optimum even with a fixed (and small) number of ends of service per iteration, while under the same conditions, the algorithm that uses the finite difference estimators converges to the wrong answer.

Keywords: simulation, stochastic approximation, likelihood ratios, perturbation analysis.

[1]Operations Research Department, Stanford University, on leave from Département d'informatique, Université Laval, Ste-Foy, Québec, Canada, G1K 7P4.

[2]Département d'informatique, Université Laval.

[3]Operations Research Department, Stanford University, Stanford, CA94305.

1. Introduction

Simulation has traditionally been used to *evaluate* the performance of complex systems, especially when analytic formulas are not available, but rarely to perform optimization. Consider a (stochastic) simulation model parametrized by a vector θ of continuous parameters, and suppose one seeks to minimize the expected value $\alpha(\theta)$ of some objective function. In principle, if $\alpha(\theta)$ is well behaved, one could estimate its gradient by simulation, and use adapted versions of classical non-linear programming algorithms. One way to estimate the gradient is to use finite differences. However, in most practical applications, obtaining a reasonably accurate estimate often require unacceptably high amounts of computer time.

Recently, new ways have been proposed to estimate the gradient of a performance measure (defined as a mathematical expectation), with respect to continuous parameters, by simulation [3, 4, 5, 6, 17, 18, 21, 23]. For "steady-state" simulations, a "single-run" optimization scheme has also been suggested [14, 22]. Combined with appropriate variance reduction techniques, these methods could enlarge substantially the class of stochastic optimization problems that can be solved. However, a good amount of research remains to be done on the theoretical properties of these methods (some of them were proposed as "heuristics"), and empirical investigations should be done.

In this paper, we report the results of numerical experiments performed on simple queueing systems. The idea was to run the algorithms (and their variants) on a problem for which we could compute analytically the optimal solution. Such an experiment was first undertaken in [22], but these authors looked at only two methods, which they presented as heuristics. One was based on perturbation analysis (PA) and the other was an adaptation of the Kiefer-Wolfowitz (KW) algorithm. They observed empirically that for the problem considered, the former method (PA) was converging much faster than the latter (KW). We prove in this paper that for the example they examined, their first method converges to the optimal solution, while their second might converge to the wrong answer. We then suggest new variants of KW that converge to the optimal solution. For the (simple) examples considered, one of these variants (which uses common random numbers and increases the simulation length at each iteration) appears to be competitive with PA. We also experiment with different variants of the likelihood ratio (LR) method [3, 4, 17, 19] (also called the score function (SF) method). Our numerical experiments deal with examples where the decision parameter vector θ has only one component. The multidimensional case certainly involves more intensive computations. As always, since these experiments were done on specific numerical examples, one should be careful in making any generalizations.

In section 2, we briefly describe a stochastic approximation algorithm for steady-state simulation optimization in a general setting. More details are given in the appendix, where we recall known convergence results, with slight adaptations. Our aim is not to give the most general results, but as stated, the theorems are general enough for a fairly large class of applications. At each iteration, the algorithm requires an estimate of the gradient $\nabla\alpha(\cdot)$ at a given point, and section 3 reviews some techniques to obtain that estimate. Our presentation is made in the steady-state setting, but it also applies (with some simplification) to terminating simulations, or to the case of infinite horizon total discounted cost, by taking $\alpha(\theta)$ as the total expected (possibly discounted) cost at parameter level θ. The initial simulation state is then fixed (or could be part of the parameter, or random with known distribution). For terminating simulations, many things simplify since the initial bias problem disappears. In section 4, we consider a simple example, similar to the one studied in [22]: one aims to minimize a function of the average system time per customer, in a M/M/1 queue for which the decision variable is the parameter of the service time distribution. We give the results of an extensive numerical investigation, and prove some properties. In the conclusion, we comment on the possible behavior of other variants of this example, and mention prospects for further research.

2. A stochastic approximation scheme

2.1. The general form of the algorithm

Consider a discrete event simulation whose evolution law depends on a *parameter* vector θ, where $\theta \in \Theta$, a compact convex subset of $\mathbf{R}^d$. Let $\alpha(\theta)$ be the steady-state *cost* of running the system at parameter level θ, and suppose one is interested in minimizing $\alpha(\theta)$ over Θ. We assume that $\alpha(\cdot)$ is continuously differentiable.

We consider a stochastic approximation algorithm of the form:

$$\theta_{n+1} := \pi_\Theta(\theta_n - \gamma_n Y_n). \tag{1}$$

where $\theta_0 \in \Theta$ is fixed (or random with known distribution). Here, for $n \geq 0$, θ_n is the parameter value at the beginning of iteration n, Y_n is an estimate of the gradient $\nabla\alpha(\theta)$ obtained at iteration n, $\{\gamma_n, n \geq 0\}$ is a positive sequence decreasing to 0 and such that $\sum_{n=0}^{\infty} \gamma_n = \infty$, and π_Θ denotes the projection on the set Θ (i.e. $\pi_\Theta(\theta)$ is the closest point to θ in Θ). Note that this can also be generalized (and the results will still hold) to the case where a different subsequence $\{\gamma_n, n \geq 0\}$ is chosen for each component of θ.

Let s_n denote the *state* of the simulation at the beginning of iteration n (description of all the objects in the system, event list, etc.). We assume that enough information is kept in s_n so that the pair (θ_n, s_n) evolves as a (possibly non-homogeneous) Markov chain. Let s_0 be the initial state. There are different ways of obtaining Y_n, and since it must be obtained in finite time, the estimator will usually be (sometimes only slightly) biased. The computation of Y_n involves say $K_r \geq 1$ simulation subruns, where for $k = 1, \ldots, K_r$, the k-th subrun is performed at parameter value $\theta_{nk} = \phi_{nk}(\theta_n)$, form initial state s_{nk}, and for duration T_{nk} (possibly random). That duration can be expressed for instance in terms of the simulation clock (units of simulation time), or as a number of regenerative cycles, or as a number of customers to be served (in a queueing system), etc. The functions ϕ_{nk}, the rules for selecting s_{nk}, and the probability distribution (or value) of T_{nk} all depend on the algorithm. We assume that there is also a rule for choosing s_{n+1} at the end of these K_r simulation subruns (it could be for instance the final state of one of the K_r subruns). Often, $K_r = 1$, $s_{n1} = s_n$, $\theta_{nk} = \theta_n$ and s_{n+1} is the state of the simulation model at the end of iteration n. For a symmetric finite difference scheme, one has $K_r = 2d$ and each θ_{nk} is θ_n plus a constant times a unit vector.

Denote by $E_n(\cdot)$ the conditional expectation $E[\cdot \mid \theta_n, s_n]$. We can write Y_n as

$$Y_n = \nabla\alpha(\theta_n) + \beta_n + \epsilon_n \tag{2}$$

where $E_n[\epsilon_n] = 0$, so that β_n represents the (conditional) bias on Y_n given (θ_n, s_n).

2.2. About the convergence

Sufficient conditions for the convergence of (1) to an optimum (or of similar algorithms, often without the projection operator) have appeared in many places (see [9, 10, 11, 15, 16, 18] and other references cited there). Some sets of conditions imply almost sure convergence, others imply only some form of weak convergence, but are often easier to verify. In the appendix, we give two sets of conditions that are slight adaptations from [9, 11]. One set is for convergence with probability one, the other one is for weak convergence.

Roughly speaking, the algorithm should converge almost surely if γ_n goes to zero fast enough to damp out in some way the variation due to the ϵ_n's, and that convergence should be to a zero of the gradient $\nabla\alpha(\theta)$ if β_n goes to zero.

3. Ways of estimating the gradient

One crucial ingredient for the algorithm sketched in the previous section is an efficient gradient estimation technique. In this section, we survey some possibilities.

3.1. Kiefer-Wolfowitz (KW)

This method is described for instance in [9, 18], without the projection operator. Here, Y_n is a finite difference estimator. Take a positive sequence $\{c_n, n \geq 0\}$ that converges to 0. Let e_i denote the d-dimensional unit vector with a 1 in position i. The estimation of Y_n involves $K_r = 2d$ simulation subruns, each subrun yielding an estimation W_{ni}^- or W_{ni}^+ of the "average" cost at a given parameter setting. More precisely, for $i = 1, \ldots, d$, we simulate from some initial state s_{ni}^- at parameter value $\pi_\Theta(\theta_n - c_n e_i)$ for duration T_n to obtain W_{ni}^-, and we simulate from state s_{ni}^+ at parameter value $\pi_\Theta(\theta_n + c_n e_i)$ for duration T_n to obtain W_{ni}^+. We then compute

$$Y_n = \sum_{i=1}^{d} \frac{(W_{ni}^+ - W_{ni}^-)e_i}{\Delta_{ni}}, \tag{3}$$

where

$$\Delta_{ni} = \|\pi_\Theta(\theta_n + c_n e_i) - \pi_\Theta(\theta_n - c_n e_i).\| \tag{4}$$

Here, $T_{nk} = T_n$ for $k = 1, \ldots, 2d$. Again, T_n can be a number of (simulated) time units, or a number of regenerative cycles, or a number of observations of some sort, etc. Usually, W_{ni}^- and W_{ni}^+ are averages over these "pieces" of simulation of "duration" T_n. We can also generalize this scheme by taking different subsequences $\{c_n, n \geq 0\}$ for the different components of θ. Note that problems might occur at the boundary of Θ: Δ_{ni} can be much smaller than $2c_n$ or may even be zero in bad cases. But we will not discuss this problem any further here. If Θ is an interval (or more generally an hyperrectangle), as will be the case for the examples treated here, it is easy to see that $\Delta_{ni} \geq c_n$.

There are two sources of (conditional) bias in Y_n. One is due to the fact that we use finite differences. Call it β_n^D. It goes to 0 as $n \to \infty$ since $c_n \to 0$. The second one is due to the initial state at the beginning of each iteration. We have

$$E_n[W_{ni}^-] = \alpha(\pi_\Theta(\theta_n - c_n e_i)) + \beta_{ni}^- \tag{5}$$

$$E_n[W_{ni}^+] = \alpha(\pi_\Theta(\theta_n + c_n e_i)) + \beta_{ni}^+ \tag{6}$$

where β_{ni}^- and β_{ni}^+ are the respective (conditional) bias on the estimate average costs due to the initial states s_{ni}^- and s_{ni}^+ (E_n is defined in section 2.1). This second (conditional) bias is then

$$\beta_n^I = \sum_{i=1}^{d} \frac{(\beta_{ni}^+ - \beta_{ni}^-)e_i}{\Delta_{ni}}. \tag{7}$$

To satisfy assumption S2 of the appendix, we need that $\beta_n = \beta_n^D + \beta_n^I \to 0$ as $n \to \infty$. In most practical situations, the bias β_{ni}^+ and β_{ni}^- decrease with T_n at a rate of approximately $1/T_n$. In that case, we will have $\beta_n^I \to 0$ if $\Delta_{ni} \geq \kappa_c c_n$ for some constant $\kappa_c > 0$ and if

$$1/(T_n c_n) \to 0. \tag{8}$$

As c_n decreases to zero, the variance on Y_n usually increases to infinity. However, if we assume that the variance of the sum of the numerators in (3) is bounded, assumption S4 of the appendix can be satisfied by choosing the sequences in such a way that

$$\sum_{n=0}^{\infty} (\gamma_n/c_n)^2 < \infty. \tag{9}$$

A reasonable choice might be to take for instance $T_n = t_a + t_b n$, $\gamma_n = \gamma_0/(n+1)$ and $c_n = c_0 n^{-1/3}$ for appropriate constants t_a, t_b, γ_0 and c_0.

Note that this method may work even if $\alpha(\cdot)$ is not differentiable. For the unconstrained case ($\Theta = \mathbb{R}^d$), if $\alpha(\cdot)$ is twice continuously differentiable, if θ^* satisfies S5 of the appendix, if we neglect the bias β_n^I, under some additional conditions on the variance of the cost estimators, and if the sequences are chosen as suggested above with $t_b = 0$, then the algorithm converges almost surely to θ^* and the convergence rate is of the order of $n^{-1/3}$ (see Theorem 2.3.5 and chapter VII in [9]). However, β_n^I is zero only for terminating simulations or when we can exploit the regenerative structure. Otherwise, taking $t_b = 0$ may lead to disaster, as will be illustrated in the next section.

One simple way to choose the initial states of the subruns is as follows. Start the first subrun from state s_n, then take the terminal state of any given subrun as the initial state of the next one. The terminal state of the last subrun will become s_{n+1}, the initial state for the next iteration. But still, we can permute the $2d$ subruns of a given iteration in any given way, and select the terminal state of any subrun for s_{n+1}. It is not clear what is the best way of doing this, if any. In any case, the KW method is usually plagued by a huge variance on the Y_n, which makes it converge very slowly, at least when the subruns are performed with "independent" random numbers.

Glynn [3] describes an alternative KW approach based on regenerative analysis. It eliminates the bias β_n^I, but the variance is usually much higher.

The package SAMOPT [1] is an implementation of the stochastic optimisation algorithm with KW, with specially tuned parameters. It was designed for terminating simulations. It also replaces Y_n by its sign.

3.2. Kiefer-Wolfowitz with common random numbers (KWC)

One way to reduce the variance in KW is to use common random numbers across the subruns at each iteration, start all the subruns from the same state, and synchronize. Of course, there is no guaranteed variance reduction, but since the subruns are aimed at comparing very similar systems, especially when c_n is small, considerable variance reductions might be obtained. The starting state s_{n+1} for the next iteration can be anyone of the $2d$ terminal states. A heuristic rule is to choose the state that was obtained from the subrun with the parameter value the closest to the new parameter value θ_{n+1}.

Implementing this method for complex simulations is not without pain. Saving the simulation state means saving the states of the random number generators, the event list, the values of all variables related to the model (but not those related to the top level algorithm), all the objects in the model, etc. In practice, many objects in the model are pointers to data structures that can be created, modified or destroyed dynamically, and whose types have been defined by the programmer. When saving the state of the system, one cannot only save the pointer values, but must make an explicit "backup" copy of all these structures. When restoring the system to a given state, these must be recopied again. Usually, the simulation package cannot do that and specific code must be written. In fact, it would be very difficult to implement "state saving" facilities in a general simulation package, because usually, the package has no way to know with certainty the structures of all the dynamic objects created by the user. All this implies overhead not only for the computer, but also for the programmer. Another source of programming overhead in KWC comes from the need to insure synchronization of the random numbers across the subruns.

When c_n is small, there is sometimes little change between the sample paths of the $2d$ subruns. One could then ask: is it possible to perform only *one* subrun and trace the few changes ? It is indeed sometimes possible, and this idea leads to what is called *finite perturbation analysis*. Taking that to the limit when c_n goes to zero, one obtains *infinitesimal perturbation analysis*.

3.3. Perturbation analysis (PA)

In this paper, PA always refers to infinitesimal perturbation analysis [5, 6, 21, 23]. The basic idea is to generate a sample path ω, viewed as a sequence of $U(0,1)$ variates, and for ω fixed, observe the effect of an infinitesimal perturbation on θ by propagating it over the sample path. Such a propagation can be done rather easily if one assumes that the perturbation on θ does not change the sequence of events, but only makes them "slide smoothly" in time. The gradient estimate is then taken as the gradient of the sample objective function for that fixed value of ω, say $\nabla_\theta h(\theta,\omega)$. Unfortunately, it is not always true that

$$E[\nabla_\theta h(\theta,\omega)] = \nabla_\theta E[h(\theta,\omega)], \tag{10}$$

since we cannot always permute the derivative and the expectation. Heidelberger et al. [6] give conditions under which PA works correctly. If PA does not work for the original system, various devices can sometimes be used to "smooth out" or transform the original problem into a problem for which PA will work correctly (see [5] for instance). These devices are usually problem-dependent. In principle, when (10) holds, PA can be viewed as a limiting version of KWC as each c_n becomes infinitesimal. However, as we will see in section 4, implementation "details" can often make a *big difference* between PA and "infinitesimal" KWC. One big advantage of PA when it works is that it requires only *one* simulation subrun per iteration, compared to $2d$ for KW or KWC.

3.4. Likelihood ratio (LR)

The likelihood ratio gradient estimation method has been introduced recently by Glynn [3, 4], Reiman and Weiss [17], and Rubinstein [19, 20] (who calls it the *score function* method). The basic idea is that $\alpha(\theta)$ can usually be viewed as the expectation of some function of θ and of the sample path ω, say $h(\theta,\omega)$, with respect to some probability measure $P_\theta(\cdot)$ over some sample space Ω. Here, ω represents all the randomness that drives the system, so that when ω is fixed, everything becomes deterministic. Typically, ω can be viewed as a sequence of random variables, some of which have a distribution that depends on θ. Usually, one cannot differentiate this expectation directly by differentiating inside the integral, because $P_\theta(\cdot)$ depends on θ, but for some $\theta_n \in \Theta$, one can rewrite

$$\alpha(\theta) = \int_\Omega h(\theta,\omega) P_\theta(d\omega) = \int_\Omega \left[h(\theta,\omega) \frac{P_\theta(d\omega)}{P_{\theta_n}(d\omega)} \right] P_{\theta_n}(d\omega). \tag{11}$$

and differentiate $\alpha(\cdot)$ by differentiating the bracketted term with respect to θ inside the integral (θ_n is viewed as a constant). This expression can then be evaluated at $\theta = \theta_n$ to obtain

$$\nabla\alpha(\theta_n) = \int_\Omega \left[\left(\nabla_\theta h(\theta,\omega) + h(\theta,\omega) \frac{\nabla_\theta P_\theta(d\omega)}{P_{\theta_n}(d\omega)} \right) \Big|_{\theta=\theta_n} \right] P_{\theta_n}(d\omega). \tag{12}$$

The bracketted expression in (12) can be used to estimate $\nabla\alpha(\theta_n)$. As in PA, only one simulation subrun is required to estimate the gradient. (In principle, the expression can also be evaluated at any value of $\theta \neq \theta_n$, yielding an estimate of the gradient everywhere in Θ, obtained by a single simulation. The method can also be generalized to higher order derivatives.)

Note that the above reasoning holds only if the derivative and expectation can be interchanged (this can be done under some regularity conditions, see [17, 20]), and if the likelihood ratio (or Radon-Nikodym derivative) exists in a neighborhood of θ_n, i.e. if the sets of positive $P_\theta(\cdot)$-measure are the same for all θ in that neighborhood. These conditions limit the method. For instance, threshold-type parameters that influence deterministically the occurence (or occurence time) of some events are ruled out. Another case where it won't work is when a component of θ represents a transition probability of some sort, and that probability goes to 0 or 1. This could happen easily when optimizing transition probabilities. Also, the regularity conditions permitting one to interchange the derivative and expectation are often satisfied, but not always.

For steady-state simulations, there is again the bias problem, since only a finite portion of the sample path can be simulated. A simple remedy is to use the same idea as in KW: increase the simulation length at each iteration. However, the variance of Y_n here usually increases linearly with T_n, so T_n should not be increased too rapidly. Assuming that the variance is proportional to T_n, one can take $\gamma_n = \gamma_0/n$ and $T_n = t_a + t_b n^p$ for $0 < p < 1$, and the conditions S1 and S4 of the appendix will be satisfied by taking $\delta_n = n^{-p/2}$. Another choice could be $T_n = t_a + t_b \ln n$. In any case, the fact that the variance increases with the simulation length is an important limitation of LR in general. We should expect LR to work much better for terminating simulations for which only a small number of random variates are generated using a given parameter.

Other variants of the LR approach circumvent the bias problem by using a regenerative approach [3, 4, 17]. If the system possesses a readily identifiable regenerative structure, $\alpha(\cdot)$ can be written as the quotient of two functions, and a likelihood ratio approach can be used to obtain an estimator of the derivative of the quotient, for each component of θ. See [3, 4, 17] for more details. The methods proposed in [4, 17] still have some bias, since they

estimate the expectation of a ratio, but the variance does not increase with the simulation length. It decreases linearly instead. Also, the bias goes to zero. Here, T_n can be taken as the number of regenerative cycles at iteration n. On the other hand, algorithm B in [3] proposes an unbiased estimator. However, its variance is usually very high. According to our experience, its practical applicability appears to be somewhat limited.

4. Example: a M/M/1 queue

4.1. The basic model and experimental setup

This example is strongly inspired from Suri and Leung [22]. Consider a M/M/1 queue with arrival rate $\lambda = 1$ and mean service time $\theta \in \Theta = [a, b]$, where $0 < a \leq b < 1$. Here, θ is the parameter to be optimized. Let $C_1 > 0$ be a constant, and let $w(\theta)$ be the average sojourn time in the system per customer, in steady-state, at parameter level θ. The objective function (to be minimized) is defined by

$$\alpha(\theta) = w(\theta) + C_1/\theta. \tag{13}$$

The optimal value θ^* can be computed quite easily in this case. Indeed, $w(\theta) = \theta/(1-\theta)$ and $\theta^* = \sqrt{C_1}/(1 + \sqrt{C_1})$ (if this value is not in $[a, b]$, the optimum is at the nearest boundary). But we can ignore momentarily these results, and try to solve the problem using the stochastic algorithm described in section 2, combined with different variants of the gradient estimation techniques described in the previous section. The solutions can then be compared to the true optimal solution for an empirical evaluation. Henceforth, we will refer to these variants as different *algorithms*.

We actually performed such an experiment as follows. For each algorithm, we made N simulation runs, each yielding an estimation of θ^*. The N initial parameter values were randomly chosen, uniformly in $[a, b]$, and the initial state (s_0) was an empty system. Across the algorithms, we used common random numbers and the same set of initial parameter values. Each run was stopped after a (fixed) total of T ends of service, but that stopping criterion was checked only between iterations. Hence, all algorithms had about the same chance and, if we neglect the differences in overhead for the gradient estimation techniques, used about the same CPU time. (The overhead was quite low in general, except for very small values of T_n, like $T_n = 1$.) The programs were written using SIMOD [12], a simulation package based on the language Modula-2. Each customer had a record associated with it, to keep track of its arrival time. These records were created dynamically and the waiting customers were put explicitly into a linked queue. For this example, some methods, like PA for instance, do not require keeping track of the individual arrival times, and in that case we could have saved a fair amount of CPU time by keeping only a counter instead of a linked queue. However, we insisted on using exactly the same simulation program for all algorithms. In fact, the simulation model and the algorithms were implemented in two

different modules, the latter being totally model independent. We also avoided using any variance reduction technique.

For each algorithm variant, we computed the empirical mean $\bar{\theta}$, standard deviation s_d and standard error s_e of the N retained parameter values, and also traced graphics of the evolution of these quantities as functions of the total number of ends of service. If y_i denotes the retained parameter value for run i (i.e. the last value of θ_n), the above quantities are defined by

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N} y_i; \qquad s_d^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{\theta})^2; \qquad s_e^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \theta^*)^2. \tag{14}$$

We also computed a confidence interval I_θ on the expectation of $\bar{\theta}$, assuming that $\sqrt{N}(\bar{\theta} - E(\bar{\theta}))/s_d$ follows a Student distribution with $N-1$ degrees of freedom.

4.2. Implementing the gradient estimation methods

Here are some specific implementation details for the different gradient estimation techniques. In this example, T_n represents the number of ends of service for each subrun of iteration n, except for the regenerative methods, where it represents the number of regenerative cycles per subrun. Hence each iteration ended at a time when a customer was leaving the system. Of course, if the queue was not empty at that point, we were careful to generate the new service time only at the beginning of the next iteration, i.e. *after* the parameter was modified. For all the methods except KWC, the final state of every simulation subrun was taken as the initial state for the next one. For the regenerative approaches, a new regenerative cycle began at the end of each busy period.

For the KW method, the average cost was estimated for each subrun j of iteration n by $C_1/\theta_n + z_{nj}$, where z_{nj} was the average system time for the customers who *leaved* during that subrun.

For PA, the gradient estimation can be computed as follows (see [22, 23]). Every time a customer leaves the system, we look at the elapsed time since the beginning of the busy period. Let ν_n be the average of these times for the T_n customers leaving at iteration n. The gradient estimate is then

$$Y_n = \nu_n/\theta_n - C_1/\theta_n^2. \tag{15}$$

For LR, we can view momentarily $\alpha(\theta_n)$ in (11) and (12) as representing the expected average cost *during the subrun of iteration* n given the initial state s_n, and ω as the T_n

service times generated during that iteration. Hence, at $\theta = \theta_n$,

$$\frac{\nabla_\theta P_\theta(d\omega)}{P_\theta(d\omega)} = \frac{\partial}{\partial\theta} \ln P_\theta(d\omega) = \frac{\partial}{\partial\theta} \ln \left(\prod_{j=1}^{T_n} f_\theta(\zeta_j) \right), \tag{16}$$

where ζ_j is the service time of the j-th customer and $f_\theta(\zeta) = (1/\theta)\exp(-\zeta/\theta)$ is the service time density. After easy manipulations, one obtains

$$Y_n = \frac{1}{\theta_n^2} \left(\left(\sum_{j=1}^{T_n} \zeta_j - T_n\theta_n \right) z_n - C_1 \right) \tag{17}$$

where z_n is the average system time for the customers who leaved during that subrun.

Explicit formulas for the quantities involved in algorithm A of [4], which implements a regenerative approach, can be derived in a similar way. In that case, one has

$$Y_n = \frac{Q_1Q_5 - Q_2Q_4}{Q_1^2} - \frac{C_1}{\theta_n^2} \tag{18}$$

where for $k = 1, 2, 4, 5$,

$$Q_k = \sum_{i=1}^{T_n} Q_{jk},$$

and for $j = 1, \ldots, T_n$, Q_{j1} is the number of departures during the j-th regenerative cycle, Q_{j2} is the total system time for those Q_{j1} customers who left during that cycle, Q_{j6} is their total service time (i.e. the length of the busy cycle), $Q_{j4} = Q_{j1}(Q_{j6} - Q_{j1}\theta_n)/\theta_n^2$, and $Q_{j5} = Q_{j2}Q_{j4}/Q_{j1}$.

We also implemented the regenerative algorithms described in [3] (with and without the arctan transformation), SAMOPT [1], and other variants, for which we will not give the details here. Algorithm B in [4] is not applicable for this example, because our cost function is not exactly of the form given in equation (4.2) of [4] (the residence time of a customer depends not only on its own service time, but also on previous ones). Kesten [7] has proposed an heuristic rule under which instead of diminishing γ_n at each iteration, one diminishes it only when the sign of the gradient estimate (for one parameter) is different from the one of the previous iteration (i.e. when the change on the parameter changes direction). The heuristic idea is that if the parameter keeps moving in the same direction, it should be because we are still far away from the optimum and so, lets give it a chance to move faster. That heuristic might help in situations where we start really far away from the optimum, and where the change on the parameter at each iteration tends to be very small. We have implemented this rule for some of our experiments.

4.3. Numerical results

Figure 1 is a graphical representation of some of the results of an experiment we made with $T = 10^6$, $N = 10$, $[a, b] = [0.01, 0.95]$ and $C_1 = 1$. The optimal solution is $\theta^* = 0.5$. The description of the algorithm variants appear on the left. For each variant, the white bar represents the standard deviation s_d and the dark bar represents the standard error s_e. Numerical values are given in the first column of table 1 (which appears further down). We computed the 95% confidence intervals I_θ as described in section 4.1, and the entries for which I_θ does not contain θ^* are indicated in table 1. KW, KWC, PA and LR have the same meaning as in section 3. LRR refers to algorithm A in [4]. PAR means perturbation analysis with a regenerative approach: each iteration is comprised of T_n regenerative cycles instead of T_n ends of service. The symbol (K) following the name of the algorithm signifies that Kesten's rule was used. The symbol (S) following KW means that instead of always simulating first at $\theta_n - c_n$ and then at $\theta_n + c_n$, we adopted the following heuristic rule for KW: if the parameter has just decreased, simulate first on the right (at $\theta_n + c_n$), otherwise simulate first on the left. The rationale is that if the parameter has just decreased, the current state has been reached by simulating at a parameter value larger than θ_n, and should thus be a statistically "better" initial state for a simulation at $\theta_n + c_n$ than at $\theta_n - c_n$ (and symmetrically if the parameter has just increased). In all cases, we had $\gamma_n = 1/n$ and for KW and KWC, we took $c_n = 0.1n^{-1/3}$. (Intuitively, $c_n = n^{-1/3}$ would have given much too wide finite difference intervals, since even after 1000 iterations, we would still have $2c_n = 0.2$, but we also tried it and, surprisingly, the results were about as good).

We can see that PA performs very well, even when T_n is fixed at a small constant. Surprisingly, KWC with a linearly increasing T_n is almost as good. When T_n is fixed to a small constant, KWC also converges rather quickly (small s_d), but the standard error s_e is very large, which means that it does not converge to the right place! Even for $T_n = 100$, the bias is still quite apparent and I_θ does not contain θ^*. KW has about the same behavior, but with larger variance. KW(S) is slightly better than KW, but not competitive with KWC or PA. PAR also has a bias problem when T_n is fixed. The problem is that with the regenerative approach, the number of ends of service during the T_n regenerative cycles is now random, and we get a bias due to the fact that we estimate a ratio with that number on the denominator. Of course, this bias goes to zero as T_n goes to infinity, and this is why PAR with $T_n = n$ works fine.

The LR methods in general have some trouble due to their large associated variance.

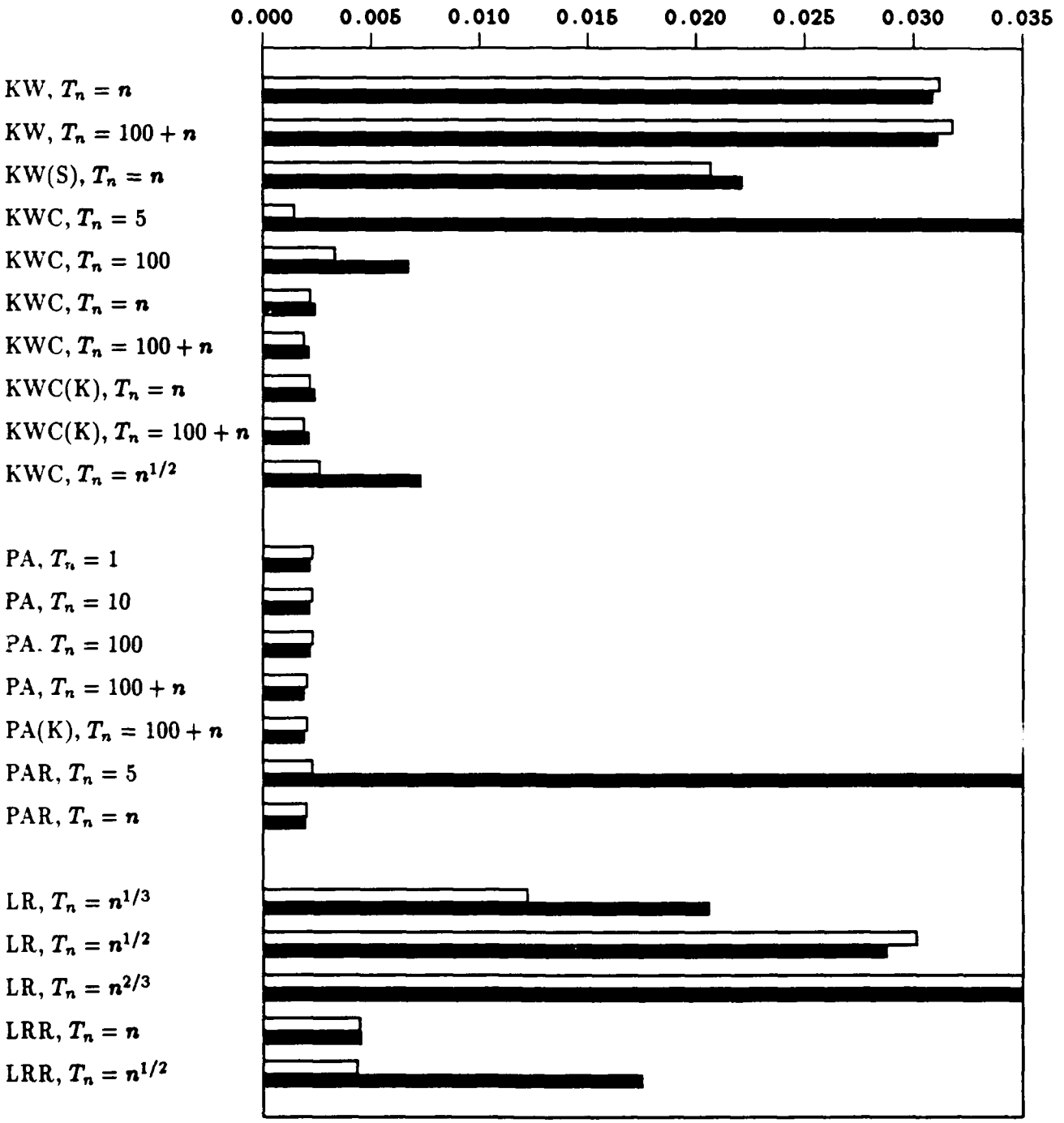

Figure 1: Performance of various methods for $C_1 = 1$, $T = 10^6$ and $N = 10$. White bar is s_d and black bar is s_e.

That variance stays low when T_n grows slowly, but then, the bias becomes more of a problem. LR with $T_n = n^p$ has large variance for large p, and for small p, the bias goes down much too slowly compared to the variance. The result is that the confidence interval I_θ, based on the N final values of θ_n, is very likely not to cover θ^*. This is what happens, for instance, with $p = 1/3$. The only LR variant that gives reasonably good results here is LRR, based on a regenerative approach, with T_n increasing linearly. With $T_n = n^{1/2}$, both LRR and KWC have the same bias problem as described above: the bias goes down too slowly and I_θ does not contain θ^*. Nevertheless, they converge (slowly) to the right answer (we verified it empirically with longer simulation runs). Kesten's rule doesn't appear to help for any of the methods. SAMOPT [1] and the algorithms described in [3] gave rather bad results (huge variances) and they do not appear in the figure. They are obviously not competitive, at least for this example. The problem with SAMOPT is that near the optimum, the gradient is very small in absolute value, and replacing it by its sign is really not a good idea. We also obtained bad results with other variants, like for instance PA with $T_n = 100 + n$ but $\gamma_n = 1/\sqrt{n}$ instead of $1/n$. Other sets of experiments were also done with $T = 10^5$ and the results were quite similar to the ones given here.

Figures 2 to 6 illustrate the evolution of the parameter as a function of the simulation length, for different algorithms. Each curve is the average of the $N = 10$ curves associated with the individual runs. On the horizontal axis, one has the total number of ends of service to date in the simulation run, and on the vertical axis, the average of the N values of θ_n observed at that point for the N runs. Each curve was actually drawn by linear interpolation, using 101 data points equally spaced on the horizontal axis (every 10^4 ends of service).

Figures 2–3 show how KWC converges much better than KW, and how KWC with $T_n = 5$ converges to somewhere around 0.65.

In figures 4–5, we see the convergence of PA for $T_n = 1$ and $T_n = 10$. It is interesting to observe here that the evolution of the parameter with these two values of T_n is almost identical. We also made other experiments with different (constant) values of T_n between 1 and 1000, and observed that the evolution was almost identical in all cases, independently of T_n. Note that all these simulations were done with common random numbers and same sets of starting parameter values. Moreover, even if the starting values are different, the evolution is again almost identical if common random numbers are used.

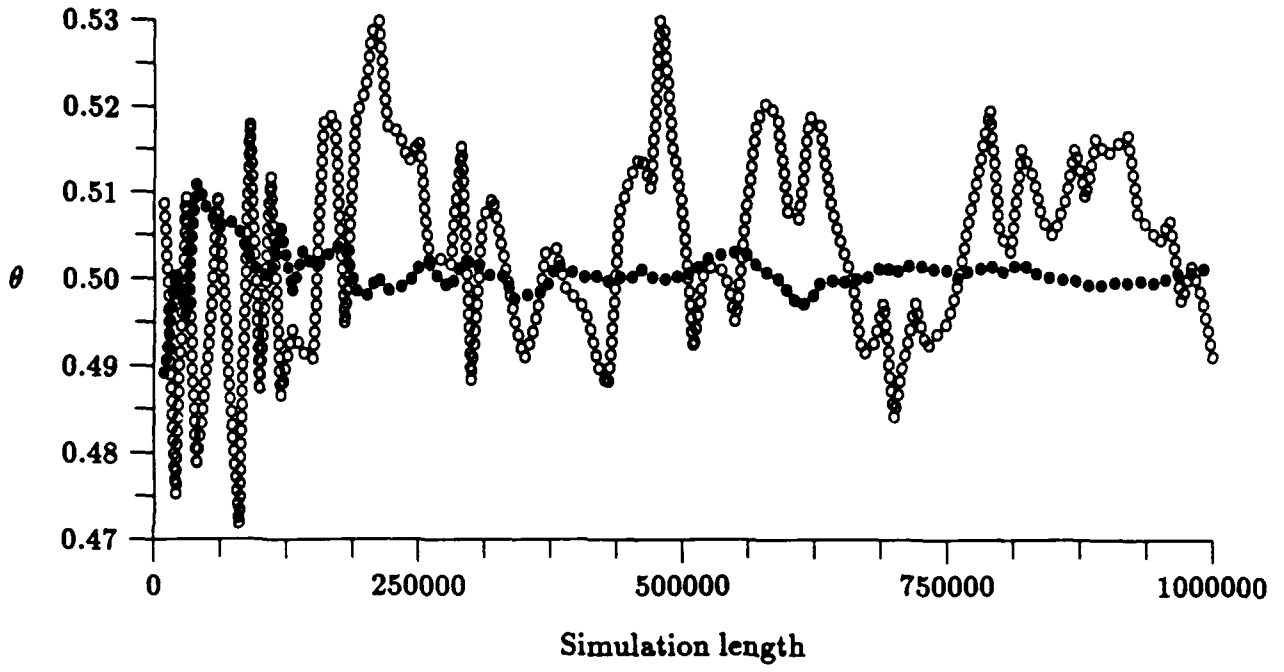

Figure 2: Convergence of KW (in white) and KWC (in black) with $T_n = n$.

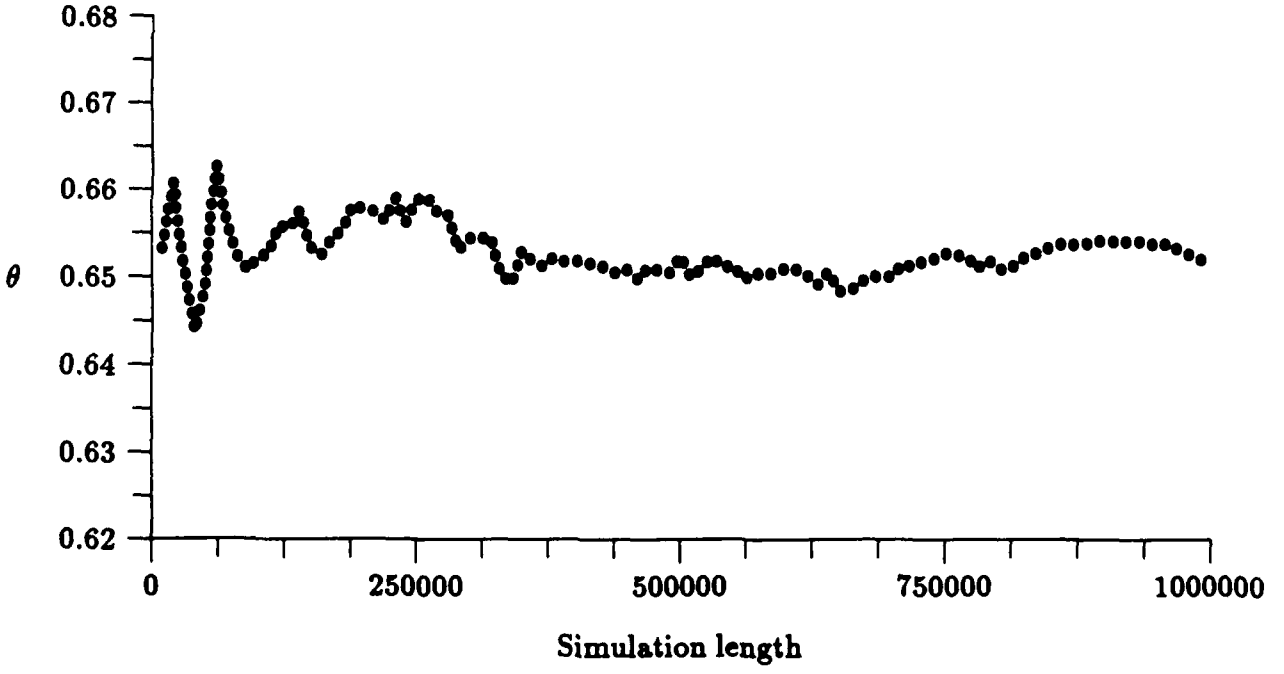

Figure 3: Convergence of KWC with $T_n = 5$.

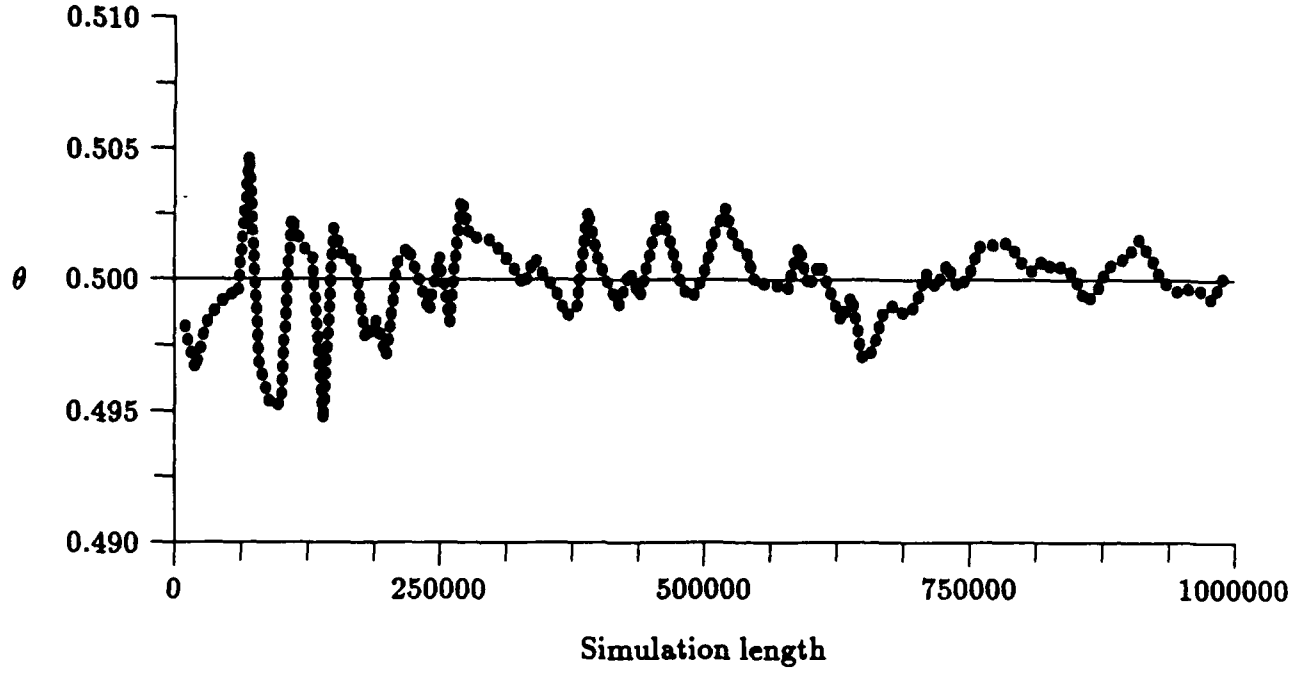

Figure 4: Convergence of PA with $T_n = 1$.

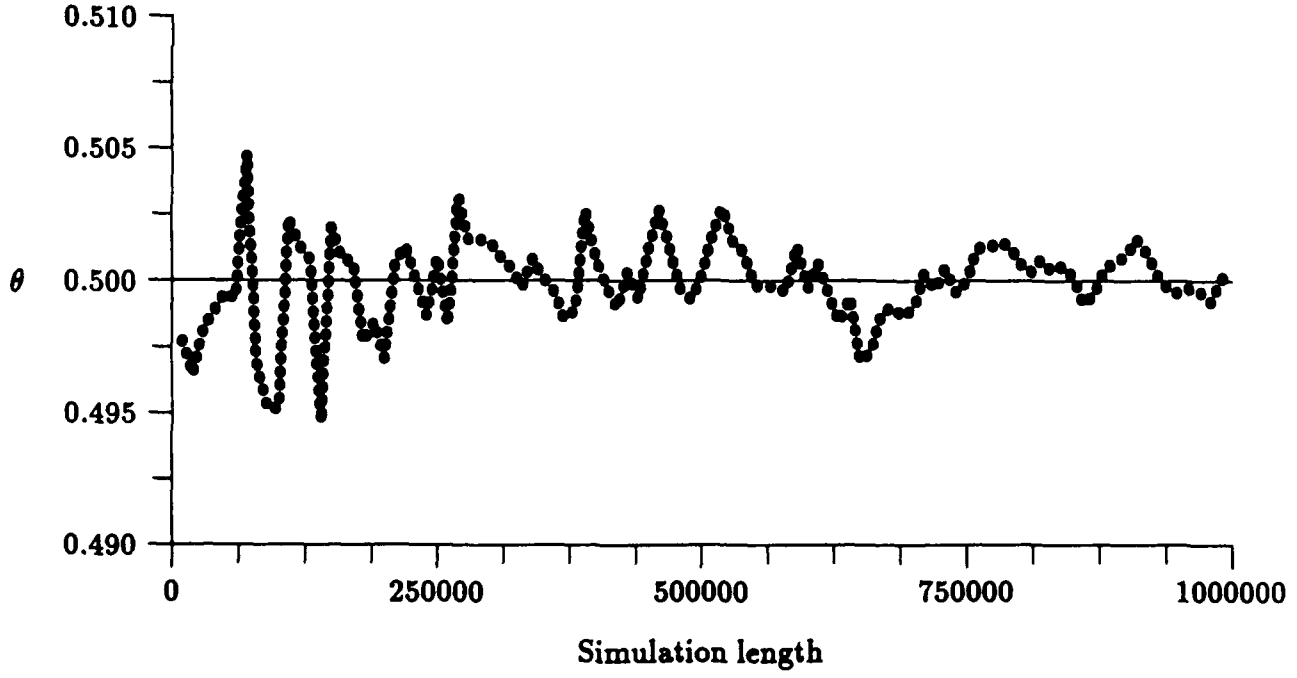

Figure 5: Convergence of PA with $T_n = 10$.

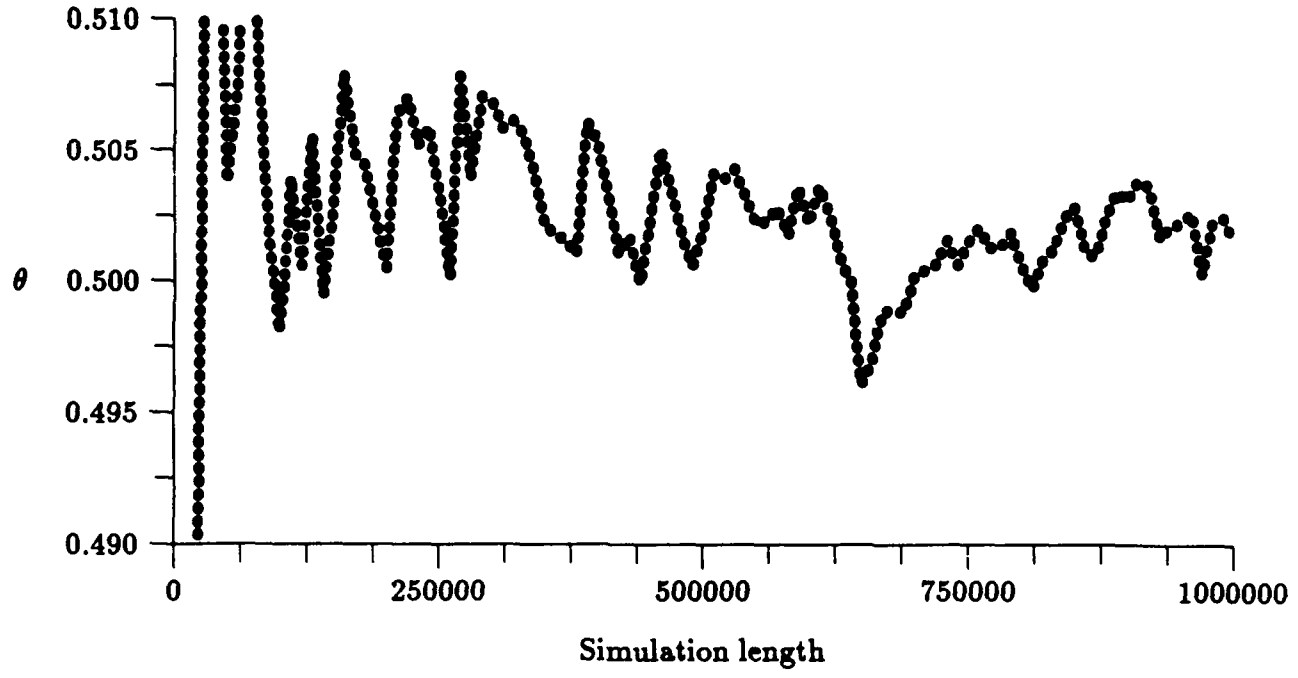

Figure 6: Convergence of LRR with $T_n = n$.

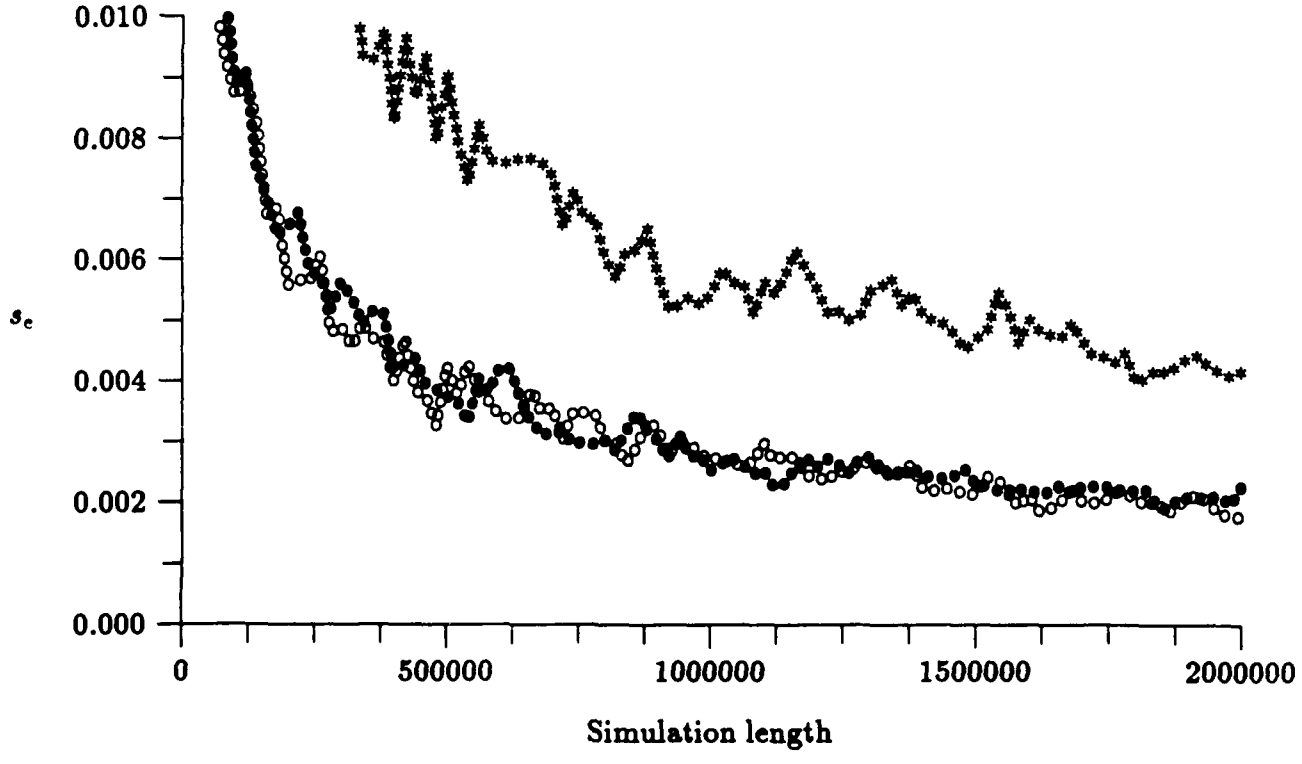

Figure 7: Evolution of s_e (N=100) for KWC (black), PA (white) and LRR (star).

	$C_1 = 1$ $(\theta^* = 1/2)$		$C_1 = 1/25$ $(\theta^* = 1/6)$		$C_1 = 25$ $(\theta^* = 5/6)$	
	s_d	s_e	s_d	s_e	s_d	s_e
KW, $T_n = n$	.03119	.03089				
KW, $T_n = 100 + n$	.03177	.03112				
KW(S), $T_n = n$	.02067	.02212				
KWC, $T_n = 5$	.00144*	.15255				
KWC, $T_n = 100$	.00332*	.00673				
KWC, $T_n = n$	.00216	.00242	.00027*	.00051	.02860	.02773
KWC, $T_n = 100 + n$	.00189	.00212	.00028*	.00055		
KWC(K), $T_n = n$	.00216	.00242				
KWC(K), $T_n = 100 + n$	.00189	.00212				
KWC, $T_n = n^{1/2}$	.00261*	.00731				
PA, $T_n = 1$	.00227	.00217				
PA, $T_n = 10$	.00227	.00216	.00053	.00051	.02402	.02575
PA, $T_n = 100$	.00229	.00219				
PA, $T_n = 100 + n$	.00203	.00192	.00045	.00043	.02685	.02848
PA(K), $T_n = 100 + n$	.00203	.00192				
PAR, $T_n = 5$	.00228*	.06175				
PAR, $T_n = n$	.00200	.00197	.00046	.00044	.02981	.03110
LR, $T_n = n^{1/3}$	.01221*	.02062				
LR, $T_n = n^{1/2}$	.03012	.02876	.02454	.02355	.04473	.05214
LR, $T_n = 10 + n^{1/2}$	.03300	.03165				
LR, $T_n = n^{2/3}$	.07494	.07115				
LRR, $T_n = n$	.00447	.00453	.00124	.00118	.07608	.07446
LRR, $T_n = 10 + n$	.00478	.00483				
LRR, $T_n = n^{1/2}$	.00443*	.01775				

Table 1: Some experimental results for $T = 10^6$, $N = 10$ and $C_1 = 1, 1/25, 25$. For the s_d values marked with an asterisk, the computed 95% confidence interval does not contain θ^*.

Figure 6 shows the evolution for LRR with $T_n = n$. The variation is larger here than for KWC and PA.

In figure 7, we can see the evolution of the standard error s_e with the simulation length, for three of the most interesting methods. These graphs are the result of a new experiment, with longer and much more numerous runs: we took $T = 2 \times 10^6$ and $N = 100$. They were also made by linear interpolation using 101 points. Again, we see that KWC and PA are roughly comparable, and that LRR has about twice their standard error (four times more variance). From these graphs, we can get an idea of the convergence rates: the standard error gets approximately cut in half when the simulation length is multiplied by four (i.e. the variance is approximately inversely proportional to the simulation length).

Simple (heuristic and approximate) statistical tests can also be made on the convergence rates. We can perform for instance another 200 independent runs of length $\tilde{T} = 500000$, compute the associated variance $\tilde{s}_d^2$, and compare it to the variance s_d^2 obtained in the above experiment, using the F statistic. Under the null hypothesis H_0: "the convergence rate is $t^{-1/2n}$ (where t denotes the simulation length), the variance for $\tilde{T}$ should be approximately four times the variance for T, and the statistic $F = 4s_d^2/\tilde{s}_d^2$ should follow (approximately) a F distribution with $(99, 199)$ degrees of freedom. We performed these tests for KWC with $T_n = n$, PA with $T_n = 10$ and LRR with $T_n = n$, and obtained the F values of 1.166, 0.0826 and 1.045 respectively. Since we know that the convergence rate cannot be faster than $t^{-1/2}$, we can test the null hypothesis against the alternative that it is t^{-x} for $x < 1/2$. In that case, H_0 is rejected at a 95% level when $F > 1.32$, which is the case for none of the three algorithms here. It should be pointed out, however, that despite the huge amount of simulation time they require, these tests are not very powerful. If we suppose for instance that the true convergence rate is $t^{-0.4}$, the probability of getting $F > 1.32$ is actually slightly smaller than 0.5. Also, the convergence rate is an asymptotic expression, and since we use finite values of T and $\tilde{T}$, we are only testing some approximation of it.

How does the speed of convergence of θ_n to θ^* compare to the speed of convergence of the cost estimator to the true average cost when θ is fixed ? We note that simply comparing the widths of the confidence intervals at the end doesn't make sense, since the parameter and the cost are not necessarily measured on the same scale. Dividing by the means to obtain relative values doesn't make sense either, there might be cases where θ^* or the average cost is zero or near zero. In any case, it is well known that the average cost estimator converges at rate $t^{-1/2}$, and we have observed the same convergence rate for θ_n. This means that we can estimate the optimum about as fast (in terms of rates),

as we can estimate the cost at a given point! We tried the following numerical example: with θ fixed at 0.5, we performed 100 independent simulation runs of length $T = 2 \times 10^6$ to estimate the average cost. The sample mean, standard deviation and mean square error of these 100 values were 3.00001, 0.00214 and 0.00213 respectively. Figure 8 illustrates the evolution of the standard error of the 100 values in terms of the simulation length.

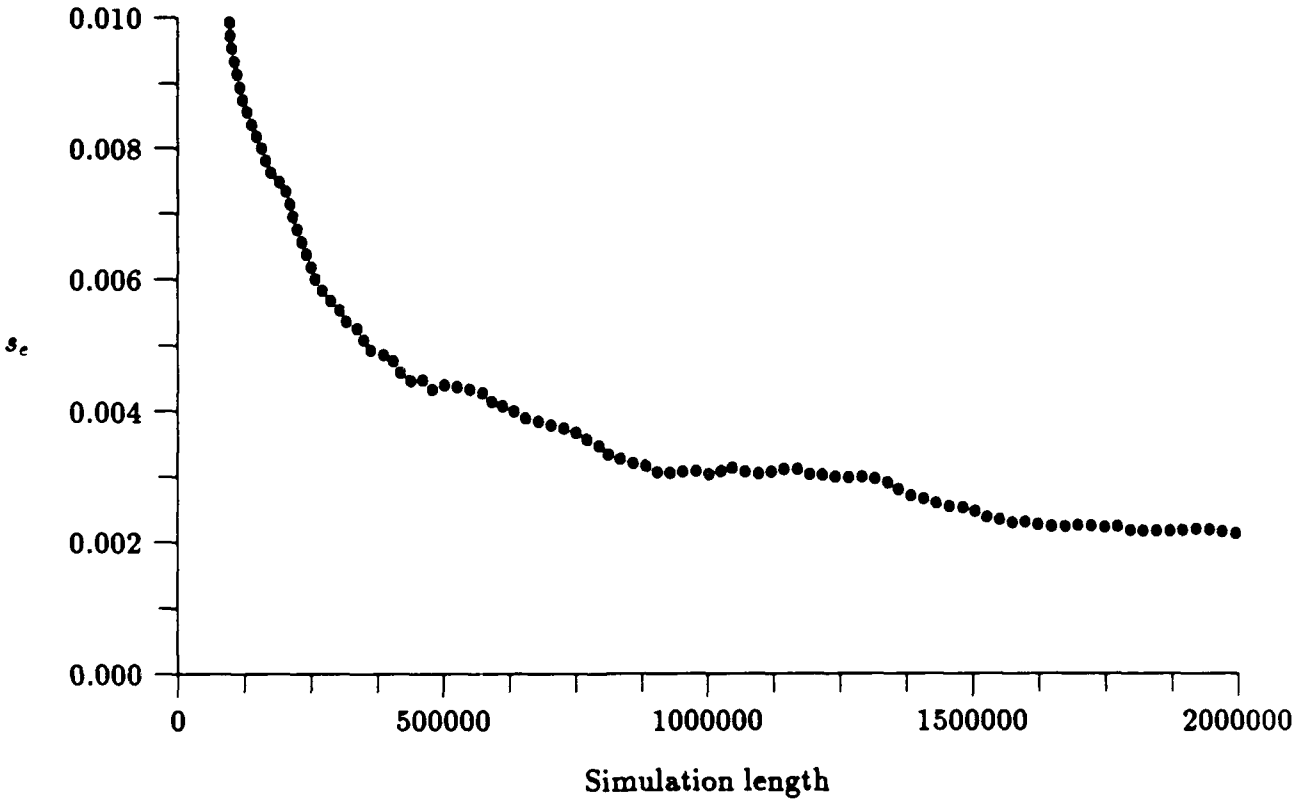

Figure 8: Evolution of the standard error for the average cost estimation with $N = 100$, $T = 2 \times 10^6$, $\theta = 0.5$ (actual cost: 3.0).

We made other sets of experiments with $C_1 = 1/25$ (for which $\theta^* = 1/6$) and $C_1 = 25$ (for which $\theta^* = 5/6$). The results appear in table 1. For $C_1 = 1/25$, the traffic intensity for θ near θ^* is low, and we get a much lower variance than for $C_1 = 1$. The opposite is true for $C_1 = 25$. The relative "rankings" of the algorithms are about the same. However, for KWC and $C_1 = 1/25$, the variance for θ_n goes down quickly and the bias does not go to zero fast enough to cope with that. The result is that for this experiment, the confidence interval I_θ does not contain θ^*. A possible remedy is to increase T_n faster. But in any case, this shows that one must be *very careful* about confidence intervals in these kinds of experiments, even if they are asymptotically valid. For $C_1 = 25$, LR is now better than LRR (θ is larger and the regenerative cycles are much longer in this case).

4.4. Proving some specific convergence properties

We will now look more closely at why for our example, for T_n constant, PA works fine while KW and KWC don't. To keep things simple, let's examine KWC with $T_n = 1$.

Lemma 1. *With $C_1 = 1$, $\Theta = [a, b]$ where $0 < a \leq b < 1$, $T_n = 1$, and a sequence $\{c_n, n \geq 0\}$ satisfying (9), KWC converges to b almost surely.*

PROOF. When we estimate the average cost using $T_n = 1$, we actually look at the time spent in the system by *one* customer, i.e. the customer being served in that subrun. This time can be expressed as $K_w + F_\theta^{-1}(u)$, where $F_\theta^{-1}(\cdot)$ is the inverse of the service time distribution (we assume that the inversion method is used to generate the service times), u is a $U(0,1)$ variate, and K_w is the (waiting) time already spent in the system by that customer. K_w is independent of what happens during that subrun, and $K_w = 0$ if the subrun began with an empty system. Hence, the average cost estimate at parameter level θ can be written as

$$h(\theta, u) = K_w + F_\theta^{-1}(u) + \frac{1}{\theta}. \tag{19}$$

and for KWC, we obtain

$$Y_n = \frac{h(\theta_n^+, u) - h(\theta_n^-, u)}{\theta_n^+ - \theta_n^-} = \frac{F_{\theta_n^+}^{-1}(u) - F_{\theta_n^-}^{-1}(u) + 1/\theta_n^+ - 1/\theta_n^-}{\theta_n^+ - \theta_n^-} \tag{20}$$

where $\theta_n^- = \pi_\Theta(\theta_n - c_n) = \max(a, \theta_n - c_n)$ and $\theta_n^+ = \pi_\Theta(\theta_n + c_n) = \min(b, \theta_n + c_n)$. Note that $\theta_n^+ - \theta_n^- \geq c_n$. For the exponential case, we have $E[F_\theta^{-1}(u) \mid \theta] = \theta$ and thus

$$E_n[Y_n] = \frac{\theta_n^+ - \theta_n^- + 1/\theta_n^+ - 1/\theta_n^-}{\theta_n^+ - \theta_n^-} = 1 + \frac{1}{\theta_n^+ \theta_n^-} \tag{21}$$

which converges to $1 - 1/\theta_n^2$ as c_n goes down to zero. Since $\theta_n \leq b < 1$, that expectation converges to a negative value everywhere on Θ. This means intuitively that θ_n will be "attracted" towards the upper bound b and we can prove it using Theorem 2. Let us redefine for the moment $\alpha(\cdot)$ such that

$$\nabla\alpha(\theta) = \frac{d\alpha(\theta)}{d\theta} = 1 - \frac{1}{\theta^2}.$$

In that case, in equation (2) we have $\beta_n = 1/\theta_n^2 - 1/\theta_n^+\theta_n^- \to 0$ as $n \to \infty$, and $E_n[\epsilon_n^2] \leq K_e/(\theta_n^+ - \theta_n^-)^2 \leq K_e/c_n^2$ for some constant K_e. Hence, S1, S2 and S4 are satisfied, with $\delta_n^2 = c_n^2/K_e$ in S4. Since $\theta = b$ is in this case the unique asymptotically stable point of (26), S5 is satisfied with $A = \Theta$, and Theorem 2 applies. ∎

For PA with constant T_n, we don't have $\beta_n \to 0$ as $n \to \infty$. Instead of using Theorem 2, we will use Theorem 4 and prove a weaker convergence result.

Lemma 2. *The algorithm of section 2 with PA, $\{\gamma_n, n \geq 0\}$ satisfying W3 of the appendix, and constant T_n converges in probability to the optimum θ_*.*

PROOF. We need to verify assumptions W1 to W7 and the result will follow from Theorem 4. Note that most complications arise because of the non-compactness of the state space (the queue length is unbounded). Let $T_n = J$, a constant. When using PA in this case, the only essential things that we must know at the beginning of each iteration are the number of customers in the system, and the elapsed time since the beginning of the current busy period (= 0 if system is empty). Let l_n and m_n be the values of these two quantities at the beginning of iteration n. To verify W1, we also put in s_{n+1} the values $\nu_{nj}, j = 1, \ldots, J$, where ν_{nj} is the elapsed time since the beginning of the current busy period when the j-th departure during iteration n occurs. Note that ν_n as defined in section 4.2 is the average of these ν_{nj}'s. For $n \geq 0$, let

$$s_{n+1} = (l_n, m_n, \nu_{n1}, \ldots, \nu_{nJ}) \in S = [0, \infty) \times \{0, 1, 2, \ldots\} \times [0, \infty)^J,$$

and let $s_0 = (0, \ldots, 0) \in S$. $\{(\theta_n, s_{n+1}), n \geq 0\}$ is obviously a Markov process, and the probability law of s_{n+1} given (θ_n, s_n) is independent of n and weakly continuous in (θ_n, s_n).

For any fixed $\theta \in [a, b]$, the system is stable and $\{s_n, n \geq 0\}$ is a Markov process with steady-state distribution $P^\theta(\cdot)$. Since the system is stable, for any $\epsilon > 0$, there is a constant $K_{\theta,\epsilon} > 0$ large enough so that $P^\theta([0, K_{\theta,\epsilon}]^{J+2}) \geq 1 - \epsilon$. Note that for any $n > 0$, s_n when θ is fixed is stochastically increasing in θ. Also, s_n is stochastically smaller when θ is allowed to move in $[a, b]$ than when it is fixed at $\theta = b$. The tightness properties required in W1 and W2 follow from these stochastic inequalities, the fact that $l_0 = m_0 = 0$, and the fact that the system is stable when $\theta \in [a, b]$.

W3 holds trivially. For W4, take $K_f = 1/a^2$, $\phi(s_{n+1}) = \nu_n$ and $\kappa = 1$. When θ is fixed at b, m_n has a steady-state distribution with bounded second moment (see [8]), and thus the same applies to ν_n. Because of the stochastic inequalities above, this also true when θ_n moves in $[a, b]$, so that $\sup_{n \geq 0} E_0(\nu_n^2) < \infty$.

W5 follows from W1 and the fact that Y_n is a continuous function of (θ_n, s_{n+1}).

For each $\epsilon > 0$, let $K_{b,\epsilon}$ be defined as above. For each $\bar{S}$ compact subset of S, there is a $n_{\bar{S}}$ such that for all $s_n \in \bar{S}$ and $i \geq n_{\bar{S}}$, $P^b(s_{n+i} \in [0, 2K_{b,\epsilon}]^{J+2} \mid s_n) \geq 1 - \epsilon$. Here, $n_{\bar{S}}$ can be viewed as a time that we give to the system to stabilize. Roughly, if $\bar{S}$ is "bigger", the initial state could be "bigger" (e.g. large initial queue size), and we will take a larger $n_{\bar{S}}$.

When θ_n is allowed to be smaller than b, this makes s_{n+i} stochastically smaller, so that this inequality still holds, and this implies W6.

From Suri and Zazanis [23, Theorem 2 and Corollary 1], the steady-state expectation of $Y_n = f(\theta_n, s_{n+1})$ when θ is *fixed* is equal to $\nabla\alpha(\theta) = 1/(1-\theta)^2 - C_1/\theta^2$. The equation $\dot{x} = \bar{\pi}(-\nabla\alpha(\theta))$ has a unique asymptotically stable point at $\theta^* = \sqrt{C_1}/(1+\sqrt{C_1})$ if this quantity is in Θ, and at the nearest boundary of Θ otherwise, so W7 is satisfied and this completes the proof. ∎

These two lemmas show that for this example, PA is *not* equivalent to "infinitesimal" KWC. Looking more closely, we can see that the difference lies in the way the methods are implemented. In PA (see [22, 23]), the perturbation "accumulator" (which holds in this case the elapsed length of the current busy period) is not reset to zero at the beginning of each iteration. Hence, the waiting times of all the customers, including the waiting that occured *before* the current iteration (if any), are perturbed appropriately. For KW or KWC, the "perturbed" parameter $(\theta_n \pm c_n)$ affects only the service times that are *generated* during its current use, and the waiting that occurs during these service times. Thus, the elapsed waiting times for the customers that are already in the queue at the beginning of a simulation subrun are influenced only by the parameter values for the previous subruns. Obviously, this introduces a bias on the estimated derivative.

5. Conclusion

Through a simple example, we have seen how a gradient estimation technique, such as finite differences with common random numbers (KWC), infinitesimal perturbation analysis (PA) or likelihood ratio (LR), can be incorporated into a stochastic approximation algorithm to get a provably convergent stochastic optimization method. We also pointed out some dangers associated with different kinds of bias.

For the example considered, PA gave the best results, but this may not be true in general. In fact, there are many examples for which some of the methods do not apply. For instance, consider the same M/M/1 queue as in section 4, but replace $w(\theta)$ in the objective function (13) by the proportion of customers that wait more than C_2 units of time, where C_2 is a positive constant. Here, PA does not work, because for a fixed finite segment of a realization ω, the derivative with respect to θ of the number of customers who wait more than C_2 is zero with probability one. As another example, keep the same objective function as in section 4, but take a different service time distribution: assume that the service time is 1 with probability θ and 2 with probability $1-\theta$, where $0 < \theta < 1$ and θ is the parameter. Again, PA doesn't apply (see [23]). Suppose now that the service time is 1 with probability p and $1+\theta$ with probability $1-p$, where p is fixed, $0 < p < 1$, and $\theta \geq 0$ is the parameter. In this case, LR (as described in section 3.4) doesn't apply since the set of possible realizations depends on θ. We don't rule out the possibility that eventual adaptations of the methods might work for these cases, but to our knowledge, this still has to be done. For many practical problems for which a threshold-type parameter has to be optimized, neither PA nor LR do apply, at least in their current forms. This certainly needs further research.

The performance of these algorithms when there are many parameters to optimize, and the incorporation of proper variance reduction techniques, are other interesting subjects for further investigation. We mentioned that PA can be viewed as a special case of LR with ω defined in a special way. In fact, depending on how we view ω in LR, we may obtain different methods to explore. In principle, PA and LR can be used to estimate higher order derivatives, but the variance is typically quite high. Is it too high to permit the implementation of good second order algorithms based on these estimates ? Again, further investigation is needed.

A. Appendix: sufficient convergence conditions

In this appendix, we give sufficient conditions for the convergence of (1) to an optimum. The first set of conditions imply almost sure convergence, the second set (usually easier to verify) imply weak convergence. These conditions are adaptations from [9, 11]. They are not given in their most general form, but are general enough for our purposes here.

In what follows, we will assume that Θ is a compact and convex set of the form $\Theta = \{\theta \in \mathbb{R}^d \mid g(\theta) \leq 0\}$, where $g(\cdot)$ is a κ_g-dimensional vector of continuously differentiable functions (constraints), as in [13, chap.10], and at any point on the boundary of Θ, the gradients of the active constraints are linearly independent. For any continuous function $v : \Theta \to \mathbb{R}^d$, one may view $v(\cdot)$ as the gradient of some objective function. Define the set of Kuhn-Tucker points associated with $v(\cdot)$ as

$$KT(v(\cdot)) = \left\{\theta \in \mathbb{R}^d \mid \exists \mu \in \mathbb{R}^{\kappa_g},\ \mu \geq 0,\ \text{such that } v(\theta) + \mu' \nabla g(\theta) = 0,\ \mu' g(\theta) = 0\right\}. \tag{22}$$

Let us introduce some notation, adapted from [9]. Define $t_n = \sum_{i=0}^{n} \gamma_i$ and $m(t) = \max\{n \mid t_n \leq t\}$ for $t \geq 0$. Let $x^0(\cdot)$ be the piecewise linear interpolation of the set of points $\{(t_n, \theta_n), n \geq 0\}$, and $x^n(\cdot)$ the left shift of $x^0(\cdot)$ defined by $x^n(t) = x^0(t + t_n)$, for $t \geq 0$. Hence, $x^n(0) = \theta_n$, and if $x^n(\cdot)$ converges to a limit $x(\cdot)$, the asymptotic properties of $x(t)$ as $t \to \infty$ can provide information on the asymptotic behavior of θ_n as $n \to \infty$. For any function $v : \Theta \to \mathbb{R}^d$, define

$$\bar{\pi}(v(\theta)) = \lim_{\delta \to 0+} \left(\frac{\pi_\Theta(\theta + \delta \cdot v(\theta)) - \theta}{\delta} \right). \tag{23}$$

Define the differential equation

$$\dot{x} = \bar{\pi}(-v(\theta)). \tag{24}$$

The set of asymptotically stable points of (24) is $KT(v(\cdot))$. The theorems below give conditions under which as $n \to \infty$, $x^n(\cdot)$ converges in some sense to a solution of (24) for a proper $v(\cdot)$. This convergence property is then used to analyse the behavior of $\{\theta_n, n \geq 0\}$. We now give a list of assumptions that will be used selectively in the next two theorems.

S1. For all $n \geq 0$, $\gamma_n \geq \gamma_{n+1} > 0$, and $\sum_{n=0}^{\infty} \gamma_n = \infty$.

S2. $\lim_{n \to \infty} \beta_n = 0$ almost surely.

S3. For each $T > 0$ and $\epsilon > 0$,

$$\lim_{n\to\infty} P\left(\sup_{j\geq n, t\leq T}\left\|\sum_{i=m(jT)}^{m(jT+t)-1}\gamma_i\epsilon_i\right\| \geq \epsilon\right) = 0. \tag{25}$$

S4. There is a positive sequence $\{\delta_n,\ n \geq 0\}$ such that $E_n[\epsilon_n'\epsilon_n] \leq 1/\delta_n^2$ and $\sum_{n=0}^{\infty}(\gamma_n/\delta_n)^2 < \infty$.

S5. There is a $\theta^* \in KT(\nabla\alpha(\cdot))$, an asymptotically stable point of

$$\dot{x} = \bar{\pi}(-\nabla\alpha(\theta)), \tag{26}$$

with domain of attraction $D_A(\theta^*)$ (in the sense of Liapounov), and almost surely, infinitely many θ_n belong to some compact $A \subseteq D_A(\theta^*)$.

Theorem 1. *(Kushner and Clark). (a) Assume S1 to S3. Then, almost surely, $x^0(\cdot)$ is uniformly continuous on $[0,\infty)$, and any limit $x(\cdot)$ of a convergent subsequence of $\{x^n(\cdot), n \geq 0\}$ satisfies (26). (b) If θ^* also satisfies S5, then $\lim_{n\to\infty}\theta_n = \theta^*$ almost surely.*

Theorem 1 is proved in [9, Theorem 5.3.1]. Condition S3 is quite general, but has low intuitive appeal, and is not always easy to verify. The theorem below uses a more restricted but more "familiar" condition. It is a variant of Proposition F in section II of [15].

Theorem 2. *Under the assumptions S1, S2, S4 and S5, the algorithm converges almost surely to θ^*.*

PROOF. It suffices to show that S3 holds, and the result will follow from part (b) of the previous Theorem. Note that under S4, the sequence $\{\sum_{i=0}^{n}\gamma_i\epsilon_i,\ n \geq 0\}$ is a martingale. For each $\epsilon > 0$, from Doob's inequality and from S4, we have, with probability one,

$$P\left(\sup_{j\geq n}\left\|\sum_{i=n}^{j}\gamma_i\epsilon_i\right\| \geq \epsilon\right) \leq \frac{K_d}{\epsilon}\sum_{i=n}^{\infty}\gamma_i^2 E_n[\epsilon_n'\epsilon_n] \leq \frac{K_d}{\epsilon}\sum_{i=n}^{\infty}(\gamma_i/\delta_i)^2. \tag{27}$$

for some constant K_d. This upper bound goes to zero as $n \to \infty$. Hence, we obtain condition $A2.2.4''$ of [9], which implies S4. ∎

Often, one must take $T_n \to \infty$ to get assumption S2. Sometimes, S2 is not satisfied, but $E_0[\beta_n] \to 0$ as $n \to \infty$, and the algorithm might also converge to the optimum. These

cases might be treated using the following (weaker) results. An example will be given in section 3. Theorem 3 is proved in [11], with slightly more general assumptions, while theorem 4 is an adaptation of the second part of Theorem 4.2.1 in [9], and can be proved in the same way (note that in the last paragraph of the proof of Theorem 4.2.1 in [9], the max should be replaced by a min). We give a new list of assumptions.

W1. Y_n is a deterministic function of (θ_n, s_{n+1}), which does not depend on n, say $Y_n = f(\theta_n, s_{n+1})$. Notice that s_{n+1} might include some of the random values generated during iteration n, or functions of them, even if it is useless or redundant information for the future evolution of the system. $\{(\theta_n, s_n), n \geq 0\}$ is a (possibly nonhomogeneous) Markov process, $\{s_n, n \geq 0\}$ is tight is a metric space S, and $P(s_{n+1} \in \cdot \mid \theta_n = \theta, s_n = s)$, defined on the Borel subsets of S, does not depend on n and is weakly continuous in (θ, s).

W2. For each fixed $\theta \in \Theta$, i.e. if $\gamma_n = 0$ and $\theta_n = \theta$ for all n, $\{s_n, n \geq 0\}$ is a Markov process with a unique invariant measure $P^\theta(\cdot)$. Also, $\{P^\theta(\cdot), \theta \in \Theta\}$ is tight.

W3. $\gamma_n > 0$ for all n, $\lim_{n\to\infty} \gamma_n = 0$, $\sum_{n=0}^{\infty} \gamma_n = \infty$ and $\sum_{n=0}^{\infty} |\gamma_{n+1} - \gamma_n| < \infty$.

W4. There are constants $K_f < \infty$ and $\kappa > 0$, and a positive valued function $\phi : S \to \mathbb{R}$, such that $\sup_{n\geq 0} E_0[|\phi(s_n)|^{1+\kappa}] < \infty$ and $|Y_n| \leq K_f(1 + \phi(s_{n+1}))$.

W5. $E[Y_n \mid \theta_n = \theta, s_n = s]$ is continuous in (θ, s).

W6. Either S is compact, or the $\{s_n, n \geq 0\}$ are mutually independent, or for each compact $\bar{S} \subset S$, there is an integer $n_{\bar{S}} < \infty$ such that for each $T > 0$, the set of probability measures $\{P((\theta_{n+j}, s_{n+j}) \in \cdot\ ,\ s_n \in \bar{S} \mid \theta_n = \theta, s_n = s),\ \theta \in \Theta,\ n \geq 0,\ j \geq n_{\bar{S}},\ t_{n+j} - t_n \leq T, \bar{S}$ compact subset of $S\}$ is tight.

W7. Assume W1 to W6 and use the notation defined there. Let $v : \Theta \to \mathbb{R}^d$ be the continuous (see W5) function defined by $v(\theta) = E^\theta[f(\theta, s)]$, where E^θ is the expectation that corresponds to the invariant measure P^θ defined in W2. There is a $\theta^* \in KT(v(\cdot))$, an asymptotically stable point of

$$\dot{x} = \bar{\pi}(-v(\theta)) = \bar{\pi}(-E^\theta[f(\theta, s)]), \tag{28}$$

with domain of attraction $D_A(\theta^*) = \Theta$.

Theorem 3. *(Kushner and Shwartz). Under assumptions W1 to W6,* $\{x^n(\cdot), n \geq 0\}$ *is tight and any weak limit of one of its subsequences satisfies the projected differential equation (28) almost everywhere with probability one.*

Theorem 4. *Under assumption W7,* θ_n *converges to* θ^* *in probability, i.e. for each* $\epsilon > 0$,

$$\lim_{n \to \infty} P(\|\theta_n - \theta^*\| \geq \epsilon) = 0. \tag{29}$$

Acknowledgements

This work has been supported by NSERC-Canada grant # A5463 and FCAR-Québec grant # EQ2831 to the first author. Part of the work was done while the first author was visiting the Operations Research Department at Stanford University. Abderrahim Loukili and Philippe Locong made some preliminary numerical experiments which were helpful. We also wish to thank Y. T. Leung and R. Suri for helpful comments.

References

[1] Azadivar, F. and Talavage, J. "Optimization of Stochastic Simulation Models", *Math. and Computers in Simulation*, **XXII** (1980), 231–241.

[2] Glynn, P. W. "Optimization of Stochastic Systems", *Proceedings of WSC'86*, The Society for Computer Simulation (1986), 52–58.

[3] Glynn, P. W. "Stochastic Approximation for Monte-Carlo Optimization", *Proceedings of WSC'86*, The Society for Computer Simulation (1986), 356–364.

[4] Glynn, P. W. "Likelihood Ratio Gradient Estimation: an Overview", *Proceedings of WSC'87*, The Society for Computer Simulation (1987), 366–375.

[5] Gong, W. B. and Ho, Y. C. "Smoothed (Conditional) Perturbation Analysis of Discrete Event Dynamical Systems", *IEEE Trans. on Automatic Control*, **AC-32**, 10 (1987), 858–866.

[6] Heidelberger, P., Cao, X.-R., Zazanis, M. A. and Suri, R., "Convergence Properties of Infinitesimal Perturbation Analysis Estimates", to appear in *Management Science* (1989).

[7] Kesten, H., "Accelerated Stochastic Approximation", *Ann. Math. Stat.*, **23** (1952), 463–467.

[8] Kleinrock, L., *Queueing Systems*, Vol. 1, Wiley, New York, 1975.

[9] Kushner, H. J. and Clarke, D. S., *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, Applied Math. Sciences, vol. 26, 1978.

[10] Kushner, H. J., *Approximation and Weak Convergence Methods for Random Processes*, The MIT Press, Cambridge, Mass., 1984.

[11] Kushner, H. J. and Shwartz, A., "An Invariant Measure Approach to the Convergence of Stochastic Approximations with State Dependent Noise", *SIAM J. on Control and Optim.*, **22**, 1 (1984), 13–24.

[12] L'Ecuyer, P., "SIMOD: Définition fonctionnelle et guide d'utilisation (version 2.0)", Report no. DIUL-RT-8804, Département d'informatique, Univ. Laval (1988).

[13] Luenberger, D. G., *Linear and Nonlinear Programming*, Addison-Wesley, second ed., 1984.

[14] Meketon, M. S., "Optimization in Simulation: a Survey of Recent Results", *Proceedings of WSC'87*, The Society for Computer Simulation (1987), 58–67.

[15] Métivier, M. and Priouret, P., "Application of a Kushner and Clark Lemma to General Classes of Stochastic Algorithms", *IEEE Trans. on Information Theory*, **IT-30**, 2 (1984), 140–151.

[16] Métivier, M. and Priouret, P., "Théorèmes de convergence presque sûre pour une classe d'algorithmes stochastiques à pas décroissant", *Prob. Theory and Related Fields*, **74** (1987), 403–428.

[17] Reiman, M. I. and Weiss, A., "Sensitivity Analysis for Simulation via Likelihood Ratios", to appear in *Operations Research* (1989).

[18] Rubinstein, R. Y., *Monte-Carlo Optimization, Simulation and Sensitivity of Queueing Networks*, Wiley, 1986.

[19] Rubinstein, R. Y., "The Score Function Approach for Sensitivity Analysis of Computer Simulation Models", *Math. and Computers in Simulation*, **28** (1986), 351–379.

[20] Rubinstein, R. Y., "Sensitivity Analysis of Computer Simulation Models via the Efficient Score", to appear in *Operations Research* (1989).

[21] Suri, R., "Infinitesimal Perturbation Analysis of General Discrete Event Dynamic Systems", *J. of the ACM*, **34**, 3 (1987), 686–717.

[22] Suri, R. and Leung, Y. T., "Single Run Optimization of Discrete Event Simulations: An Empirical Study Using the M/M/1 Queue", Technical Report No. 87-3, Dept. of Indust. Eng., University of Wisconsin—Madison (1987).

[23] Suri, R. and Zazanis, M., "Perturbation Analysis Gives Strongly Consistent Estimates for the M/G/1 Queue", *Management Science*, **34**, 1 (1988), 39–64.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

MASTER COPY - FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): Technical Report No. 31	5 MONITORING ORGANIZATION REPORT NUMBER(S): ARO 25839.4-MA
6a NAME OF PERFORMING ORGANIZATION: Dept. of Operations Research; 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION: U. S. Army Research Office
6c ADDRESS (City, State, and ZIP Code): Stanford, CA 94305-4022	7b ADDRESS (City, State, and ZIP Code): P. O. Box 12211, Research Triangle Park, NC 27709-2211
8a NAME OF FUNDING / SPONSORING ORGANIZATION: U. S. Army Research Office; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: DAAL03-88-K-0063
8c ADDRESS (City, State, and ZIP Code): P. O. Box 12211, Research Triangle Park, NC 27709-2211	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Stochastic Optimization by Simulation:
Some Experiments with a Simple Steady-State Queue

12 PERSONAL AUTHOR(S)
Pierre L'Ecuyer, Nataly Giroux, and Peter W. Glynn

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Technical	FROM _____ TO _____	February 1989	32

16 SUPPLEMENTARY NOTATION
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
simulation, stochastic approximation, likelihood ratios, perturbation analysis.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

(Please see other side)

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code) / 22c OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ABSTRACT

New approaches like perturbation analysis and the likelihood ratio method have been proposed recently to estimate the gradient of a performance measure with respect to some continuous parameters in a dynamic stochastic system. In this paper, we experiment the use of these estimators in stochastic approximation algorithms, to perform so-called "single-run optimizations". We also compare them to finite difference estimators, with and without common random numbers. The experiments are done on a simple M/M/1 queue. The performance measure involves the average system time per customer, and the optimal solution is easy to compute analytically, which facilitates the evaluation of the algorithms. We also demonstrate some properties of the algorithms. In particular, we show that using perturbation analysis, the single-run optimization converges to the optimum even with a fixed (and small) number of ends of service per iteration, while under the same conditions, the algorithm that uses the finite difference estimators converges to the wrong answer.