Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. James C. Spall

ISBN: 0-471-33052-3

CHAPTER 14

SIMULATION-BASED OPTIMIZATION I: REGENERATION, COMMON RANDOM NUMBERS, AND SELECTION METHODS

This and the next chapter consider the important case where Monte Carlo simulations are the primary source of input information during the search process. Search and optimization play a major role in two aspects of simulation analysis—building the simulation (parameter estimation) and simulation-based optimization (using the simulation as a proxy for the actual system in an optimization process). The focus here is on the latter problem of simulation-based optimization. In particular, we discuss some of the ways in which special properties associated with simulations—properties not generally available in other settings—can be used to enhance search and optimization. The resulting algorithms are generally special cases of some of the algorithms seen previously.

Section 14.1 provides general background. Section 14.2 introduces the important concept of regeneration, whereby a system periodically resets itself. Section 14.3 is a short summary of finite-difference-type methods for simulation-based optimization with emphasis on the standard finite-difference and simultaneous perturbation estimators that were encountered in general form in Chapters 6 and 7. Based on these gradient estimators for stochastic approximation, Section 14.4 discusses the important concept of common random numbers, one of the most useful properties of simulations and one that is not typically available in physical systems. Section 14.5 considers statistical methods for selecting the best option among several candidate solutions, including techniques based on common random numbers. This section connects to some of the discrete optimization methods of Chapter 12. Section 14.6 offers some concluding remarks, including a brief discussion of the limitations of simulation-based optimization.

14.1 BACKGROUND

14.1.1 Focus of Chapter and Roles of Search and Optimization in Simulation

Many real-world problems are too complex to be solved by analytical means and are therefore studied via computer simulation. Although simulation has

traditionally been viewed as a method of last resort, recent advances in hardware, software, and user interfaces have made simulation more of a first-line means of attacking many problems. The focus in this and the next chapter will be on *stochastic simulations*, sometimes called *Monte Carlo simulations*. These rely on the internal generation of pseudorandom numbers (Appendix D) to represent the randomness in relevant events (although some of the methods apply as well in simulations that are purely deterministic). As we will see, stochastic simulations are intimately connected to the problem of *noisy* measurements for search and optimization as posed in Chapter 1 (Property A in Subsection 1.1.3) and seen repeatedly in other chapters of this book.

This and the next chapter discuss ways in which some of the unique properties associated with simulations can be used to enhance search and optimization. The specific algorithms here are generally special cases of some of the algorithms seen previously. This chapter focuses on methods where the simulation output is used *directly* in the optimization process. This is in contrast to stochastic gradient-based methods, which require extensive knowledge of the inner workings of the simulation. This chapter considers stochastic approximation algorithms with the finite-difference or simultaneous perturbation gradient estimation methods of Chapters 6 and 7 and statistical comparison methods similar to the material in Chapter 12. The exception to this focus on direct optimization is a brief discussion on gradient-based methods in the regeneration discussion (Section 14.2).

In contrast, in Chapter 15, knowledge of the internal structure of the simulation is used to construct sophisticated gradient estimators for use in optimization algorithms of the stochastic gradient form (Chapter 5) or the deterministic nonlinear programming form (via the sample path method). One special case of the results in Chapter 15 (the infinitesimal perturbation method) was summarized in Section 5.3 as an illustration of the stochastic gradient algorithm.

Some of the discussion in this and the next chapter pertains to queuing systems. A standard notation for such systems is *GI*/*G*/*c*, where *GI* stands for the distribution of the times between arrivals into the queuing network, *G* stands for the distribution of the service times to process an arrival, and *c* stands for the number of servers. The notation "*GI*" here means that the interarrivals have a general (*G*) distribution and are independent (*I*). An important special case is the *M*/*M*/1 queue, where *M* represents an exponential distribution for the interarrival and service time distributions and there is one server in the network. (The notation "*M*" comes from the fact that with an exponential distribution, the probability of a future occurrence—arrival or service completion, as appropriate—is independent of how long it has been since the last arrival or since the service began. This is the ***M***arkovian or ***M***emoryless property of the exponential distribution.)

A cautionary note: Serious, "industrial strength" uses of simulation in search and optimization (or other applications) generally involve many implementation details and an exploitation of structure unique to the particular application. It is obviously not possible in the two chapters here to discuss many

of the clever strategies that are exploited in practice. For example, a large amount of the simulation-based literature is focused on *discrete-event* systems. Such systems are characterized by jumps in the state of a system as a result of the occurrence of an event (e.g., an arrival into a queuing network). A large number of modern systems in communications, manufacturing, transportation, healthcare, and other areas are discrete-event systems. Properties associated with discrete events can be heavily exploited in many serious simulation-based optimization problems to enhance the performance of the relevant algorithms.

The focus of this and the next chapter is on some important generic principles rather than the many specialized strategies associated with discrete-event or other systems. A user with a serious application is directed to the vast specialized literature in simulation, including the texts of Glasserman (1991a), Pflug (1996), Fu and Hu (1997), Banks (1998), Rubinstein and Melamed (1998), Cassandras and Lafortune (1999), and Law and Kelton (2000). A survey with accompanying discussion is given in Fu (2002).

Search and optimization algorithms play a critical role in at least two aspects of simulation analysis. One is in *building* the simulation model through the determination of the optimal values of the parameters internal to the simulation. The other role is in *using the simulation* to optimize the *real system* of interest after the simulation has been built (i.e., after the internal model parameters are determined). In the first role, θ in the notation of this book represents the fundamental internal simulation model parameters that are assumed to hold across a variety of scenarios. In the second role for optimization involving simulations, θ has a different meaning, representing parameters that may be varied in the real system and in running the simulation (i.e., θ is associated with the simulation input).

For example, in a simulation of vehicle traffic in a network, the parameters in the first role (model building) might represent fundamental characteristics associated with the network structure (e.g., the mean arrival rates into the network and/or the topology of the network as reflected by the distances between intersections). These correspond to fixed quantities that the analyst cannot generally alter in the real system. Parameters in the second role (system optimization via simulation) might, for example, be the settings for the traffic signals in the real network. So, one might run the simulation for a variety of traffic signal settings, observing which setting produces the best performance for the *fixed* network characteristics represented by the simulation parameters found in the model-building phase. The critical process of building simulations is a large area that is closely tied to some of issues seen elsewhere in this book, especially parameter estimation for models (a.k.a. system identification) (Chapter 3; Sections 4.2, 5.1, and 5.2); the bias-variance tradeoff, model selection, and the information matrix (Chapter 13); and experimental design (Chapter 17).

Because many of the methods discussed elsewhere in this book are directly applicable to the process of building the simulation model, this and the next chapter focus on the *other* (second) role for simulation optimization, as discussed in the preceding paragraphs. So, θ typically represents some system design or control parameters that are being determined with the aid of the

simulation. It is assumed that the simulation has been properly constructed using the appropriate combination of prior knowledge of the process and estimation of the fundamental model parameters.

The use of simulations for optimizing real systems is sometimes controversial. The controversy usually centers on the question of whether the simulation is an adequate representation of the real process. We will not dwell on this debate. The answer to such a question centers on the specific goals of the analysis relative to the inherent strengths and weaknesses of the simulation software, the validity of the mathematical conditions embedded in the software (e.g., are the true system arrivals really Poisson distributed?), and the care with which the simulation is exercised by the users of the software. For motivation, we will refer several times to the following example in the sections below.

Example 14.1—Plant layout. Suppose that an industrial engineer is interested in finding the optimal placement of machines on a plant (factory) floor to maximize the product output in an assembly operation. Important constraints in this process include the types of material being assembled and the order of assembly, the kinds of labor skills that can reasonably be employed, workplace safety, product life, and so on. An obvious—but very costly—means for solving this problem is to simply try different machine placements in the real plant and conduct operational tests. This, clearly, is almost never feasible in practice.

Suppose, on the other hand, that there is a credible simulation of the plant output as a function of machine placement. This simulation must take into account the overall design of the plant, the movement of material, the role of people on the plant floor, and the other constraints mentioned above, all weighted by relative importance. (The parameters determined during the model-building phase would encompass these fixed aspects of the system, corresponding to the first role of search and optimization in simulation, as discussed above.) One can try different machine placements in the *simulation* of the system and evaluate the performance at each configuration. Here, $\boldsymbol{\theta}$ represents the vector of placement coordinates for the machines. This simulation-based approach may be used to provide the optimal solution if the simulation is a faithful replication of the actual system. A key issue here is the choice of intermediate test machine placements in the simulation for evaluation purposes en route to obtaining the best placement. Optimization methods provide the mechanism for determining these test machine placements via the rules that govern the iteration-by-iteration selection of $\boldsymbol{\theta}$ (machine location) values. ❑

14.1.2 Statement of the Optimization Problem

In simulation-based optimization, we are faced with solving a version of the problem introduced in Chapter 5, namely to minimize the loss function:

$$L(\boldsymbol{\theta}) = E[Q(\boldsymbol{\theta}, \boldsymbol{V})] \tag{14.1}$$

over $\boldsymbol{\theta} \in \Theta$, where $\boldsymbol{V}$ represents the amalgamation of the (pseudo) random effects in the simulation (presumably manifesting themselves in a way indicative of the random effects in the actual system) and Q represents a *sample* realization of the loss function calculated from running the simulation. In general, the solution to $\min_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta})$ is not unique. For ease of discussion, however, we will follow precedent and generally refer to *the* solution $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta})$. Most of the results here follow directly when there is a *set* of solutions Θ^*.

In the plant layout example above, Q might represent the (negative) output for one day given a particular set of random effects $\boldsymbol{V}$ (individual machine performance, labor productivity, material quality, arrival rates for needed parts from external suppliers or other areas of the plant, etc.), while $\boldsymbol{\theta}$ represents the vector of location coordinates for the machines being positioned. Hence one simulation run produces one value of Q. Note that the direct simulation output may or may not represent a value of Q (which is assumed to be a scalar number). For example, in the plant layout problem again, if the simulation output were a stream of hour-by-hour production rates broken down by different product categories, then a transformation to cumulative daily output weighted by the relative importance of different categories may be needed to obtain Q.

To focus the discussion in this and the next chapter, we suppose—unless noted otherwise—that $\boldsymbol{V}$ represents the collection of *direct* random processes in the simulation rather than the collection of underlying uniformly distributed $U(0, 1)$ (pseudo) random variables that may be generating the processes. Hence, for example, in an $M/M/c$ queuing problem, $\boldsymbol{V}$ represents the Poisson-distributed arrivals and exponential-distributed service times versus the underlying uniform variables that are being used (say, with the inverse transform method—Appendix D) to generate the relevant Poisson and exponential distributed random variables. We denote the ith element of $\boldsymbol{V}$ by $\boldsymbol{\mathcal{V}}_i$.

Although $Q = Q(\boldsymbol{\theta}, \boldsymbol{V})$ is shown in (14.1) as an explicit function of both $\boldsymbol{\theta}$ and $\boldsymbol{V}$, there are many cases where the dependence on $\boldsymbol{\theta}$ is only implicit via the effect of $\boldsymbol{\theta}$ on the probability distribution used in generating $\boldsymbol{V}$. For example, in a queuing system where $\boldsymbol{\mathcal{V}}_i$ represents the processing time of customer i, there may be interest in minimizing the total processing time of all customers. Here $Q = Q(\boldsymbol{\theta}, \boldsymbol{V}) = \sum_i \boldsymbol{\mathcal{V}}_i$. The dependence on $\boldsymbol{\theta}$ is implicit via its effect on the probability distribution of the processing times. Suppose, in particular, that the distribution of $\boldsymbol{V}$ is governed by a probability density function $p_{\boldsymbol{V}}(\cdot|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents parameters of the density that can be controlled by the system designer. The ultimate dependence on $\boldsymbol{\theta}$ (although not explicit in Q) appears in the loss function, as shown by writing $L(\boldsymbol{\theta}) = E(\sum_i \boldsymbol{\mathcal{V}}_i)$ where the expectation is calculated based on $p_{\boldsymbol{V}}(\cdot|\boldsymbol{\theta})$ over the domain for $\boldsymbol{V}$.

The goal, therefore, is to find the $\boldsymbol{\theta}$ producing the best *mean* value of Q by using runs of the available simulation. Subject to the caveats above on the validity of the simulation as a representation of the real system, this is then tantamount to finding the best value of $\boldsymbol{\theta}$ for the *real* system. The remainder of this chapter discusses several approaches to simulation-based optimization.

14.2 REGENERATIVE SYSTEMS

14.2.1 Background and Definition of the Loss Function $L(\theta)$

Consider the common case where the system under study is a dynamic (time-varying) process. One of the significant issues in simulation-based optimization is the choice of how much time should be represented in the runs of the simulation. That is, in the process of simulation-based optimization, one chooses θ values and runs simulations to produce $Q(\theta, V)$ values (or some quantity related to $Q(\theta, V)$ such as its derivative with respect to θ). In many cases, however, the amount of time to be *represented* by the simulation (not to be confused with the CPU or clock time involved in actually executing the simulation) may not be obvious. For example, in the plant layout problem above, an analyst might wonder if the loss function—and hence simulation-based loss *measurement*—should represent one hour of plant operations? One day? One year? The answer to this question clearly depends on the goals of the analysis. For example, if one is concerned with optimizing an inventory control system in a plant or retail establishment, it would be useful to run the simulation over *at least* one complete cycle of initial stocking, drawing the inventory down, and restocking.

A common means of answering the question above is by appealing to the concept of *regeneration*. A regenerative system has the property of returning periodically to some particular probabilistic state from which it effectively starts anew. That is, the system returns to conditions under which the probabilities of various future outcomes are the same as the a priori probabilities of achieving those outcomes at earlier incarnations of that state. An equivalent way of viewing regenerative processes is that at the particular regeneration times, the future behavior of the system is independent of the past behavior.

Some of the most common examples of regenerative processes are queuing networks where the system periodically "starts over." For example, daily traffic flow in a road network may be *probabilistically* approximately identical day-to-day over the course of the Monday–Friday workweek. Note that the regeneration times may be random for some systems and deterministic (i.e., knowable in advance) for others. The traffic flow example is a candidate for either of these settings. For example, key traffic indicators may reach a defined free-flow state immediately prior to the buildup of traffic congestion in the morning rush period at *randomly* varying times in the morning as a function of ambient traffic, weather conditions, accidents, and so on. In this setting the length of the regeneration period will vary about a mean of 24 hours. On the other hand, for some regions with very predictable conditions, the random effects may be negligible, indicating that the regeneration time associated with the defined pre-rush traffic conditions is known with nearly perfect certainty. Another common example of a regenerative process is an (s, S) inventory control system. When the inventory drops below s, an order is triggered to bring the inventory to level S, where the system regenerates (under the appropriate conditions on demand, backorders, etc.).

In general, the regeneration points are a set of deterministic or random times $\tau_0 < \tau_1 < \tau_2 < \ldots$ such that the system probabilistically restarts itself at each τ_i. Under modest conditions, the regenerative periods (or cycles) will be independent, identically distributed (i.i.d.) stochastic processes. If the regeneration points are also random, the periods have i.i.d. random lengths, $\tau_1 - \tau_0, \tau_2 - \tau_1, \ldots$. The example below illustrates this point.

Example 14.2—Queuing system. Consider a single-server queuing system *GI/G*/1 with a general (unknown) distribution for the interarrival and service times. Let W_k and S_k be the waiting and service time, respectively, for the kth customer into the system. Let $A_{k|k+1}$ be the interarrival time between the kth and $(k + 1)$st customer. Suppose that the system starts with zero wait ($W_0 = 0$) (so the first customer always encounters zero wait). Assume that $\{S_k\}$ and $\{A_{k|k+1}\}$ are i.i.d. sequences with means μ_S and μ_A, respectively. By this i.i.d. assumption, a customer arriving later in the process who encounters a zero wait ($W_k = 0$) will face the same probabilistic conditions as an earlier customer. If $\mu_S < \mu_A$, it is known that W_k will, in fact, return to the state of zero wait for an infinite subset of indices within $k \in [0, 1, 2, \ldots)$ (Rubinstein, 1981, p. 193). Hence this process is regenerative based on a return state of zero wait.

Figure 14.1 shows a plot of the waiting time encountered by 17 customers into a queuing system. We see that customers 1, 3, 4, 7, 11, and 16 face zero wait. Let us assume that the first regeneration period begins with the arrival of the first customer. So, the first regeneration period represents the time between the arrival of the first customer and the arrival of the next customer with zero wait (customer 3 here). This system is one where regeneration times are random.

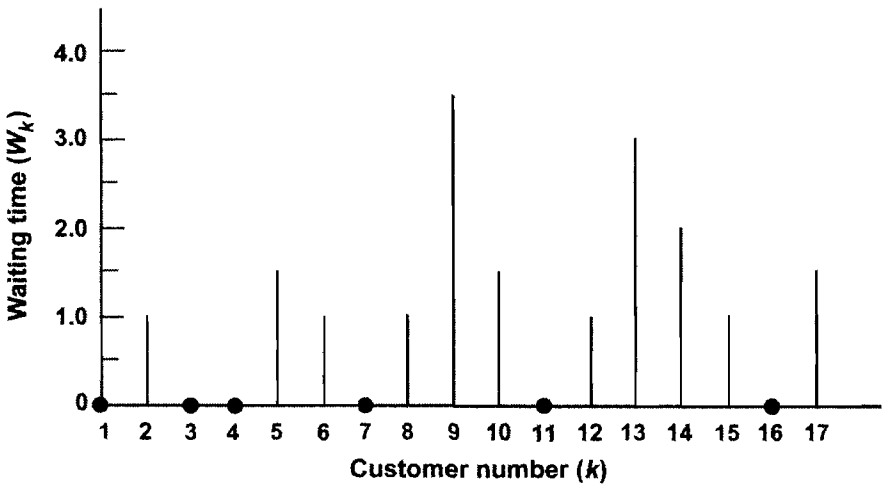

Figure 14.1. Zero and nonzero wait times depicting regeneration for *GI/G*/1 queue. Beginning with customer 1, the system regenerates at customers 3, 4, 7, 11, and 16.

Figure 14.1 depicts five complete regeneration periods, encompassing the system being turned on (say, time 0) and customers 1–15. The five periods correspond to the intervals with customers 1–2, 3, 4–6, 7–10, and 11–15. Also shown is the beginning of a sixth period with customer 16 (with the period having an unknown end). The length of the five random regeneration periods are $A_{1|2} + A_{2|3}$, $A_{3|4}$, $A_{4|5} + A_{5|6} + A_{6|7}$, $A_{7|8} + A_{8|9} + A_{9|10} + A_{10|11}$, and $A_{11|12} + A_{12|13} + A_{13|14} + A_{14|15} + A_{15|16}$. ❑

Simulation-based optimization when the system has a regenerative structure has been considered extensively (e.g., Rubinstein, 1981, Sects. 6.5 and 6.6; Fu, 1990; L'Ecuyer and Glynn, 1994; Fu and Hill, 1997; Rubinstein and Melamed, 1998, Sect. 3.7; Tang et al., 1999). Regenerative systems provide a logical basis for determining how long to run a simulation at each iteration of the optimization algorithm. In particular, one can consider the optimization of system performance using sets of data that completely cover one or more regeneration periods. This has the advantage of providing groups of data that are i.i.d., greatly easing analytical analysis of the system, such as required in some optimization approaches. This contrasts with nonregenerative simulations, where one may have no formal basis for determining how long to run a simulation to provide a representative sample of the process.

A typical loss function used in concert with regeneration is

$$L(\boldsymbol{\theta}) = \frac{E(C_{\boldsymbol{\theta}})}{E(\ell_{\boldsymbol{\theta}})}, \tag{14.2}$$

where $C_{\boldsymbol{\theta}}$ represents some random cost accrued during a regeneration period of length $\ell_{\boldsymbol{\theta}}$ with $E(\ell_{\boldsymbol{\theta}}) > 0$. In the $L(\boldsymbol{\theta}) = E[Q(\boldsymbol{\theta}, \boldsymbol{V})]$ notation in Section 14.1 and elsewhere, $Q(\boldsymbol{\theta}, \boldsymbol{V}) = C_{\boldsymbol{\theta}}/E(\ell_{\boldsymbol{\theta}})$ (so $\boldsymbol{V}$ represents the random terms entering $C_{\boldsymbol{\theta}}$). Below, we sometimes add an index i to $C_{\boldsymbol{\theta}}$ and $\ell_{\boldsymbol{\theta}}$ to indicate the cost and length associated with a specific regeneration period—the ith. When no index is used, as in (14.2), the reference is to the generic random variable for an arbitrary regeneration period.

In (14.2), "length" $\ell_{\boldsymbol{\theta}}$ typically represents elapsed time or number of objects or individuals that are processed during a period. So, the loss function is the expected cost accrued during a regeneration period relative to the expected length of the period. Because the regeneration periods are i.i.d., there is no index on L associated with a particular period; all periods produce the same mean values in the numerator and denominator. While the example above is in terms of queuing problems, regeneration applies to arbitrary stochastic processes. Rubinstein and Melamed (1998, Examples 3.7.2 and 3.7.4), for instance, consider a problem in inventory control. The example below shows how the general structure above produces a specific loss function for use in an optimization problem.

Example 14.3—Loss function for a queuing system. Consider the *GI/G/1* queuing problem of Example 14.2. Suppose that the i.i.d. service and interarrival times S_k and $A_{k|k+1}$ depend on $\boldsymbol{\theta}$. Then, we can define a cost $C_{\boldsymbol{\theta}}$ as the total time in the system for customers in a given regeneration period. The kth customer's time in the system is the sum of the wait time W_k and service time S_k. Further, let $\ell_{\boldsymbol{\theta}}$ denote the total number of customers processed during a period. Given the i.i.d. nature of the interarrivals and the service times, $E(C_{\boldsymbol{\theta}})$ and $E(\ell_{\boldsymbol{\theta}})$ do not depend on the specific period (i.e., the regeneration periods are i.i.d.). Hence, $L(\boldsymbol{\theta})$ according to (14.2) is an expression of the typical wait time for each customer. Following Fu (1990) and L'Ecuyer and Glynn (1994), this loss for an arbitrary regeneration period can be written as

$$L(\boldsymbol{\theta}) = \frac{E\left[\sum_{k\in\text{period}}(W_k + S_k)\right]}{E(\text{number of customers in period})},$$

where the period is arbitrary because the regeneration process is i.i.d. (the numerator sum is over a *random* number of customers in one period). A similar optimization criterion is considered in Rubinstein and Melamed (1998, Sect. 3.7), except that the problem is inverted from that above in that the one seeks to *maximize* the number of customers processed per unit time. ❑

14.2.2 Estimators of $L(\boldsymbol{\theta})$

Given the regenerative structure, there is a natural estimator of L at any value of $\boldsymbol{\theta}$. This estimator can form the basis for input into one of the search and optimization algorithms that depend only on noisy measurements of the loss function (such as the stochastic approximation [SA] algorithms, finite-difference SA [FDSA] and simultaneous perturbation SA [SPSA], introduced in Chapters 6 and 7). Let $C_{\boldsymbol{\theta},i}$ denote the total cost over the ith regeneration period, with $\ell_{\boldsymbol{\theta},i}$ the corresponding length. Then, directly motivated by the definition in (14.2), an estimator of $L = L(\boldsymbol{\theta})$ is

$$\hat{L}_N(\boldsymbol{\theta}) \equiv \frac{N^{-1}\sum_{i=1}^{N} C_{\boldsymbol{\theta},i}}{N^{-1}\sum_{i=1}^{N} \ell_{\boldsymbol{\theta},i}} = \frac{\sum_{i=1}^{N} C_{\boldsymbol{\theta},i}}{\sum_{i=1}^{N} \ell_{\boldsymbol{\theta},i}}, \tag{14.3}$$

where N is the number of regeneration periods. Suppose that the variances of $C_{\boldsymbol{\theta},i}$ and $\ell_{\boldsymbol{\theta},i}$ exist (i.e., are finite). Then, by the strong law of large numbers (Appendix C, including Exercise C.6),

$$\lim_{N\to\infty} \hat{L}_N(\boldsymbol{\theta}) = \lim_{N\to\infty} \frac{N^{-1}\sum_{i=1}^{N} C_{\boldsymbol{\theta},i}}{N^{-1}\sum_{i=1}^{N} \ell_{\boldsymbol{\theta},i}} = \frac{E(C_{\boldsymbol{\theta}})}{E(\ell_{\boldsymbol{\theta}})} = L(\boldsymbol{\theta}) \text{ a.s.}$$

However, $\hat{L}_N(\theta)$ is a *biased* estimator for any finite N because

$$E[\hat{L}_N(\theta)] = E\left(\frac{\sum_{i=1}^N C_{\theta,i}}{\sum_{i=1}^N \ell_{\theta,i}}\right) \neq \frac{E\left(N^{-1}\sum_{i=1}^N C_{\theta,i}\right)}{E\left(N^{-1}\sum_{i=1}^N \ell_{\theta,i}\right)} = L(\theta).$$

This bias indicates that, in general, the use of $\hat{L}_N(\theta)$ as the input $y = y(\theta)$ in an optimization algorithm must be done with caution. Further, the bias generally depends on θ; so, in forming *differences* of loss measurements at two distinct values of θ, the biases generally do not cancel. The FDSA (Chapter 6) and SPSA (Chapter 7) algorithms have varying conditions on the bias of the input. If the input is unbiased, the relevant conditions for all of the algorithms are satisfied. Hence, it is desirable to seek an unbiased input. We analyze the bias in more detail after the following example illustrating (14.3).

Example 14.4—Estimate of $L(\theta)$ for a queuing system. Let us use the data depicted in Figure 14.1 to construct the estimator of $L(\theta)$ given in (14.3). As in Example 14.3, the cost C_θ is the waiting time plus the service time. Further, let the length ℓ_θ be measured in terms of the number of customers processed in a regeneration period. For the θ used in generating the data, the estimate of $L(\theta)$ based on (14.3) is

$$\frac{\sum_i \text{wait \& service times for customers in period } i}{\sum_i \text{number of customers in period } i} = \frac{\sum_i \sum_{k \in \text{period } i} (W_k + S_k)}{\text{total number of customers}},$$

where the sums are over all regeneration periods. Figure 14.1 shows five complete regeneration periods and the beginning of a sixth. Suppose that the cumulative service time over the five complete periods is 10.0. This leads to an estimate of

$$\frac{\overbrace{1.0+0+2.5+6.0+7.0}^{\text{wait times}} + \overbrace{10.0}^{\substack{\text{cum. service}\\ \text{time}}}}{2+1+3+4+5} = \frac{26.5}{15} = 1.77.$$

(Note that only 15 of the 17 depicted arrivals are used due to the lack of information on the sixth regeneration period, which is only known to contain *at least* two arrivals.) As discussed above, however, this estimate of $L(\theta)$ at the given θ is biased. Modified conditions or estimators are considered below to mitigate this bias. ❑

Let us now discuss three cases where $\hat{L}_N(\theta)$ (or a closely related estimate) may be unbiased—or nearly unbiased—and may therefore be useful as an input to an optimization algorithm. The first case is where the number of

regeneration periods N is large enough so that $\hat{L}_N(\boldsymbol{\theta})$ is reasonably close to $L(\boldsymbol{\theta})$ *and* where it is valid, correspondingly, to say that $E[\hat{L}_N(\boldsymbol{\theta})]$ is close to $L(\boldsymbol{\theta})$. (The latter result does not automatically follow since a.s. convergence does *not* generally imply convergence of the mean; see, e.g., Example C.4 in Appendix C.) The well-known dominated convergence theorem from analysis provides sufficient conditions for $E[\hat{L}_N(\boldsymbol{\theta})]$ to converge to $L(\boldsymbol{\theta})$ as $N \to \infty$; see Subsection C.2.3 in Appendix C. One specific form of the dominated convergence theorem, the bounded convergence theorem (Corollary 2 to Theorem C.1), holds if $\ell_{\boldsymbol{\theta}} \geq m > 0$ and $0 \leq C_{\boldsymbol{\theta}} \leq M < \infty$. Under these conditions on $\ell_{\boldsymbol{\theta}}$ and $C_{\boldsymbol{\theta}}$, it follows that $0 \leq \hat{L}_N(\boldsymbol{\theta}) \leq M/m < \infty$ for all N, satisfying the key condition of the bounded convergence theorem and guaranteeing that $E[\hat{L}_N(\boldsymbol{\theta})] \to L(\boldsymbol{\theta})$ as $N \to \infty$. Suppose that the dominated (or bounded) convergence theorem hold for all $\boldsymbol{\theta}$. Then, for each $\boldsymbol{\theta}$, $E[\hat{L}_N(\boldsymbol{\theta})]$ is close to $L(\boldsymbol{\theta})$ for N reasonably large. For many practical applications, the resulting small bias (i.e., $E[\hat{L}_N(\boldsymbol{\theta})] \approx L(\boldsymbol{\theta})$, not $E[\hat{L}_N(\boldsymbol{\theta})] = L(\boldsymbol{\theta})$) would be acceptable in an optimization algorithm that formally requires an unbiased loss estimate because it would not be expected to affect the iteration process in a significant way.

The second case associated with controlling the bias of $\hat{L}_N(\boldsymbol{\theta})$ is when $\ell_{\boldsymbol{\theta}}$ is not a random quantity. We mentioned above a couple of periodic systems where this might be true (e.g., daily operations in a factory, where $\ell_{\boldsymbol{\theta}}$ involves the processing of a repeatable number of parts or the elapsed time over which a certain number of operations are performed). Many other such systems exist. Here the denominator in (14.3) is replaced by the deterministic $\ell_{\boldsymbol{\theta}}$, leading to

$$\hat{L}_N(\boldsymbol{\theta}) \equiv \frac{N^{-1}\sum_{i=1}^{N} C_{\boldsymbol{\theta},i}}{\ell_{\boldsymbol{\theta}}}, \tag{14.4}$$

which is clearly unbiased as an estimator of $L(\boldsymbol{\theta})$. Note that, in general, $L(\boldsymbol{\theta})$ in this setting still depends on a *ratio* as in (14.2) (i.e., $\ell_{\boldsymbol{\theta}}$ continues to depend on $\boldsymbol{\theta}$). So it is not sufficient to simply minimize the numerator $E(C_{\boldsymbol{\theta}})$.

The third case associated with controlling the bias returns to the general definition of $L(\boldsymbol{\theta})$ involving random $C_{\boldsymbol{\theta}}$ and $\ell_{\boldsymbol{\theta}}$ values. The distinction here is that the estimate of $L(\boldsymbol{\theta})$ is based on the $C_{\boldsymbol{\theta}}$ values being statistically independent of the $\ell_{\boldsymbol{\theta}}$ values. This contrasts with the estimate in (14.3) where the $C_{\boldsymbol{\theta}}$ and $\ell_{\boldsymbol{\theta}}$ values come from the same regeneration periods and will likely be highly dependent. Suppose that there are two groups of mutually exclusive regeneration periods, one group generating the values for $C_{\boldsymbol{\theta}}$ and the other group generating the values for $\ell_{\boldsymbol{\theta}}$. Then, an obvious modification of the estimator in (14.3) is

$$\hat{L}_{N_1,N_2}(\boldsymbol{\theta}) \equiv \frac{N_1^{-1}\sum_{i\in\text{Group(1)}} C_{\boldsymbol{\theta},i}}{N_2^{-1}\sum_{i\in\text{Group(2)}} \ell_{\boldsymbol{\theta},i}}, \tag{14.5}$$

where Group(1) and Group(2) denote the indices for the two groups of mutually exclusive regeneration periods and N_1 and N_2 denote the number of regeneration periods in each of the two groups. So, the numerator sum represents the sample mean cost over the periods in the first group and the denominator sum is the sample mean length of all periods in the second group. For example, if there are a total of five regeneration periods at the indicated $\boldsymbol{\theta}$, the numerator sum might be based on data from $N_1 = 2$ of the periods while the denominator sum relies on data from the remaining $N_2 = 3$ periods. (This mutually exclusive idea is also used below in the context of gradient-based methods.)

In contrast to the estimate $\hat{L}_N(\boldsymbol{\theta})$, the independence of the regeneration periods allows the factoring into separate expectations associated with $C_{\boldsymbol{\theta}}$ and with $\ell_{\boldsymbol{\theta}}$:

$$E[\hat{L}_{N_1,N_2}(\boldsymbol{\theta})] = E\left(N_1^{-1}\sum_{i\in\text{Group(1)}} C_{\boldsymbol{\theta},i}\right)E\left(\frac{1}{N_2^{-1}\sum_{i\in\text{Group(2)}} \ell_{\boldsymbol{\theta},i}}\right)$$

$$= E(C_{\boldsymbol{\theta}})E\left(\frac{N_2}{\sum_{i\in\text{Group(2)}} \ell_{\boldsymbol{\theta},i}}\right). \tag{14.6}$$

Nevertheless, like $\hat{L}_N(\boldsymbol{\theta})$, $\hat{L}_{N_1,N_2}(\boldsymbol{\theta})$ remains a biased estimator because the rightmost expectation in (14.6) is *not* equal to $1/E(\ell_{\boldsymbol{\theta}})$. Unlike $\hat{L}_N(\boldsymbol{\theta})$, however, $\hat{L}_{N_1,N_2}(\boldsymbol{\theta})$ in (14.5) is in a form for which the bias can be bounded using a powerful tool from probability theory—the *Kantorovich inequality* (e.g., Clausing, 1982).

The Kantorovich inequality relates $1/E(X)$ to $E(1/X)$ for a positive random variable X. In particular, if $0 < m \le X \le M < \infty$ (a.s.) for some positive constants $m \le M$, the Kantorovich inequality states that

$$1 \le E(X)E\left(\frac{1}{X}\right) \le \frac{(m+M)^2}{4mM}. \tag{14.7}$$

(More precise Kantorovich-type bounds are available under stronger conditions on the distribution of the random variable X; see, e.g., Wilkins, 1955; Clausing, 1982; or Watson, 1987. The lower bound in (14.7) also follows by Jensen's inequality of probability theory [see, e.g., Laha and Rohatgi, 1979, p. 368].) Hence, $E(1/X) \ge 1/E(X)$ for such bounded positive random variables X. In our case, $\ell_{\boldsymbol{\theta}}$ plays the role of X. Let $0 < m(\boldsymbol{\theta}) \le \ell_{\boldsymbol{\theta}} \le M(\boldsymbol{\theta})$ (in general, it is possible that $C_{\boldsymbol{\theta}} < 0$, implying that, potentially, $L(\boldsymbol{\theta}) < 0$). Applying the Kantorovich inequality (14.7) to the estimate in (14.5) yields the following upper bound to the magnitude of the bias of $\hat{L}_{N_1,N_2}(\boldsymbol{\theta})$:

$$0 \le |\text{bias}| = \left|E[\hat{L}_{N_1,N_2}(\boldsymbol{\theta})] - L(\boldsymbol{\theta})\right| \le |L(\boldsymbol{\theta})| \frac{[M(\boldsymbol{\theta}) - m(\boldsymbol{\theta})]^2}{4m(\boldsymbol{\theta})M(\boldsymbol{\theta})} \tag{14.8}$$

(Exercise 14.1). Note that the absolute value signs in (14.8) are superfluous in the common case where $L(\boldsymbol{\theta}) \ge 0$.

Table 14.1 shows some values for the Kantorovich-based bound as a function of the ratio $M(\boldsymbol{\theta})/m(\boldsymbol{\theta})$. The biases shown are normalized by $L(\boldsymbol{\theta})$ (i.e., the right-hand side of (14.8) divided through by $|L(\boldsymbol{\theta})|$, leaving the unit-free ratio in terms of $M(\boldsymbol{\theta})$ and $m(\boldsymbol{\theta})$). Hence the values shown are the bias as a fraction of the loss value. The table shows, as expected, that as $M(\boldsymbol{\theta})$ and $m(\boldsymbol{\theta})$ get closer, the bias gets smaller. In the limit, where $M(\boldsymbol{\theta}) = m(\boldsymbol{\theta})$, we have the deterministic $\ell_{\boldsymbol{\theta}}$ case of (14.4), where the bias is zero. When the upper bound to $\ell_{\boldsymbol{\theta}}$ is twice the lower bound, we see that the bias can be no more that 12.5 percent of the loss value. When the upper bound to $\ell_{\boldsymbol{\theta}}$ is only 10 percent greater than the lower bound, the bias can be no more than 0.23 percent of the loss value. For most practical applications, such a small bias would be considered negligible, indicating that the estimate $\hat{L}_{N_1,N_2}(\boldsymbol{\theta})$ would be acceptable in an optimization algorithm that formally requires an unbiased loss estimate.

Applications such as the traffic and inventory examples mentioned at the beginning of this section, where *overall* patterns may exhibit little variation, would be the type of problems for which a fairly narrow range on $\ell_{\boldsymbol{\theta}}$ might be specified, leading to a negligible bias in the estimate of the loss function. Another class of problems would be queuing systems where the arrival times and service times are fairly predictable, leading to a relatively tight bound on $\ell_{\boldsymbol{\theta}}$.

Table 14.1. Kantorovich-based upper bound to the magnitude of the bias of loss estimate (14.5) as a fraction of $L(\boldsymbol{\theta})$.

$\frac{M(\boldsymbol{\theta})}{m(\boldsymbol{\theta})}$	Upper bound to $\frlvert\frac{\text{bias}}{L(\boldsymbol{\theta})}\frrvert$
1.10	0.0023
1.25	0.0125
1.50	0.0417
2.00	0.125

14.2.3 Estimates Related to the Gradient of $L(\boldsymbol{\theta})$

As introduced in the stochastic gradient discussion of Chapter 5, and as considered in detail for simulations in Chapter 15, powerful optimization methods are available if a direct unbiased estimate of the gradient $\boldsymbol{g}(\boldsymbol{\theta}) = \partial L/\partial \boldsymbol{\theta}$ can be determined. Such stochastic gradient methods represent an application of (Robbins–Monro) root-finding stochastic approximation. While the rest of this chapter emphasizes methods for optimization that do not rely on direct measurements of the gradient, this subsection shows how regeneration can be exploited to create a direct gradient estimator. This gradient-based structure motivates a nongradient (SPSA)-based estimator in Section 14.3.

Consider the basic loss estimate $\hat{L}_N(\boldsymbol{\theta})$ in (14.3). A direct estimate of the gradient is available by differentiating $\hat{L}_N(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Fu (1990) shows that this estimate is strongly convergent (i.e., a.s. convergent to $\boldsymbol{g}(\boldsymbol{\theta})$ as $N \to \infty$) if the following condition governing the interchange of differentiation and expectation (see Appendix A) holds:

$$E(\ell_{\boldsymbol{\theta}})E\left(\frac{\partial C_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\right) - E(C_{\boldsymbol{\theta}})E\left(\frac{\partial \ell_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\right) = E(\ell_{\boldsymbol{\theta}})\frac{\partial E(C_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} - E(C_{\boldsymbol{\theta}})\frac{\partial E(\ell_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}}. \quad (14.9)$$

Analogous to $\hat{L}_N(\boldsymbol{\theta})$ itself, however, the estimate $\partial \hat{L}_N(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ is *biased* for finite N (i.e., $E[\partial \hat{L}_N(\boldsymbol{\theta})/\partial \boldsymbol{\theta}] \neq \boldsymbol{g}(\boldsymbol{\theta})$). So, in light of the regularity conditions for stochastic gradient methods requiring an unbiased gradient estimate (Section 5.1), this estimate is formally inappropriate.

By recasting the root-finding problem slightly, however, Fu (1990) and L'Ecuyer and Glynn (1994) introduce a means for coping with the bias in the estimate $\partial \hat{L}_N(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$. In particular, under the assumptions stated for the definition of $L(\boldsymbol{\theta})$ in (14.2), finding the optimizing root $\boldsymbol{\theta}^*$ of $\boldsymbol{g}(\boldsymbol{\theta}) = \mathbf{0}$ (the root producing a global minimum of $L(\boldsymbol{\theta})$) is equivalent to finding the optimizing root of

$$\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta}) \equiv E(\ell_{\boldsymbol{\theta}})\frac{\partial E(C_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} - E(C_{\boldsymbol{\theta}})\frac{\partial E(\ell_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \mathbf{0}, \quad (14.10)$$

where the notation $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta})$ is meant to evoke an expression *equivalent* to $\boldsymbol{g}(\boldsymbol{\theta})$ in the sense that both functions have the same zero (see Exercise 14.2). Now, we need to find an unbiased estimate of $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta})$. One way is to observe at least two regenerative periods at a specified $\boldsymbol{\theta}$, calculating $C_{\boldsymbol{\theta}}$ and $\partial C_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ from one or more regenerative periods and $\ell_{\boldsymbol{\theta}}$ and $\partial \ell_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ from the remaining regenerative periods (so $C_{\boldsymbol{\theta}}$ and $\partial C_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ are statistically independent of $\ell_{\boldsymbol{\theta}}$ and $\partial \ell_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$). The same notion was used in the modified loss estimate in (14.5), where N_1 regenerative periods were used to estimate $C_{\boldsymbol{\theta}}$ while N_2 different periods were used to estimate $\ell_{\boldsymbol{\theta}}$.

Extending this idea to the gradient-based setting here, one can form the estimate of $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta})$:

$$\hat{\boldsymbol{g}}_{\text{eqv}}(\boldsymbol{\theta}) = \overline{\ell_{\boldsymbol{\theta}}} \, \overline{\frac{\partial C_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}} - \overline{C_{\boldsymbol{\theta}}} \, \overline{\frac{\partial \ell_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}}, \tag{14.11}$$

where the overbars denote a sample mean of the indicated quantity over the relevant regeneration periods. For example, if there are five regeneration periods at the indicated $\boldsymbol{\theta}$, the sample means of $C_{\boldsymbol{\theta}}$ and $\partial C_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ might be based on data from $N_1 = 2$ of the periods while the sample means of $\ell_{\boldsymbol{\theta}}$ and $\partial \ell_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ rely on data from the remaining $N_2 = 3$ periods. With the minimum number of required regeneration periods—two—the sample means are the individual values of $C_{\boldsymbol{\theta}}$ and $\partial C_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ from one period and $\ell_{\boldsymbol{\theta}}$ and $\partial \ell_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ from the other period (i.e., equivalent to the right-hand side of (14.11) with no overbars). Fu (1990) establishes the following result:

Proposition 14.1. Suppose that the samples of $\ell_{\boldsymbol{\theta}}$ are independent of the samples of $\partial C_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ and that the samples of $\partial \ell_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$ are independent of the samples of $C_{\boldsymbol{\theta}}$. Further, suppose that the interchange condition given in (14.9) holds. Then, $\hat{\boldsymbol{g}}_{\text{eqv}}(\boldsymbol{\theta})$ is an unbiased estimator of $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta})$.

Proof. We have

$$\begin{aligned} E[\hat{\boldsymbol{g}}_{\text{eqv}}(\boldsymbol{\theta})] &= E\left(\overline{\ell_{\boldsymbol{\theta}}} \, \overline{\frac{\partial C_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}} - \overline{C_{\boldsymbol{\theta}}} \, \overline{\frac{\partial \ell_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}} \right) \\ &= E(\overline{\ell_{\boldsymbol{\theta}}}) E\left(\overline{\frac{\partial C_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}} \right) - E(\overline{C_{\boldsymbol{\theta}}}) E\left(\overline{\frac{\partial \ell_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}} \right) \\ &= E(\ell_{\boldsymbol{\theta}}) E\left(\frac{\partial C_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \right) - E(C_{\boldsymbol{\theta}}) E\left(\frac{\partial \ell_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \right), \end{aligned}$$

where the second equality follows by the assumed independence. By (14.9) and the definition of $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta})$ in (14.10), the proof is complete. ❑

Given the estimate in (14.11), one can now use the root-finding version of the stochastic approximation algorithm (Chapter 4) to attempt to find the optimizing root $\boldsymbol{\theta}^*$ of $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta}) = \mathbf{0}$. In particular, the required input to the root-finding algorithm at a given value of $\boldsymbol{\theta}$ is $\hat{\boldsymbol{g}}_{\text{eqv}}(\boldsymbol{\theta})$. So, in the root-finding notation, in going from the estimate of $\boldsymbol{\theta}$ at iteration k to the estimate at iteration $k + 1$, the input for unconstrained or constrained SA is $\boldsymbol{Y}_k(\hat{\boldsymbol{\theta}}_k) = \hat{\boldsymbol{g}}_{\text{eqv}}(\hat{\boldsymbol{\theta}}_k)$. From Proposition 14.1, an unbiasedness condition of root-finding SA (Section 4.3) is satisfied: $E[\boldsymbol{Y}_k(\hat{\boldsymbol{\theta}}_k) \,|\, \hat{\boldsymbol{\theta}}_0, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_k] = \boldsymbol{g}_{\text{eqv}}(\hat{\boldsymbol{\theta}}_k)$ provided that the required

regeneration periods at $\hat{\theta}_k$ satisfy the independence conditions of Proposition 14.1 *and* are also independent from the periods generated at earlier values, $\hat{\theta}_0$, $\hat{\theta}_1, \ldots, \hat{\theta}_{k-1}$. Hence, the Robbins–Monro root-finding SA algorithm can be used when at least two regeneration periods are observed in the simulation at each $\hat{\theta}_k$ (in order to produce the estimate $Y_k(\hat{\theta}_k) = \hat{g}_{\text{eqv}}(\hat{\theta}_k)$) *and* when the regeneration periods are produced from an independent random number seed at each $\hat{\theta}_k$. This unbiasedness holds at each $\hat{\theta}_k$ for any *finite* number of regeneration periods satisfying Proposition 14.1.

For a *GI/G*/1 queue, Fu (1990) and L'Ecuyer and Glynn (1994) employ $Y_k(\hat{\theta}_k) = \hat{g}_{\text{eqv}}(\hat{\theta}_k)$ with root-finding SA to find the solution to $g_{\text{eqv}}(\theta) = 0$, where

$$g_{\text{eqv}}(\theta) = 0 \iff \frac{\partial L(\theta)}{\partial \theta} = \frac{\partial(\text{wait time/customer})}{\partial \theta} = 0$$

($\Leftrightarrow$ is read as "if and only if"). However, the requirement for obtaining the gradients $\partial C_\theta/\partial\theta$ and $\partial \ell_\theta/\partial\theta$ needed to compute $\hat{g}_{\text{eqv}}(\hat{\theta}_k)$ is stringent, as it involves detailed knowledge of the formulas and probability distributions embedded in the simulation. Thus, Section 14.3 discusses an application of this general setting when it is not possible to directly calculate the gradients needed in $\hat{g}_{\text{eqv}}(\hat{\theta}_k)$. This approach uses the simultaneous perturbation gradient estimate (Chapter 7) to approximate the gradients.

14.3 OPTIMIZATION WITH FINITE-DIFFERENCE AND SIMULTANEOUS PERTURBATION GRADIENT ESTIMATORS

Chapters 4–7 discussed general methods and theory for stochastic approximation. SA methods appear to be the leading formal techniques for optimization in a simulation context. Among the reasons for the popularity are the algorithmic similarity to standard deterministic methods (i.e., the similarity in structure between SA and steepest descent-type algorithms) and the strong theoretical basis for accommodating noisy loss and/or gradient measurements. As discussed previously, the latter concern is inherent in the Monte Carlo simulation context. Central to the SA approach is some type of estimator for the gradient $\partial L/\partial\theta$.

Chapters 6 and 7 discussed two gradient estimators that do not rely on direct gradient information (FDSA and SPSA). Both have been applied in simulation-based optimization, with the basic FDSA being the oldest and best-known approach. (Based on the root-finding SA algorithm, Chapter 15 focuses on direct stochastic gradient estimators of $\partial L/\partial\theta$ in the simulation context, as discussed in general form in Chapter 5.) In general, FDSA and SPSA apply to

arbitrary simulations and loss functions $L(\boldsymbol{\theta}) = E[Q(\boldsymbol{\theta}, \boldsymbol{V})]$, with Q representing the simulation output, as defined in Section 14.1. In particular, there is no need to assume a special form for the simulation, such as regenerative, to employ these gradient-free methods (although the end of this section discusses some unique structure that *is* available if the simulation is for a regenerative system).

In fact, these gradient-free algorithms are the most popular of the iterative optimization algorithms for simulation: "...this estimator [FDSA], or some variant of it, remains the method of choice for the majority of practitioners" (Fu and Hu, 1997, p. 5). This may be surprising given the much greater effort that has gone into the more sophisticated model-based approaches to be discussed in Chapter 15 (such as infinitesimal perturbation analysis and the likelihood ratio method), but this is a reflection of the (usually) much greater ease of implementation. The finite-difference and simultaneous perturbation gradient estimators are based on values of $Q(\boldsymbol{\theta}, \boldsymbol{V})$ (representing the noisy measurements of $L(\boldsymbol{\theta})$) at appropriately perturbed values of $\boldsymbol{\theta}$, as discussed in Chapters 6 and 7. There is no need to know the inner workings of the simulation, as required in the model-based approaches, which need gradients such as $\partial Q/\partial \boldsymbol{\theta}$. Rather, one simply specifies $\boldsymbol{\theta}$ and runs the simulation—precisely what the simulation was designed to do!—to produce outputs $Q(\boldsymbol{\theta}, \boldsymbol{V})$.

The FDSA and SPSA gradient estimates are as follows. The ith component of the two-sided finite-difference gradient estimate is

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_k) = \frac{Q(\hat{\boldsymbol{\theta}}_k + c_k \boldsymbol{\xi}_i, \boldsymbol{V}_{ki}^{(+)}) - Q(\hat{\boldsymbol{\theta}}_k - c_k \boldsymbol{\xi}_i, \boldsymbol{V}_{ki}^{(-)})}{2c_k},$$

where the $\boldsymbol{V}_{ki}^{(\pm)}$ terms represent the Monte Carlo random sequences in the given simulations, $\boldsymbol{\xi}_i$ is the unit vector with a 1 in the ith place and a 0 elsewhere, and $c_k \to 0$ as $k \to \infty$. (Note that $\boldsymbol{V}_{ki}^{(\pm)}$, in general, depends on the value of $\boldsymbol{\theta}$, either directly or indirectly through the dependence of the distribution generating the $\boldsymbol{V}$ values on $\boldsymbol{\theta}$.)

Analogously, the ith component of the simultaneous perturbation gradient estimate is

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_k) = \frac{Q(\hat{\boldsymbol{\theta}}_k + c_k \boldsymbol{\Delta}_k, \boldsymbol{V}_k^{(+)}) - Q(\hat{\boldsymbol{\theta}}_k - c_k \boldsymbol{\Delta}_k, \boldsymbol{V}_k^{(-)})}{2c_k \Delta_{ki}},$$

where the Monte Carlo generated p-dimensional random perturbation vector, $\boldsymbol{\Delta}_k = [\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}]^T$, has a user-specified distribution satisfying conditions discussed in Sections 7.2 and 7.3 (a valid distribution is Bernoulli ± 1 for the individual components Δ_{ki}). Unlike the FDSA gradient estimate, there is only a single subscript index on $\boldsymbol{V}$, reflecting the simultaneous perturbation of all components in $\boldsymbol{\theta}$.

Given a gradient estimate above, the procedure for determining an estimate of the optimal parameters $\boldsymbol{\theta}^*$ follows exactly as in Chapters 6 and 7. Namely, an initial guess $\hat{\boldsymbol{\theta}}_0$ for the parameters is made and then simulations are run to form the gradient estimates, $\hat{\boldsymbol{g}}_0(\hat{\boldsymbol{\theta}}_0)$, $\hat{\boldsymbol{g}}_1(\hat{\boldsymbol{\theta}}_1),\ldots$, needed in updating the $\boldsymbol{\theta}$ estimate according to $\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ (or $\hat{\boldsymbol{\theta}}_{k+1} = \Psi_\Theta[\hat{\boldsymbol{\theta}}_k - a_k\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)]$ when there is a constraint mapping Ψ_Θ, as discussed in Sections 4.1 and 7.7).

With the exception of certain properties associated with common random numbers discussed below, the relative performance of FDSA and SPSA in the simulation context will be as discussed in the generic description of Chapters 6 and 7. The relative performance follows the familiar fundamental relationship on the cost of the gradient approximations: Given that one simulation run produces one value of Q, then $2p$ runs of the simulation are required to produce the standard two-sided finite-difference gradient estimate while only two runs are needed for the simultaneous perturbation estimate. An example simulation-based implementation illustrating this relative cost is in Fu and Hill (1997).

There is also a convenient connection of the FDSA and SPSA approaches to the regenerative structure of Section 14.2. Suppose that the system under study is regenerative and that the loss function in (14.2), $L(\boldsymbol{\theta}) = E(C_{\boldsymbol{\theta}})/E(\ell_{\boldsymbol{\theta}})$, will be used. It was shown in Subsection 14.2.3 that finding the optimum $\boldsymbol{\theta}^*$ via $\boldsymbol{g}(\boldsymbol{\theta}) = \partial L/\partial\boldsymbol{\theta} = \boldsymbol{0}$ is equivalent to finding the optimizing root of $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta}) = \boldsymbol{0}$, where $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta})$ is defined in (14.10). An unbiased estimate $\hat{\boldsymbol{g}}_{\text{eqv}}(\boldsymbol{\theta})$ was described in (14.11), which is based on sample means of the gradients $\partial C_{\boldsymbol{\theta}}/\partial\boldsymbol{\theta}$ and $\partial\ell_{\boldsymbol{\theta}}/\partial\boldsymbol{\theta}$ across several simulations (with the simulations for $\partial C_{\boldsymbol{\theta}}/\partial\boldsymbol{\theta}$ being independent of the simulations for $\partial\ell_{\boldsymbol{\theta}}/\partial\boldsymbol{\theta}$). This unbiased estimate is appropriate for a root-finding SA algorithm, as discussed in Subsection 14.2.3.

Suppose that the simulation is sufficiently complex so that it is not possible to compute the gradients, $\partial C_{\boldsymbol{\theta}}/\partial\boldsymbol{\theta}$ and $\partial\ell_{\boldsymbol{\theta}}/\partial\boldsymbol{\theta}$. Finite differences (L'Ecuyer and Glynn, 1994) or simultaneous perturbation (Fu and Hill, 1997) could then be used to estimate the gradients. The SPSA-based approximations to the two required gradients lead to the following gradient-free estimate of the ith component of $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta})$ at the kth iteration:

$$\hat{g}_{\text{eqv};ki}(\hat{\boldsymbol{\theta}}_k) = \frac{C_k^{(+)} - C_k^{(-)}}{2c_k\Delta_{ki}}\ell_k^{(-)} - C_k^{(-)}\frac{\ell_k^{(+)} - \ell_k^{(-)}}{2c_k\Delta_{ki}}$$

$$= \frac{C_k^{(+)}\ell_k^{(-)} - C_k^{(-)}\ell_k^{(+)}}{2c_k\Delta_{ki}},$$

where $C_k^{(\pm)} = C_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k, \boldsymbol{V}_k^{(\pm)})$ and $\ell_k^{(\pm)} = \ell_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k, \boldsymbol{V}_k^{(\pm)})$. (In the spirit of SPSA, the same two pairs of values, $C_k^{(+)}$, $\ell_k^{(+)}$ and $C_k^{(-)}$, $\ell_k^{(-)}$, representing two runs of the simulation, are used for all p components of

$\hat{g}_{\text{eqv};k}(\hat{\theta}_k)$.) Fu and Hill (1997) discuss the conditions under which the above gradient estimate produces a convergent SPSA algorithm.

14.4 COMMON RANDOM NUMBERS

14.4.1 Introduction

Common random numbers (CRNs) provide one of the most useful tools in simulation-based optimization, being widely applied because of their effectiveness, ease of implementation, and intuitive appeal. CRNs provide—among other benefits—a means for reducing the variability of the gradient estimate that serves as the input to the SA iteration process. This, in turn, is able to reduce the error in the θ estimation process. CRNs have been studied extensively (see, e.g., Glasserman and Yao, 1992; L'Ecuyer and Perron, 1994; Rubinstein and Melamed, 1998, pp. 90–94; and Kleinman et al., 1999). We restrict our attention here to their use in the gradient-free estimators for SA algorithms discussed in Section 14.3 (FDSA and SPSA). Most of the explicit technical points are illustrated with SPSA, but it should be clear how the points will translate to the FDSA approach. Some of the references above include discussion on applications of CRNs beyond their use in such SA gradient estimators.

The essence of CRNs in a simulation-based optimization context is to use the same random numbers in the simulations being differenced in forming the gradient approximations. That is, the two function measurements in the numerator of one component of the gradient approximation each use a sequence of Monte Carlo-generated random variables, V, to generate the various random events in the process. CRNs are based on using the *same* values of V in both simulations. In this way, the change information being provided by the differences in the numerators of the gradient approximations is the result of the changes in the θ elements and not the changes due to different random variables in the Monte Carlo simulations. Consequently, CRNs can reduce the variability of the gradient approximations and thereby improve the performance of the optimization process.

One of the key conditions for CRNs to be effective is that the random numbers being generated in the simulations forming the difference should be used in comparable ways. For example, the random numbers generating the mth arrival into a queue in one simulation should correspond to the random numbers generating the mth arrival in the other simulation. This comparability manifests itself in requirements of monotonicity and synchronization, as described below. The following example is intended to help clarify the role of CRNs in practice.

Example 14.5—Plant layout revisited. Let us return to the plant layout problem (Example 14.1) for a *conceptual* example of CRNs. Suppose that we are using the SPSA algorithm to maximize the mean number of assembly operations

completed on a daily basis and that each calculation of $-Q$ represents one day's worth of completed operations (the negative sign reflects our standard use of Q as a loss measurement for a *minimization* problem). Suppose further that the only randomness in the process is the arrival times of parts into the assembly operations area. In forming the gradient estimate at one iteration, we need to collect the two noisy loss measurements $Q(\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k, \boldsymbol{V}_k^{(\pm)})$ where $\boldsymbol{V}_k^{(+)}$ and $\boldsymbol{V}_k^{(-)}$ represent the sequence of arrival times throughout the day of the parts under the machine locations $\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k$ and $\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k$.

If the calculations were to be done based on *real* experiments on the factory floor, then it is likely that the parts would arrive at different times in the two experiments needed (equivalent to $\boldsymbol{V}_k^{(+)} \neq \boldsymbol{V}_k^{(-)}$ in the simulation) since the experiments would be conducted on different days. Hence the difference in $Q(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k, \boldsymbol{V}_k^{(+)})$ and $Q(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k, \boldsymbol{V}_k^{(-)})$ would be due to differences in $\boldsymbol{\theta}$ *and* the difference in random effects (i.e., $\boldsymbol{V}_k^{(+)} \neq \boldsymbol{V}_k^{(-)}$). However, in simulation-based optimization, we may be able to impose the requirement that $\boldsymbol{V}_k = \boldsymbol{V}_k^{(+)} = \boldsymbol{V}_k^{(-)}$ through picking the same random number seed in generating the uniform random variables that are transformed to generate $\boldsymbol{V}_k^{(+)}$ and $\boldsymbol{V}_k^{(-)}$. Note that CRNs do not apply across iterations (i.e., $\boldsymbol{V}_k \neq \boldsymbol{V}_j$ for $k \neq j$) in order to ensure sufficient variability in the optimization process.

In determining the appropriateness of CRNs, consider two cases: (i) Parts can enter the assembly area at any time, irrespective of the state of the current assembly operations, and (ii) parts are not allowed to enter the assembly area until queues at certain critical machines are sufficiently cleared (i.e., there is limited buffer capacity). In (i), the random effects $\boldsymbol{V}$ are independent of the value of $\boldsymbol{\theta}$ (the machine locations). In (ii), the value of $\boldsymbol{\theta}$ *does* affect the arrival times because some arrival times will be delayed if the overall set of machine locations is causing excessive queue lengths at certain critical machines.

Under case (i) above, it is automatic to be able to impose CRNs by simply having the arrival times be the same in the two simulations. Under (ii), CRNs are feasible *provided* that the two sets of machine locations represented by $\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k$ are sufficiently close so that parts can be accepted into the assembly area equally in both simulations (so the parts enter the assembly area at the same times for each pair of simulated days). Under CRNs in either case (i) or (ii), the variability of the gradient estimate will be isolated to the difference in production output due to the change in machine location $\boldsymbol{\theta}$. Of course, to ensure that an optimal solution to $\boldsymbol{\theta}$ is found, the CRNs $\boldsymbol{V}_k$ should change at every iteration to represent the range of reasonable part arrivals in practice. ❑

Slightly more formally, we can see the positive effect of CRNs by recalling the well-known formula governing the variance of the difference of two random variables X and Y:

$$\text{var}(X - Y) = \text{var}(X) + \text{var}(Y) - 2\,\text{cov}(X, Y), \tag{14.12}$$

where we may view X and Y as representing the two numerator terms in one of the gradient approximations. It is apparent that *increasing* the covariance (i.e., increasing the correlation) between X and Y *decreases* the overall variance. In the simulation-based context, we can usually increase the covariance by using the same random number stream in the underlying simulations generating the two parts of the differences. CRNs are implemented very simply by using the same random number seed(s) in the two simulations (although this alone may not guarantee that the benefits of CRNs apply; see the discussion on partial CRNs in Subsection 14.4.3).[1]

14.4.2 Theory and Examples for Common Random Numbers

The benefit of the CRN method comes from a reduction in the variance of the difference $y_k^{(+)} - y_k^{(-)} = Q(\hat{\theta}_k + c_k\Delta_k, V_k^{(+)}) - Q(\hat{\theta}_k - c_k\Delta_k, V_k^{(-)})$ appearing in the numerator of the SPSA gradient approximation, where the $V_k^{(\pm)}$ terms represent the vector of Monte Carlo-generated random variables used in the two measurements. Intuitively, this variance reduction is achieved by CRNs when $Q(\hat{\theta}_k + c_k\Delta_k, V)$ and $Q(\hat{\theta}_k - c_k\Delta_k, V)$ tend to move in the same way for a change in V. This is formalized in Proposition 14.2. Suppose (as is typical) that the $V_k^{(\pm)}$ are generated as streams of random variables that have been formed from a transformation of a stream of $U(0, 1)$ random variables, say $U_k^{(\pm)}$. Further, suppose that the $V_k^{(\pm)}$ can be generated by the inverse transform method (Appendix D), with each component of $V_k^{(\pm)}$ generated by a corresponding single component of $U_k^{(\pm)}$. That is, write $V_{ki}^{(+)} = F_{ki}^{-1}(U_{ki}^{(+)})$ and $V_{ki}^{(-)} = G_{ki}^{-1}(U_{ki}^{(-)})$, where F_{ki} and G_{ki} represent the cumulative distribution functions for scalar components $V_{ki}^{(+)}$ and $V_{ki}^{(-)}$ (not necessarily the same distribution functions in general) and $U_{ki}^{(\pm)}$ represent the corresponding scalar components of $U_k^{(\pm)}$.

Note that there are two sources of variability in $y_k^{(+)} - y_k^{(-)}$: that due to $\pm c_k\Delta_k$ and that due to $V_k^{(\pm)}$. The former source is desirable since it reveals information about the shape of the loss function L. The latter source, on the other hand, is undesirable since the $V_k^{(\pm)}$ are the direct causes of the measurement noise (i.e., if there were no $V_k^{(\pm)}$ contributions, then $y_k^{(+)} - y_k^{(-)} = L(\hat{\theta}_k + c_k\Delta_k) - L(\hat{\theta}_k - c_k\Delta_k)$).

With CRNs, one may reduce the conditional variance $\text{var}[(y_k^{(+)} - y_k^{(-)}) | \hat{\theta}_k, \Delta_k]$ by picking $V_k^{(+)}$ to be as close as possible to $V_k^{(-)}$ through generating $U_k^{(+)}$ and $U_k^{(-)}$ in a manner so that they are as close as

[1] A complementary form to CRNs, *antithetic random numbers*, uses the summation form, $\text{var}(X + Y)$, to reduce the variability for sums of simulation runs. Rubinstein and Melamed (1998, pp. 90–94) show how antithetic random variables can be used to reduce variability when considering averages of simulation outputs. Chapter 16 also mentions an application in averaging multiple runs of the Markov chain Monte Carlo algorithms.

possible to each other. (Note that the conditional variance expressions here depend only on the current value of the θ estimate, $\hat{\theta}_k$, not on previous θ estimates. This contrasts with the conditioning in the general theoretical results of Chapter 7, which depend on the current and the previous θ estimates. In the simulation-based context of the current chapter, it is unnecessary to condition on the previous θ estimates because the V values generated at iteration k are assumed independent of the V values generated at previous iterations.) Reducing this conditional variance then translates into a corresponding reduction in the conditional and unconditional variance of the elements of the gradient estimate $\hat{g}_k(\hat{\theta}_k)$. In the ideal case, $U_k^{(+)} = U_k^{(-)}$, implying that $V_k^{(+)} = V_k^{(-)}$ (not always feasible in practice, as discussed below).

Typically, the practical implementation of CRNs is achieved by simply using the same random number seed in generating the sequences $U_k^{(+)}$ and $U_k^{(-)}$ (i.e., using the same seed achieves the same result as storing the sequence of random numbers for reuse, but it is usually easier to implement). Minimizing the variance of $y_k^{(+)} - y_k^{(-)}$ contributes to reducing the overall variability of the gradient estimate $\hat{g}_k(\hat{\theta}_k)$, which, in turn, reduces the variability of $\hat{\theta}_k$ (the ultimate interest!). The following result from Rubinstein et al. (1985) provides useful sufficient conditions for determining when CRNs can minimize the variance in the gradient estimate. Note that CRNs can *reduce* the variance (as opposed to *minimize* the variance) under conditions much weaker than those below (see, e.g., Exercise 14.6(b)). All statements involving conditional expectations (e.g., $\text{var}[\hat{g}_{ki}(\hat{\theta}_k) \mid \hat{\theta}_k]$) apply a.s.

Proposition 14.2. Suppose that $V_k^{(+)}$ and $V_k^{(-)}$ are each composed of a common number of independent components and that dependence is allowed only between like components of $V_k^{(+)}$ and $V_k^{(-)}$ (i.e., $V_{ki}^{(+)}$ and $V_{ki}^{(-)}$ may be dependent for all i, but $V_{ki}^{(+)}$ and $V_{kj}^{(-)}$ must be independent for $i \neq j$). Assume that $V_k^{(+)}$ and $V_k^{(-)}$ can be generated by the inverse transform method applied to each component of $U_k^{(+)}$ and $U_k^{(-)}$ (Appendix D). For almost all values of Δ_k, suppose that $Q(\hat{\theta}_k + c_k\Delta_k, V_k^{(+)})$ and $Q(\hat{\theta}_k - c_k\Delta_k, V_k^{(-)})$ are monotonic (nonincreasing or nondecreasing) in the same direction in each component of $V_k^{(+)}$ and $V_k^{(-)}$ within the range of allowable values for the components. (The direction may be nonincreasing for some components, nondecreasing for others.) Then $\text{var}[\hat{g}_{ki}(\hat{\theta}_k) \mid \hat{\theta}_k]$ is minimized at each i when $U_k^{(+)} = U_k^{(-)} = U_k$.

Sketch of Proof. Taking $y_k^{(\pm)} = Q(\hat{\theta}_k \pm c_k\Delta_k, V_k^{(\pm)})$, Rubinstein et al. (1985) (see also Rubinstein and Melamed, 1998, pp. 90–92) show that $\text{var}[(y_k^{(+)} - y_k^{(-)}) \mid \hat{\theta}_k, \Delta_k]$ is minimized when $U_k^{(+)} = U_k^{(-)} = U_k$ under the stated assumptions. The result then follows by working with the definition $\hat{g}_{ki}(\hat{\theta}_k) = (y_k^{(+)} - y_k^{(-)})/(2c_k\Delta_{ki})$ and converting to a conditioning solely on $\hat{\theta}_k$ (see Exercise 14.5). ❑

The result in Proposition 14.2 is intuitively plausible since $Q(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ and $Q(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ tend to move together as $\boldsymbol{V}$ changes. The monotonicity assumption states that *both* $Q(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ and $Q(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ must increase (or decrease) as the ith component of $\boldsymbol{V}$ increases. The direction of change in Q may differ with increases in different components of $\boldsymbol{V}$, but $Q(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ and $Q(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ must still change in the same direction.

The full result in Rubinstein et al. (1985) allows for more general dependence among the elements of $\boldsymbol{V}_k^{(+)}$ and $\boldsymbol{V}_k^{(-)}$ than the independence required in Proposition 14.2. We do not explore those extensions here, as they get into concepts (e.g., total positivity) that are beyond the scope of this book (the independence stated here is a special case). Glasserman and Yao (1992) provide a comprehensive discussion of the role of conditions such as monotonicity and the inverse transforms for generating the random variables ($\boldsymbol{V}$). They note, for example, that although the inverse transform method is part of the proposition here (and similar results), it is not necessary to *use* the inversion method in actual implementation. Rather, it must be assumed only that the inversion method is feasible for this problem. Other methods are acceptable if they produce the correct distribution. Glasserman and Yao (1992) provide a detailed discussion on the relationship of monotonicity to the synchronization of the random variables in the simulations.

In practice, how does one know if the monotonicity assumption in Proposition 14.2 is satisfied? There is no easy answer to this in general, but let us sketch a scenario where the assumption will be satisfied. Suppose, for example, that $Q(\boldsymbol{\theta}, \boldsymbol{V})$ is a continuous function of $\boldsymbol{\theta}$ or, more generally, that small changes in $\boldsymbol{\theta}$ do not alter the fundamental ordering and number of events represented in the simulation output. Then, for sufficiently small c_k (large k) the functions $Q(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ and $Q(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ will behave nearly identically in $\boldsymbol{V}$ (since $\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k$ will be nearly the same as $\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k$). Suppose further that $Q(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ and $Q(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k, \boldsymbol{V})$ are differentiable in $\boldsymbol{V}$ near some operating point $\boldsymbol{V}^*$ where the derivatives $\partial Q(\hat{\boldsymbol{\theta}}_k \pm c_k\boldsymbol{\Delta}_k, \boldsymbol{V})/\partial \boldsymbol{V}$ are nonzero, and that the randomly generated elements of $\boldsymbol{V}$ deviate from $\boldsymbol{V}^*$ by only a small magnitude. Then, there is evidence that the monotonicity is satisfied because $Q(\boldsymbol{\theta}, \boldsymbol{V})$ is locally (near $\boldsymbol{V}^*$) monotonic in $\boldsymbol{V}$ by the continuity, and $\boldsymbol{V}_k^{(+)}$ and $\boldsymbol{V}_k^{(-)}$ take on values near $\boldsymbol{V}^*$.

The following examples illustrate the types of problems for which Proposition 14.2 applies.

Example 14.6—Network design. Consider a transportation network with multiple origins and destinations. Suppose that an "omniscient authority" wishes to optimize some network design parameters ($\boldsymbol{\theta}$) to minimize the mean overall network cost. Let $\mathcal{V}_i$ represent the (random) cost of travel over the ith link in the network, with the links indexed by i. Suppose that the design parameters $\boldsymbol{\theta}$

appear in the probability distribution for the random costs (e.g., θ may represent aspects of the traffic signal operations). Let $OD(j)$ represent the collection of links associated with the jth customer's origin–destination pair, $j = 1, 2, \ldots, m$. For example, $OD(j) = \{2, 4, 7, 16\}$ indicates that the jth customer traveled over the links labeled 2, 4, 7, and 16 in going from origin to destination. With $\boldsymbol{V}$ representing the collection of travel times $\{\mathcal{V}_i\}$ the overall cost associated with travel over the network is

$$Q(\theta, \boldsymbol{V}) = \sum_{j=1}^{m} \sum_{i \in OD(j)} \mathcal{V}_i .$$

Note that θ does not appear explicitly in Q; rather, it appears implicitly via its effect on the distribution of the travel costs $\mathcal{V}_i$ (this phenomenon is considered in more detail in Section 15.2 in the context of likelihood ratio methods for gradient estimation). It is clear that Q is nondecreasing in each component of $\boldsymbol{V}$. Hence, if the elements of $\boldsymbol{V}$ can be generated by the inverse transform method, Proposition 14.2 indicates that CRNs provide a minimum variance gradient estimate. ❑

Example 14.7—Linear model. Suppose that the simulation output can be represented as a simple linear function of θ, i.e., $\boldsymbol{Z} = \boldsymbol{B}\theta$, where $\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{V})$ ($\boldsymbol{B}$ may be a *nonlinear* function of the random inputs $\boldsymbol{V}$). Given that $L(\theta) = E[Q(\theta, \boldsymbol{V})] = E[(\boldsymbol{Z} - \boldsymbol{d})^T(\boldsymbol{Z} - \boldsymbol{d})]$, where $\boldsymbol{d}$ is some "desired" vector, let us consider the variability of the SPSA gradient estimator. Note that

$$\begin{aligned} y_k^{(\pm)} &= Q(\hat{\theta}_k \pm c_k \Delta_k, \boldsymbol{V}_k^{(\pm)}) = [(\theta_k^{(\pm)})^T \boldsymbol{B}(\boldsymbol{V}_k^{(\pm)})^T - \boldsymbol{d}^T][\boldsymbol{B}(\boldsymbol{V}_k^{(\pm)})\theta_k^{(\pm)} - \boldsymbol{d}] \\ &= (\theta_k^{(\pm)})^T \boldsymbol{B}(\boldsymbol{V}_k^{(\pm)})^T \boldsymbol{B}(\boldsymbol{V}_k^{(\pm)})\theta_k^{(\pm)} - 2\boldsymbol{d}^T \boldsymbol{B}(\boldsymbol{V}_k^{(\pm)})\theta_k^{(\pm)} + \boldsymbol{d}^T \boldsymbol{d}, \end{aligned} \tag{14.13}$$

where $\theta_k^{(\pm)} = \hat{\theta}_k \pm c_k \Delta_k$. Proposition 14.2 indicates that CRNs provide a minimum variance gradient estimate when a positive change in the ith element of $\boldsymbol{V}_k^{(+)}$ and $\boldsymbol{V}_k^{(-)}$, $i = 1, 2, \ldots$, causes the same direction of change (either positive or negative) in $y_k^{(+)}$ and $y_k^{(-)}$ (i.e., monotonicity). Suppose that $\boldsymbol{V}$ can lie in a small domain surrounding some $\boldsymbol{V}^*$ and that $\boldsymbol{B}(\boldsymbol{V})$ is differentiable. Then if the corresponding components of $\partial y_k^{(+)}/\partial \boldsymbol{V}$ and $\partial y_k^{(-)}/\partial \boldsymbol{V}$ have the same sign at $\boldsymbol{V}^*$, the monotonicity assumption is satisfied.

As a specific example, suppose that $p = 2$, $\boldsymbol{V} = [\mathcal{V}_1, \mathcal{V}_2]^T$, and

$$\boldsymbol{B}(\boldsymbol{V}) = \begin{bmatrix} \mathcal{V}_1 & 0 \\ 0 & \mathcal{V}_2^2 \end{bmatrix}.$$

Then, from (14.13) and letting $\boldsymbol{d} = [d_1, d_2]^T$, the assumptions of Proposition 14.2 can be evaluated by considering the signs of

$$\frac{\partial y_k^{(\pm)}}{\partial V_1} = 2(\theta_{k1}^{(\pm)})^2 V_1 - 2\theta_{k1}^{(\pm)} d_1,$$

$$\frac{\partial y_k^{(\pm)}}{\partial V_2} = 4(\theta_{k2}^{(\pm)})^2 V_2^3 - 4\theta_{k2}^{(\pm)} d_2 V_2,$$

over the range of possible values for $\theta_k^{(\pm)}$ and V. For example, if $\hat{\theta}_k = [0, 0]^T$, the elements of $c_k\Delta_k$ are i.i.d. Bernoulli ± 0.1, and $\boldsymbol{d} = [1, 1]^T$, then from the derivative expressions above, it is known that $y_k^{(+)}$ and $y_k^{(-)}$ satisfy Proposition 14.2 if the possible V values lie in the region $(10, \infty] \times (\sqrt{10}, \infty]$. Both $y_k^{(+)}$ and $y_k^{(-)}$ are monotonically increasing in both components of V for V restricted to this region.

The essential form of the numerator $y_k^{(+)} - y_k^{(-)}$ in the SPSA gradient approximation is

$$\hat{\theta}_k^T [\boldsymbol{B}(V_k^{(+)})^T \boldsymbol{B}(V_k^{(+)}) - \boldsymbol{B}(V_k^{(-)})^T \boldsymbol{B}(V_k^{(-)})]\hat{\theta}_k$$
$$-2\boldsymbol{d}^T [\boldsymbol{B}(V_k^{(+)}) - \boldsymbol{B}(V_k^{(-)})]\hat{\theta}_k + O_P(c_k), \qquad (14.14)$$

where the term $O_P(c_k)$ represents a collection of random terms that are multiplied by c_k (the subscript P denotes a *probabilistic* version of standard big-O). Under assumptions such as those discussed in the preceding paragraph, Proposition 14.2 indicates that the resulting gradient approximation has minimum variance under CRNs. From (14.14), we see that under CRNs, $\text{var}[(y_k^{(+)} - y_k^{(-)}) \,|\, \hat{\theta}_k, \Delta_k] = O(c_k^2)$ ($c_k \to 0$ as usual) since all terms but the $O_P(c_k)$ term on the right-hand side of (14.14) cancel each other. (Because the conditional variances here are random variables, all statements here involving a conditional variance are only guaranteed to hold on a set of probability 1 [i.e., in the a.s. sense]—see Appendix C.) Under non-CRNs, $\text{var}[(y_k^{(+)} - y_k^{(-)}) \,|\, \hat{\theta}_k, \Delta_k] \geq$ constant > 0 for all k. This leads to $\text{var}[\hat{g}_{ki}(\hat{\theta}_k) \,|\, \hat{\theta}_k]$ being $O(1)$ under CRNs, but $O(c_k^{-2})$ under non-CRNs. (Because $a_k \to 0$ sufficiently fast, the SPSA iterate $\hat{\theta}_k$ converges under non-CRNs as $k \to \infty$ despite this diverging $O(c_k^{-2})$ variance.) This difference is indicative of the benefits of CRNs in forming the gradient estimate, and, in fact, is one example of a more general property of gradient estimates under CRNs (see the discussion on asymptotics below). ❑

The above results focus on finite-sample analysis of CRNs. If one considers *asymptotic* performance, the benefits of CRNs occur without the need to invoke the monotonicity and inverse transform restrictions. These benefits follow from $c_k \to 0$, and the resulting negligible distinction between the two inputs $Q(\hat{\theta}_k + c_k\Delta_k, V)$ and $Q(\hat{\theta}_k - c_k\Delta_k, V)$. There exists theory to formalize these benefits. The theory pertains to the asymptotic distribution of the iterate $\hat{\theta}_k$, as we now discuss.

The convergence conditions for FDSA and SPSA with a gradient estimate based on CRNs are, not surprisingly, very similar to those in Sections 6.4–6.5 and 7.3–7.4 (see L'Ecuyer and Yin, 1998; Kleinman et al., 1999). Suppose that each iteration uses a common value for the V input in the Q values forming the gradient estimate. For example, with SPSA, one has $V_k = V_k^{(+)} = V_k^{(-)}$ by picking the same random number seed in generating the uniform random variables that are transformed to generate $V_k^{(+)}$ and $V_k^{(-)}$ (subject to $V_k^{(+)}$ and $V_k^{(-)}$ being used in the same way—the comparability condition—as discussed at the beginning of the section). L'Ecuyer and Yin (1998) showed that CRNs increased the best asymptotic rate of stochastic convergence for FDSA from the standard $O(1/k^{1/3})$ rate discussed in Section 6.5 to the same $O(1/\sqrt{k})$ rate achieved in the gradient-based version of the root-finding SA algorithm (Section 4.4). Hence the use of CRNs compensates in some sense for the fact that only function measurements (versus gradients) are being used.

Kleinman et al. (1999) show that SPSA can also achieve the $O(1/\sqrt{k})$ rate with CRNs. This is achieved via an asymptotic normality result similar to Section 7.4 for basic SPSA. Recall the standard gain forms: $a/(k+1+A)^{\alpha}$ and $c_k = c/(k+1)^{\gamma}$ with a, c, α, and γ all strictly positive and $A \geq 0$. Under conditions identical to those in Section 7.4, except that $4\gamma - \alpha > 0$ (versus $3\gamma - \alpha/2 \geq 0$ and $\alpha - 2\gamma > 0$ in Section 7.4), SPSA with CRNs satisfies

$$k^{\alpha/2}(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*) \xrightarrow{\text{dist}} N(\mathbf{0}, \boldsymbol{\Sigma}_{\text{CRN}}), \tag{14.15}$$

where $\boldsymbol{\Sigma}_{\text{CRN}}$ is a covariance matrix that resembles $\boldsymbol{\Sigma}_{\text{SP}}$ as cited in Section 7.4. Because $\alpha \leq 1$, the maximum rate of convergence is $O(1/\sqrt{k})$. Unlike the SPSA asymptotic normality in Section 7.4—but like the root-finding SA algorithm result in Section 4.4—the mean of the asymptotic distribution is $\mathbf{0}$. The formerly optimal gain values ($\alpha = 1$, $\gamma = 1/6$) are not allowed here by the condition $4\gamma - \alpha > 0$; rather, $\gamma > 1/4$ with the asymptotically optimal $\alpha = 1$. The increase in rate of convergence is driven by a reduction of the variance of the elements in $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ from $O(c_k^{-2})$ to $O(1)$. Examples 14.6 and 14.7 conceptually or analytically illustrated this variance reduction in two problems. The two-part study in Examples 14.8 and 14.9 shows specific *numerical* improvement in a particular problem.

Example 14.8—Numerical illustration of CRNs versus non-CRNs for SPSA. Consider the loss function

$$L(\boldsymbol{\theta}) = \boldsymbol{\theta}^T\boldsymbol{\theta} + E\left[\sum_{i=1}^{p} \exp(-\boldsymbol{\theta}_i t_i)\right], \tag{14.16}$$

where $\theta = [t_1, t_2, \ldots, t_p]^T$, the components satisfy $t_i \geq 0$ for all $i = 1, 2, \ldots, p$, and the $\mathcal{V}_i$ are independently generated, exponentially distributed random variables with means $1/\lambda_i$, $i = 1, 2, \ldots, p$. This loss function is used in an example in Kleinman et al. (1999). In SPSA, we constrain the elements of $\hat{\theta}_k$ and $\hat{\theta}_k \pm c_k\Delta_k$ to be nonnegative. (Because constraints on $\hat{\theta}_k \pm c_k\Delta_k$ interfere with the "natural" simultaneous perturbation process, the gradient estimate may not have the standard $O(c_k^2)$ bias. This only affects the early iterations because once the elements of $\hat{\theta}_k$ are safely away from zero and c_k is sufficiently small, elements of $\hat{\theta}_k \pm c_k\Delta_k$ will also automatically be nonnegative.) Using the relatively simple loss function above in lieu of a complex loss based on a full Monte Carlo simulation allows for a computationally efficient study and complete control over the implementation of CRNs. Hence, the measured loss $Q(\theta, V) = \theta^T\theta + \sum_{i=1}^{p} \exp(-\mathcal{V}_i t_i)$ may be viewed as a proxy for a large simulation output where $V = [\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_p]^T$.

Suppose that $p = 10$ and that the exponentially distributed $\mathcal{V}_i$ are generated according to the inverse-transform method (Appendix D). The distributional parameters λ_i and the corresponding elements within θ^* are given in Table 14.2 (the λ_i were chosen according to a uniform distribution between 0.5 and 2.0; these values are taken from Kleinman et al., 1999). The θ^* values are found by use of a deterministic optimization method applied to the equivalent analytical form for the loss in (14.16): $L(\theta) = \theta^T\theta + \sum_{i=1}^{10} \lambda_i/(\lambda_i + t_i)$ (Exercise 14.8). Of course, θ^* is only known because of the simplicity of the "simulation" represented by Q; in practice, θ^* is not known prior to the stochastic search process. Note that the $\mathcal{V}_i$ are independent and that Q is monotonically

Table 14.2. Values of distribution parameters λ_i and elements of θ^*.

i	λ_i	ith element of θ^*
1	1.1025	0.286
2	1.6945	0.229
3	1.4789	0.248
4	1.9262	0.211
5	0.7505	0.325
6	1.3267	0.263
7	0.8428	0.315
8	0.7247	0.327
9	0.7693	0.323
10	1.3986	0.256

nonincreasing for an increasing $\mathcal{V}_i$ for any value of $t_i \geq 0$. Because of the constraints on the elements of $\hat{\theta}_k$ and $\hat{\theta}_k \pm c_k\Delta_k$ to be nonnegative, Proposition 14.2 indicates that CRNs produce the minimum variance for the components of $\hat{g}_k(\hat{\theta}_k)$.

Let us use the above values to compare the CRN and non-CRN results with SPSA based on a total of n iterations (i.e., the terminal value of the iterate is $\hat{\theta}_n$). Consider the standard gain sequence forms, $a_k = a/(k+1)^\alpha$ and $c_k = c/(k+1)^\gamma$, $0 \leq k \leq n-1$. For both CRN and non-CRN, let $a = 0.7$, $c = 0.5$. For the CRN case, $\alpha = 1$ and any $1/4 < \gamma < 1/2$ are asymptotically optimal (Kleinman et al. 1999; see also the above-mentioned condition $4\gamma - \alpha > 0$); we use $\alpha = 1$ and $\gamma = 0.49$. For the non-CRN case, the asymptotically optimal $\alpha = 1$ and $\gamma = 1/6$ are used (see Section 7.4). An initial condition $\hat{\theta}_0 = [1, 1, \ldots, 1]^T$ is employed for all runs. Table 14.3 presents the results of the study; the loss values and normalized parameter estimation errors represent the sample mean of the 50 terminal values based on 50 replications at the indicated number of iterations. The values of $L(\hat{\theta}_n)$ in the table were computed based on the above-mentioned analytical form for the loss function.

We can also compute P-values associated with the comparison of the CRN and non-CRN results. Consider the terminal loss values $L(\hat{\theta}_n)$ as the basis for the comparisons. Consider a statistical test with a null hypothesis that the mean terminal L value with CRNs is no different than the mean terminal L value for non-CRNs. The terminal losses for the two samples (CRN and non-CRN) were generated independently. Hence, we use an unmatched pairs t-test with nonidentical variance, as discussed in Appendix B. (While the $L(\hat{\theta}_n)$ values from the 50 replications are not likely to represent a sample from a normal

Table 14.3. Comparison of CRN and non-CRN terminal loss values and parameter estimate errors with SPSA. Values shown are the sample means of the loss and the sample means of the norm of the error in θ estimate from 50 replications ($L(\hat{\theta}_0) = 15.3025$; $L(\theta^*) = 8.72266$).

Total iterations n	CRN		Non-CRN	
	$L(\hat{\theta}_n)$	$\frac{\|\hat{\theta}_n - \theta^*\|}{\|\hat{\theta}_0 - \theta^*\|}$	$L(\hat{\theta}_n)$	$\frac{\|\hat{\theta}_n - \theta^*\|}{\|\hat{\theta}_0 - \theta^*\|}$
100	9.5222	0.2581	9.7000	0.2931
1000	8.7268	0.02195	8.7354	0.04103
10,000	8.7230	0.00658	8.7253	0.01845
100,000	8.7227	0.00207	8.7232	0.00819

distribution, the reasonably large sample size of 50 for each of the two samples provides some justification for the t-test, as discussed in Appendix B.) For n = 1000, 10,000, and 100,000, the P-values for the null hypothesis of equality between CRN and non-CRN are less than 10^{-10}, indicating strong rejection of the null hypothesis in favor of the alternative hypothesis that CRN produces lower terminal loss values. For n = 100, the mean loss for CRN is lower than for non-CRN, but the large two-sided P-value of 0.63 does not indicate statistical significance of this advantage. The power of CRN strongly manifests itself *after* the algorithm reaches 100 iterations. ❑

Example 14.9—Rate of convergence analysis for CRN and non-CRN results in Example 14.8. Figure 14.2 plots the means of the 50 values of $\left\|n^{\beta/2}(\hat{\theta}_n - \theta^*)\right\|$ for $\beta = 1$ or $\beta = 2/3$, as a function of n = 100, 1000, 10,000, and 100,000. In the CRN case, the magnitudes of the norms stabilize nicely at $\beta = 1$, consistent with the asymptotic normality result in (14.15) that shows a stochastic rate of convergence for the θ estimate that is $O(1/\sqrt{n})$ at the selected values of α and γ. In the non-CRN case, the magnitudes *grow* with increasing n at $\beta = 1$ (beginning with n = 1000), but are nicely bounded at $\beta = 2/3$; this is consistent with the slower rate of convergence of $O(1/n^{1/3})$ for basic SPSA, as given in Section 7.4. ❑

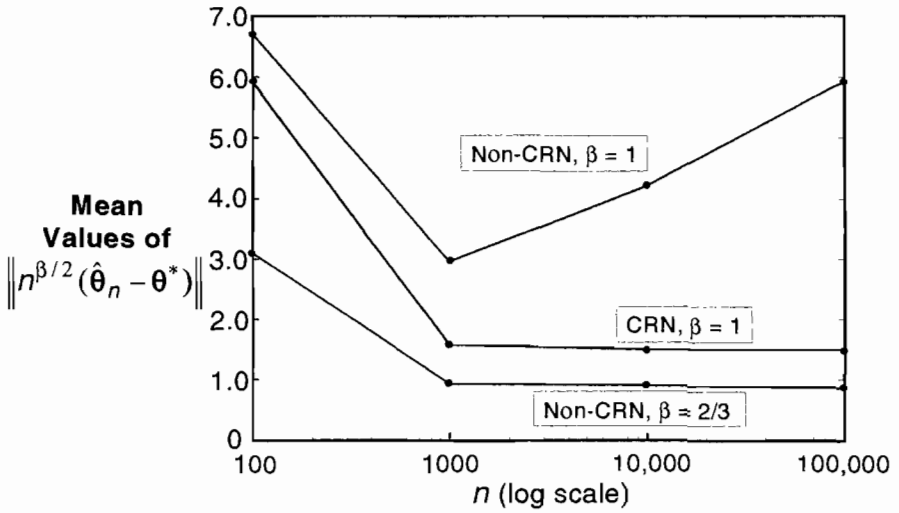

Figure 14.2. Sample means of $\left\|n^{\beta/2}(\hat{\theta}_n - \theta^*)\right\|$ for $\beta = 1$ or 2/3. Stochastic rate of convergence for CRN is $O(1/\sqrt{n})$ and for non-CRN is $O(1/n^{1/3})$, consistent with the indicated two curves that are essentially flat for large n. The remaining growing curve shows that non-CRN values are diverging when multiplied by scale factor that is too large ($\sqrt{n}$). (No curve is shown for CRN with $\beta = 2/3$; such a curve would decay to zero.)

14.4.3 Partial Common Random Numbers for Finite Samples

A difficulty in applying CRNs is that even when initializing both simulations with the same seed, the difference in the two θ values used, $\hat{\theta}_k \pm c_k\Delta_k$, may cause the simulations to process the sequences of Monte Carlo variables $V_k^{(\pm)}$ in different ways. In particular, the use of the random numbers in the two simulations may not be synchronized. This issue is especially relevant in the finite-sample ($k << \infty$) setting, where c_k may be relatively large (so the two design values $\hat{\theta}_k \pm c_k\Delta_k$ may be relatively far apart). This lack of synchronization was suggested in the context of case (ii) of the plant layout problem in Example 14.5, where the dependence of part arrivals on θ may alter the number of parts in the system.

More generally, it is easy to see how the simulations may become unsynchronized by considering a queuing system. For example, suppose that θ represents parameters associated with service rates in a network and V represents the (random) length of service times for the arrivals into the system. Then, if the simulation associated with $\hat{\theta}_k + c_k\Delta_k$ allows for faster service than the simulation associated with $\hat{\theta}_k - c_k\Delta_k$, it is possible that more arrivals will be processed in the $\hat{\theta}_k + c_k\Delta_k$ simulation. Hence, $V_k^{(+)}$ contains elements (service times) not present in $V_k^{(-)}$. More generally, if V represents many random effects (arrivals, services, etc.), then the processing of extra customers in one simulation will not only make the number of elements in the two V vectors different, but will generally make the definitions of ordinally corresponding elements different. In particular, the mth element of $V_k^{(+)}$ generally has a different physical meaning than the mth element of $V_k^{(-)}$.

Kleinman et al. (1999) discuss the implications of the above based on the notion of *partial CRNs*. In this discussion of partial CRNs, we focus on the SPSA formulation, but analogous ideas apply in the FDSA approach. This approach is built from the observation that in the early iterations, when c_k is relatively large, the θ values used in the two simulations (i.e., $\hat{\theta}_k \pm c_k\Delta_k$) will differ by a greater amount, increasing the likelihood that the two random streams $V_k^{(\pm)}$ will not be synchronized. However, in the later iterations, when c_k is small, there is little difference in the two θ values. Hence, it is more likely that the sequences $V_k^{(\pm)}$ will be synchronized. In fact, since $c_k \to 0$, the sequences are guaranteed in the limit to be fully synchronized if the simulation output is a continuous function of θ (e.g., if $Q(\theta, V)$ is continuous in θ and V and its distribution function do not depend on θ). This reminds us that the *asymptotic* distribution theory mentioned above for CRNs, expression (14.15), is applicable even in cases where the synchronization is not automatically true at two given parameter values $\hat{\theta}_k \pm c_k\Delta_k$ for a specific k.

What can one expect when the simulation budget does not allow k to get large enough so that c_k is small enough to have full synchronization of $V_k^{(\pm)}$?

Kleinman et al. (1999) consider this situation. The analysis of the partial CRN setting proceeds by decomposing $Q(\theta, V)$ into two parts, the first corresponding to the sequence of events where the random elements in $V_k^{(\pm)}$ are synchronized and the second where they are unsynchronized. This decomposition applies when $Q(\theta, V)$ is naturally defined in terms of a sum of events occurring over a period (e.g., the sum of the wait times for all customers in a queuing system over a specified period). The decomposition is not typically applicable when the simulation output is only available in toto at the end of the simulation run (say, a war gaming situation where the criterion is whether a battle is won or lost, which is known only after a simulation is completed).

As an example of applicability of partial CRNs, suppose in the queuing problem that the period being simulated represents two hours of a network's operations. In the first hour, the difference in the service rates governed by $\hat{\theta}_k \pm c_k\Delta_k$ does not cause any of the random elements in the two sequences $V_k^{(\pm)}$ to be used differently. At the beginning of the second hour, an extra arrival is processed in the $\hat{\theta}_k + c_k\Delta_k$ simulation (relative to the $\hat{\theta}_k - c_k\Delta_k$ simulation), causing all subsequent uses of $V_k^{(+)}$ to be unsynchronized with $V_k^{(-)}$. Intuitively, one might expect some benefits since there should be less variability for at least some of the input into the difference $y_k^{(+)} - y_k^{(-)} = Q(\hat{\theta}_k + c_k\Delta_k, V_k^{(+)}) - Q(\hat{\theta}_k - c_k\Delta_k, V_k^{(-)})$. This can be formalized for general problems by writing

$$\begin{aligned} y_k^{(+)} - y_k^{(-)} &= Q_1(\hat{\theta}_k + c_k\Delta_k, V_{k_1}^{(+)}) + Q_2(\hat{\theta}_k + c_k\Delta_k, V_{k_2}^{(+)}) \\ &\quad - Q_1(\hat{\theta}_k - c_k\Delta_k, V_{k_1}^{(-)}) - Q_2(\hat{\theta}_k - c_k\Delta_k, V_{k_2}^{(-)}) \\ &= \underbrace{Q_1(\hat{\theta}_k + c_k\Delta_k, V_{k_1}) - Q_1(\hat{\theta}_k - c_k\Delta_k, V_{k_1})}_{\text{synchronized part}} \\ &\quad + \underbrace{Q_2(\hat{\theta}_k + c_k\Delta_k, V_{k_2}^{(+)}) - Q_2(\hat{\theta}_k - c_k\Delta_k, V_{k_2}^{(-)})}_{\text{unsynchronized part}}, \end{aligned} \tag{14.17}$$

where the subscripts 1 or 2 denote the first (synchronized) or second (unsynchronized) part as defined above and $V_{k_1} = V_{k_1}^{(+)} = V_{k_1}^{(-)}$. With such a decomposition, it is possible to show that the variance of the gradient estimate $\hat{g}_k(\hat{\theta}_k)$ is reduced over that from a standard (independent random number) implementation. As in the full CRNs case above, this guarantee follows when $Q_1(\hat{\theta}_k + c_k\Delta_k, V)$ and $Q_1(\hat{\theta}_k - c_k\Delta_k, V)$ are both monotonic in the same direction in each component of V (Exercise 14.10). Hence, there is some benefit in attempting to synchronize the simulation input as much as possible, which is usually achieved via the simple method of using the same random number seed for the two simulations entering the gradient approximation.

Kleinman et al. (1999) show some numerical results associated with partial CRNs for the relatively simple $p = 10$ exponential problem in Example 14.8. This problem is designed so that it is possible to implement full CRNs (not always possible in practice, of course!). As expected, the error improves as we move from the independent random number case to the partial case and then to the full CRNs case. Kleinman et al. (1999) use the initial condition and SPSA gain settings of Example 14.8. In the study of this reference, the sample mean (100 replications) of the normalized distance between $\boldsymbol{\theta}^*$ and the terminal $\hat{\boldsymbol{\theta}}_k$ at $k = 10{,}000$ reduces from 0.019 in the independent random number case (no CRNs or partial CRNs) to 0.0071 in the partial CRNs setting and to 0.0065 in the full CRNs setting.

14.5 SELECTION METHODS FOR OPTIMIZATION WITH DISCRETE-VALUED $\boldsymbol{\theta}$

Frequently, simulations are used to choose the best system design among several candidate designs. This can be cast into a problem of simulation-based optimization over a discrete search space. For example, in the plant layout example of Section 14.1, it may be that each of the machines can only be placed at a small set of specified locations, perhaps constrained to this set of locations due to the need to meet safety regulations and other restrictions on the flow of people and material on the factory floor. This section is a summary of certain techniques that have been used when the elements of $\boldsymbol{\theta}$ can take on only discrete values. If, say, each of p machines can be placed at one of M locations, then the search domain Θ has M^p values to be considered.

The simplest discrete setting, of course, is when Θ contains only a small number of elements. Here, relatively simple statistical methods can be used to estimate the best option. Even in this relatively simple case, however, it may require significant effort to reliably determine $\boldsymbol{\theta}^*$, the fundamental reason being that we do not (typically) observe L at each candidate solution. Rather, a noisy measurement of L is observed as a consequence of the Monte Carlo aspect of the simulation. So it is usually necessary to use some type of averaging procedure and associated statistical tests to identify an optimal value of $\boldsymbol{\theta}$.

In cases where the search space Θ contains many options, the relatively simple averaging ideas above are usually not feasible. Here, we are driven toward other methods that offer the prospect of at least an approximately optimal solution in a reasonable time. Section 2.3 on random search discussed some methods for coping with noise; Andradóttir (1998) discusses some others. All of these methods are based on the principle of moving iteratively from one feasible solution to a neighboring feasible solution in the discrete space. The difference in the algorithms rests on the mechanics of moving from one point to the next and the means by which the value of a new point is compared with that of the old point.

Aside from the basic random search algorithms above, Chapters 8–10 show that simulated annealing (SAN) and evolutionary algorithms can be used for discrete optimization. Recall, however, that standard SAN or evolutionary algorithms are not designed to cope with *noisy* loss measurements. Some work has been done for the noisy case, especially with a discrete search space (e.g., Gelfand and Mitter, 1989; Fox and Heine, 1995). Andradóttir (1998) also presents a variant of SAN for the simulation-based discrete optimization case. Section 7.7 mentioned that SPSA—which *is* explicitly designed for the case of noisy function measurements—can sometimes be used for discrete optimization. Ahmed and Alkhamis (2002) embed a selection method of the type in this section *within* SAN to cope with the randomness.

To address the problem of choosing the best option from among a relatively small number of possibilities, we return to the theme of Chapter 12 on statistical comparisons. *Relatively small* typically means up to 20 options (Goldsman and Nelson, 1998), although there is no hard upper bound to the number of options (see Nelson et al., 2001, for extensions to a greater number of options). In particular, we summarize a statistical comparisons approach that provides statistical guarantees on the likelihood of choosing the best option or an option that is close to the best option. The approach here, which is based on the notion of an indifference zone, is certainly not the only useful approach in simulation-based optimization via statistical comparisons. The reader is directed to Goldsman and Nelson (1998) or Swisher and Jacobson (1999) for a broader survey of related techniques. Although this section is largely self-contained, the reader might find it useful to skim Chapter 12, especially Section 12.1, for some motivational material. The general approach described below is one of the most popular means for selecting among competing alternatives. The approach is also well suited to exploit common random numbers.

Suppose that the aim is to find a unique $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{\theta})$, where $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$ represents the K possible values for $\boldsymbol{\theta}$. Hence, Θ is a set of discrete options. As in Chapter 12, let $\bar{L}_i$ represent a sample mean of measured values of L at $\boldsymbol{\theta} = \boldsymbol{\theta}_i$. In particular, let $y_k(\boldsymbol{\theta}_i) = L(\boldsymbol{\theta}_i) + \varepsilon_{ki}$ represent the kth measurement of L at the given $\boldsymbol{\theta}_i$, $k = 1, 2, \ldots, n_i$ (say), and ε_{ki} represent the mean-zero noise term. For our purposes here, each $y_k(\boldsymbol{\theta}_i)$ represents the output $Q(\cdot)$ of one simulation run where the simulation parameter vector $\boldsymbol{\theta}$ has been set to $\boldsymbol{\theta}_i$. Then, $\bar{L}_i = n_i^{-1} \sum_{k=1}^{n_i} y_k(\boldsymbol{\theta}_i)$. Note that when $K = 2$, the standard two-sample t-tests from basic statistics (Appendix B) can be used to statistically determine whether $\boldsymbol{\theta}_1$ or $\boldsymbol{\theta}_2$ is preferred via whether $\bar{L}_1$ is lower than $\bar{L}_2$—or vice versa—in a statistically significant sense. The main interest here is the more difficult *multiple comparisons* case, where $K \geq 3$.

Recall from Section 12.5 that the indifference zone approaches are guaranteed with probability at least $1 - \alpha$ to provide a solution equal to $\boldsymbol{\theta}^*$ when the true loss value at $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}^*$ is at least δ units greater than $L(\boldsymbol{\theta}^*)$. The term *indifference* arises because the user is willing to tolerate any solution $\boldsymbol{\theta}_i$ such that the loss $L(\boldsymbol{\theta}_i)$ is in the range $[L(\boldsymbol{\theta}^*), L(\boldsymbol{\theta}^*) + \delta)$, called the *indifference zone*. As

δ gets smaller, the user has more stringent requirements, requiring a greater amount of sampling to form the sample means $\bar{L}_i$ as a way of distinguishing the difference between the candidate loss values $L(\boldsymbol{\theta}_i)$.

The selection process is based on picking the $\boldsymbol{\theta}_i$ associated with the lowest $\bar{L}_i$ as best. Let the random event "correct selection" correspond to correctly selecting the unique value of $\boldsymbol{\theta}_i$ equal to $\boldsymbol{\theta}^*$ based on a set of data (simulation runs). Formally, an indifference zone estimate satisfies the following relationship:

$$P(\text{correct selection}) \geq 1 - \alpha$$
$$\text{whenever } L(\boldsymbol{\theta}_i) - L(\boldsymbol{\theta}^*) \geq \delta \text{ for all } \boldsymbol{\theta}_i \neq \boldsymbol{\theta}^*$$

(Rinott, 1978). Although it is always possible that the analyst will make an incorrect selection, the expression above provides a guarantee that a correct selection is highly likely when $L(\boldsymbol{\theta}_i)$ is at least δ units larger than $L(\boldsymbol{\theta}^*)$ for $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}^*$. There are no such probabilistic guarantees if there are suboptimal $\boldsymbol{\theta}_i$ having a loss value within δ of $L(\boldsymbol{\theta}^*)$ (hence the term *indifference*!).

As with most other techniques we have seen, there are variations in implementation. Initial results for indifference zone selection assumed *known* variances for the measurements (e.g., Bechhofer, 1954). Later results—as presented here—consider the more practical setting where the variances are unknown and must be estimated (e.g., Rinott, 1978). We present below two versions of a general two-stage approach that seems to be the most useful in typical simulation applications. In both versions, the first stage is directed at estimating the variances of the measurements $y_k(\boldsymbol{\theta}_i)$ for all i. These variances are used to determine the sample size needed to meet the indifference zone requirement. The second stage is used to complete the analysis based on collecting additional measurements. In general, the approach does not require that the (unknown) variances be identical across candidate solutions $\boldsymbol{\theta}_i$.

The two step-by-step techniques below represent the two versions of the general two-stage indifference zone approach. The first technique assumes independent simulations, while the second uses common random numbers. Both procedures assume that the $y_k(\boldsymbol{\theta}_i)$ (equivalently, the ε_{ki}) are normally distributed. At first glance, this normality assumption may seem restrictive in simulation problems. However, to the extent that the simulation output is the average of many "small" random effects, central limit theory (Subsection C.2.4 in Appendix C) suggests that the output will be nearly normally distributed. Likewise, if each $y_k(\boldsymbol{\theta}_i)$ represents the average of several independent replications of a simulation, then $y_k(\boldsymbol{\theta}_i)$ will be approximately normally distributed. Nelson and Goldsman (2001) discuss this further.

The independent sampling technique is introduced in Rinott (1978). The CRN-based technique is described in Nelson and Matejcik (1995).

Two-Stage Indifference Zone Selection with Independent Sampling

Step 0 **(Initialization)** Choose δ, α, and the sample size for the first stage n_0.

Step 1 **(Beginning of first stage)** At each $\boldsymbol{\theta}_i$, run the simulation n_0 times, collecting an i.i.d. sample of size n_0 for each of the candidate $\boldsymbol{\theta}_i$. The simulations should also be independent across the $\boldsymbol{\theta}_i$ (so all Kn_0 required simulation runs should be independent).

Step 2 For each $\boldsymbol{\theta}_i$, compute the sample variance from the appropriate n_0 measurements in step 1 (using the standard s^2 form in Appendix B). Denote the sample variances by s_i^2, $i = 1, 2, \ldots, K$.

Step 3 Find the value h from Table 8.3 in Goldsman and Nelson (1998) (or equivalent table in, say, Wilcox, 1984). Compute the total sample sizes n_i for each $\boldsymbol{\theta}_i$, including both the first and second stages of testing:

$$n_i = \max\left\{n_0, \left\lceil \frac{h^2 s_i^2}{\delta^2} \right\rceil\right\}, \tag{14.18}$$

where $\lceil \cdot \rceil$ means to round up to the next integer.

Step 4 **(Beginning of second stage)** At each $\boldsymbol{\theta}_i$, run the simulation $n_i - n_0$ times, collecting an i.i.d. sample. This sample should be independent of the initial n_0 measurements and independent of measurements for other values of $\boldsymbol{\theta}$.

Step 5 Use the combined measurements collected in steps 1 and 4 to form the sample means $\bar{L}_i = n_i^{-1} \sum_{k=1}^{n_i} y_k(\boldsymbol{\theta}_i)$ for $i = 1, 2, \ldots, K$.

Step 6 Select the $\boldsymbol{\theta}_i$ corresponding to the lowest $\bar{L}_i$ as the best. It is known that the probability of this $\boldsymbol{\theta}_i$ corresponding to $\boldsymbol{\theta}^*$ is at least $1 - \alpha$ whenever $L(\boldsymbol{\theta}_j) - L(\boldsymbol{\theta}^*) \geq \delta$ for all $\boldsymbol{\theta}_j \neq \boldsymbol{\theta}^*$.

We now present the related indifference zone technique based on CRNs. The advantage of this technique is that for a given indifference zone (δ) and probability of selection ($1 - \alpha$), smaller sample sizes (i.e., fewer simulation runs) can provide the same performance guarantees as the independent sampling above. The smaller required sample sizes follow from the reduced variability provided by CRNs in comparing different scenarios. The CRN procedure is implemented at each k by having the simulation runs that produce $y_k(\boldsymbol{\theta}_1)$, $y_k(\boldsymbol{\theta}_2), \ldots, y_k(\boldsymbol{\theta}_K)$ use the same random number seed and satisfy the synchronicity requirements discussed in Section 14.4. The common seed is changed at each k.

It was noted above that the indifference zone procedures are based on an assumption that the $y_k(\boldsymbol{\theta}_i)$ are normally distributed. To ensure the validity of the CRN procedure, Nelson and Matejcik (1995) introduce the additional assumption that the covariance matrix of the vector of correlated measurements $[y_k(\boldsymbol{\theta}_1), y_k(\boldsymbol{\theta}_2), \ldots, y_k(\boldsymbol{\theta}_K)]^T$ has a structure called *sphericity*. We will not consider the full generality of sphericity, but note that one important special case is that at each k,

$\text{var}[y_k(\boldsymbol{\theta}_i)]$ is a constant independent of i and $\text{cov}[y_k(\boldsymbol{\theta}_i), y_k(\boldsymbol{\theta}_j)]$ is a constant independent of i and j. Nelson and Matejcik (1995) demonstrate that the CRN procedure is robust to departures from these assumptions under the usual CRN condition that $\text{cov}[y_k(\boldsymbol{\theta}_i), y_k(\boldsymbol{\theta}_j)] > 0$ for all k, i, j. Nelson and Matejcik (1995) also discuss a method based on the Bonferroni inequality (Section 12.3) that does not require the sphericity condition, but this method requires larger sample sizes to compensate for the relative lack of structure.

Two-Stage Indifference Zone Selection with CRN (Dependent) Sampling

Step 0 (**Initialization**) Choose δ, α, and the sample size for the first stage n_0.

Step 1 (**Beginning of first stage**) At each $\boldsymbol{\theta}_i$, run the simulation n_0 times, collecting an i.i.d. sample of size n_0 for each of the candidate $\boldsymbol{\theta}_i$. Use CRNs across the $\boldsymbol{\theta}_i$. That is, at each k, the simulation runs producing $y_k(\boldsymbol{\theta}_1)$, $y_k(\boldsymbol{\theta}_2)$,..., $y_k(\boldsymbol{\theta}_K)$ use the same random number seed and satisfy the synchronicity requirements; the seed is changed at each k.

Step 2 Let $M_1(i)$, $M_2(k)$, and M_3 represent the following three sample means:

$$M_1(i) = \frac{1}{n_0}\sum_{k=1}^{n_0} y_k(\boldsymbol{\theta}_i),$$

$$M_2(k) = \frac{1}{K}\sum_{i=1}^{K} y_k(\boldsymbol{\theta}_i),$$

$$M_3 = \frac{1}{Kn_0}\sum_{i=1}^{K}\sum_{k=1}^{n_0} y_k(\boldsymbol{\theta}_i) = \frac{1}{K}\sum_{i=1}^{K} M_1(i) = \frac{1}{n_0}\sum_{k=1}^{n_0} M_2(k).$$

Then compute the overall sample variance according to

$$s^2 = \frac{2\sum_{i=1}^{K}\sum_{k=1}^{n_0}\left[y_k(\boldsymbol{\theta}_i) - M_1(i) - M_2(k) + M_3\right]^2}{(K-1)(n_0-1)}. \qquad (14.19)$$

Step 3 Find the value $t = t^{(\alpha)}_{K-1,(K-1)(n_0-1)}$, where t is a critical value from the multivariate t-distribution in, say, Table 8.2 in Goldsman and Nelson (1998) (or equivalent table in, say, Hochberg and Tamhane, 1987, App. 3, Table 4). Compute the common total sample size n for each $\boldsymbol{\theta}_i$, which includes both the first and second stages of testing:

$$n = \max\left\{n_0, \left\lceil \frac{t^2 s^2}{\delta^2} \right\rceil\right\}. \qquad (14.20)$$

Step 4 (**Beginning of second stage**) At each $\boldsymbol{\theta}_i$, run the simulation $n - n_0$ times to collect an i.i.d. sample (independent of the initial n_0 measurements). Use CRNs across the $\boldsymbol{\theta}_i$.

Step 5 Use the combined measurements collected in steps 1 and 4 to form the sample means $\bar{L}_i = n^{-1}\sum_{k=1}^{n} y_k(\boldsymbol{\theta}_i)$ for $i = 1, 2, \ldots, K$.

Step 6 Select the $\boldsymbol{\theta}_i$ corresponding to the lowest $\bar{L}_i$ as the best. It is known that the probability of this $\boldsymbol{\theta}_i$ corresponding to $\boldsymbol{\theta}^*$ is at least $1 - \alpha$ whenever $L(\boldsymbol{\theta}_j) - L(\boldsymbol{\theta}^*) \geq \delta$ for all $\boldsymbol{\theta}_j \neq \boldsymbol{\theta}^*$.

Let us now present two examples of the above indifference zone procedures. The first example considers the procedure based on independent sampling while the second uses CRNs. The second example illustrates the reduction in number of simulation runs made possible by use of CRNs.

Example 14.10—Selection for $K = 4$ problem with independent sampling. Let us consider an example using the indifference zone method with independent sampling. Example 14.11 considers the same general setting with CRNs. Suppose that an analyst is evaluating four options via simulation and that the output is at least approximately normally distributed. Unknown to the analyst, suppose that the four options produce output having the distributions:

For all k: $y_k(\boldsymbol{\theta}_1) \sim N(0, 1)$, $y_k(\boldsymbol{\theta}_2) \sim N(0.25, 1)$, $y_k(\boldsymbol{\theta}_3) \sim N(0.25, 1)$, and $y_k(\boldsymbol{\theta}_4) \sim N(0.4, 1)$.

Because $y_k(\boldsymbol{\theta}_1)$ has the lowest mean (i.e., the lowest $L(\boldsymbol{\theta}_i)$), $\boldsymbol{\theta}^* = \boldsymbol{\theta}_1$. Suppose that $n_0 = 15$, $\delta = 0.25$, and $\alpha = 0.05$. Hence, only $L(\boldsymbol{\theta}_1)$ lies in the indifference zone $[L(\boldsymbol{\theta}^*), L(\boldsymbol{\theta}^*) + \delta)$. From Goldsman and Nelson (1998, Table 8.3), the appropriate critical value is $h = 3.285$. Using simulated data generated via the MATLAB **randn** generator, we obtain the samples of size n_0 and the associated sample variances s_i^2. The sample variances and the resulting total sample sizes n_i using (14.18) are as follows:

$$
\begin{aligned}
s_1^2 &= 0.912 \quad \Rightarrow \quad n_1 = 158, \\
s_2^2 &= 1.042 \quad \Rightarrow \quad n_2 = 180, \\
s_3^2 &= 0.873 \quad \Rightarrow \quad n_3 = 151, \\
s_4^2 &= 0.708 \quad \Rightarrow \quad n_4 = 123.
\end{aligned}
$$

This completes the first stage of the procedure.

Based on collecting the additional $n_i - n_0$ measurements for each of the four options, the analyst obtains the data needed for the second stage. The resulting sample means based on the combined data from the first and second stages are $\bar{L}_1 = -0.095$, $\bar{L}_2 = 0.238$, $\bar{L}_3 = 0.142$, and $\bar{L}_4 = 0.422$. Because $\bar{L}_1$ is

the lowest sample mean and the true loss functions $L(\theta_j)$ at $\theta_j \neq \theta^*$ are at least $\delta = 0.25$ unit greater than $L(\theta^*)$, the analyst knows with probability at least $1 - \alpha = 0.95$ that θ_1 corresponds to θ^*. Of course, because truth is available here, we know that θ_1 is, in fact, equal to θ^*. ❑

Example 14.11—Selection for $K = 4$ problem with CRN sampling. Suppose that the overall problem setting and n_0, δ, and α are the same as in Example 14.10. In the approach here, however, suppose that the analyst can implement the CRN-based indifference zone approach. This example illustrates that more efficient analysis is possible through a reduction in the sample size needed to achieve equivalent accuracy. Suppose that the analyst knows that the simulation outputs $y_k(\theta_i)$ have the same variance for all k, i. Further, suppose that the analyst knows that with CRNs for the simulations run at θ_1, θ_2, θ_3, and θ_4, $\text{cov}(y_k(\theta_i), y_k(\theta_j))$ does not depend on k, i, or j. Because the data are assumed normally distributed, the above-mentioned conditions for the CRN-based indifference zone method are met.

Unknown to the analyst, suppose that the four options produce output having the joint distribution

$$\begin{bmatrix} y_k(\theta_1) \\ y_k(\theta_2) \\ y_k(\theta_3) \\ y_k(\theta_4) \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0.25 \\ 0.25 \\ 0.4 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix} \right)$$

for all k. Because the variances are all unity, the off-diagonal covariances of 0.5 are equal to the correlations. Note that the means (i.e., the $L(\theta_i)$) are identical to the means in Example 14.10, indicating that $\theta^* = \theta_1$. From Goldsman and Nelson (1998, Table 8.2), the appropriate critical value is $t^{(\alpha)}_{K-1,(K-1)(n_0-1)} = t^{(0.05)}_{3,42} = 2.127$. To generate the jointly distributed data, we applied the MATLAB **sqrtm** function to the above covariance matrix, producing a matrix square root (see Appendix A). This square root is used to multiply the output from the **randn** generator. Then, the appropriate mean components are added. From the resulting data and the MATLAB M-file **var_CRNindiffzone** at the book's Web site (which implements (14.19)), it is found that $s^2 = 0.8066$. This leads to a sample size of $n = 59$ by (14.20). As a reflection of the value of CRNs, this sample size is much smaller than the sample sizes of 123 to 180 needed in the independent sampling case of Example 14.10.

The above completes the first stage of the procedure. Based on collecting the additional $n - n_0 = 44$ measurement for each of the four options, the analyst obtains the data needed for the second stage. The resulting sample means from the data in the first and second stages are $\bar{L}_1 = -0.126$, $\bar{L}_2 = 0.015$, $\bar{L}_3 = 0.329$, and $\bar{L}_4 = 0.407$. Hence, as in Example 14.10, the analyst selects θ_1 as the best

option. The analyst would know with probability at least $1 - \alpha = 0.95$ that this selection correctly corresponds to θ^*. ❑

14.6 CONCLUDING REMARKS

This chapter considered several issues that are important in using Monte Carlo simulations to optimize a process. We did not address the issue of building or validating the simulation model, as it was assumed that the simulation is a faithful representation of the process of interest. (Chapter 13 considered issues relevant to the construction of a simulation—or other—model of a process.) The use of simulations for optimizing (or at least improving) real processes is one of the most powerful applications of Monte Carlo methods. Using simulations can provide substantial cost savings over attempting to optimize the real process directly. In fact, in many applications it would be impossible to directly optimize a system by making changes to the process. Think, for example, of the problem of optimizing the operations of an airport by varying the terminal, taxiway, and runway locations. It is obviously not possible to carry out that optimization on the actual airport. There are, of course, countless other such systems.

Certainly, simulations are not a panacea for system design and optimization. There are many opportunities for abuse of the general approach if one is not aware of fundamental limitations. Many of the general issues and limitations on modeling discussed in Chapter 13 apply to simulation models. Misusing a "good" simulation or using a fundamentally flawed simulation may lead to solutions that are far from optimal and perhaps even dangerous to the operation of the associated physical system.

As abstractions of the real process, all simulations have limits of credibility. One must be aware of the limits in using the simulation for optimization and design. Such limits may result from simplifications in the mathematical representations in the simulation, restrictions following from regularity conditions for the validity of the embedded mathematical procedures, potential flaws in the pseudorandom number generator(s) (Appendix D), neglect of subtle interactions between a system and its environment, and the omission of critical inputs that perhaps are only relevant in some settings. For the validity of a simulation-based optimization of a real system, the search space (Θ) should correspond to the domain over which the simulation produces credible output.

An analyst should be aware that the slick graphics, animation, and data presentation of many modern commercial simulation packages do not automatically confer greater credibility to the output. Further, in many commercial packages, the underlying mathematical assumptions may be hidden from the user. In using a simulation to solve an optimization or other problem, there is simply no shortcut to having a sufficiently deep understanding of the real system and its relationship to the simulation.

With the limitations above in mind, the methods of this chapter are powerful means of improving processes. We discussed regenerative systems and how that structure can be exploited in simulations. Further, we discussed the use

of gradient-free methods of optimization (FDSA and SPSA) that are designed to cope with noisy measurements of the loss function. The principle of common random numbers is a powerful means for variance reduction that can be used in concert with optimization to increase the effectiveness of the simulation runs. Finally, we discussed methods for selecting the best option from several candidates based on the notion of an indifference zone. These selection methods provide statistical guarantees on the quality of the choice.

The next chapter continues the theme of simulation-based optimization, but with an emphasis on gradient-based methods. These methods are important special cases of root-finding stochastic approximation as introduced in Chapter 4 and the general stochastic gradient methods of Chapter 5.

EXERCISES

14.1 Derive the bound in (14.8) to the bias for the regenerative estimate $\hat{L}_{N_1,N_2}(\boldsymbol{\theta})$ of the loss function (14.2).

14.2 For the regenerative-process-based loss function in (14.2), show that finding $\boldsymbol{\theta}$ such that $\boldsymbol{g}_{\text{eqv}}(\boldsymbol{\theta}) = \mathbf{0}$ in (14.10) yields the same solution as $\boldsymbol{g}(\boldsymbol{\theta}) = \mathbf{0}$.

14.3 Let C_θ and ℓ_θ be independent random variables with respective density functions $p_C(c) = e^{-c}$, $c > 0$, and $p_\ell(l)$ equal to a uniform density over the interval (m, M) (i.e., $U(m, M)$), $0 < m < M$. Consider the problem of estimating $L = E(C_\theta)/E(\ell_\theta)$ using (14.5) with $N_1 = N_2 = 1$. Compute the exact bias in $\hat{L}_{N_1,N_2} = \hat{L}_{1,1}$ (normalized by the value of L) for $m = 1$, $M = 2$ and $m = 1$, $M = 3$. Contrast the two biases with the corresponding bounds found by the Kantorovich inequality in (14.7) and (14.8).

14.4 Let X and Y be scalar random variables with known distribution functions F_X and F_Y. Assume that both X and Y can be generated in Monte Carlo fashion by the inverse transform method based on an underlying uniform random variable (Appendix D). Suppose that one is going to estimate $E(X - Y)$ by repeated generation of values of X and Y.

(a) Using the inverse transform method, describe a CRN and a non-CRN means by which this estimation can be carried out.

(b) Prove using (14.12) that the CRN method will yield a variance for the estimate that is less than or equal to the variance for the non-CRN estimate. (Hint: If $f_1(\cdot)$ and $f_2(\cdot)$ are two nondecreasing functions, $E[f_1(Z)f_2(Z)] \geq E[f_1(Z)]E[f_2(Z)]$ for any random variable Z taking values in the domains of $f_1(\cdot)$ and $f_2(\cdot)$ provided that the indicated expectations exist [Tong, 1980, p. 13].)

14.5 Complete the proof of Proposition 14.2. In particular, take the stated result of Rubinstein et al. (1985) as a given and expand on the second sentence of the current sketch of the proof.

14.6 Suppose that the simulation output can be represented as a linear regression $\boldsymbol{Z} = \boldsymbol{B}\boldsymbol{\theta} + \boldsymbol{V}$, where $\boldsymbol{V}$ is a noise vector with finite covariance matrix ($\boldsymbol{B}$ is not dependent on $\boldsymbol{V}$). Given that $L(\boldsymbol{\theta}) = E[Q(\boldsymbol{\theta}, \boldsymbol{V})] = E[(\boldsymbol{Z} - \boldsymbol{d})^T(\boldsymbol{Z} - \boldsymbol{d})]$, where $\boldsymbol{d}$

is some "desired" vector, analyze the variability of $y_k^{(+)} - y_k^{(-)}$ under CRNs in a manner similar to that of Example 14.7. In particular:

(a) Write down the expression for the measurements $y_k^{(\pm)}$ (in terms of the given regression model) and isolate the specific part of $y_k^{(+)}$ and $y_k^{(-)}$ that have the opposite sign under CRNs. In practice, what tends to mitigate the effects of this opposite sign?

(b) Compare in general terms (detailed calculation not required) the value of $\text{var}[(y_k^{(+)} - y_k^{(-)}) \,|\, \hat{\theta}_k, \Delta_k]$ under CRNs and under non-CRNs (with independent noise terms $V_k^{(+)}$ and $V_k^{(-)}$). Show that the variance under CRNs is reduced even if the monotonicity assumption of Proposition 14.2 is not satisfied.

14.7 Consider the setting of Example 14.7 with $\dim(\boldsymbol{Z}) = 2$, $\boldsymbol{V} = [\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3]^T \sim N(\boldsymbol{0}, \boldsymbol{I}_3)$, $\boldsymbol{d} = \boldsymbol{0}$, and $\boldsymbol{B} = \begin{bmatrix} \mathcal{V}_1^2 & \mathcal{V}_2 \\ 0 & \mathcal{V}_3 \end{bmatrix}$. Suppose that the SPSA gradient approximation will be used conditioned on $\theta = \hat{\theta}_k = [1, 1]^T$, where the Δ_{ki} are symmetric Bernoulli ± 1 distributed.

(a) Let $c_k = 1$. For both the CRN and non-CRN case, determine by any convenient means (analytically or Monte Carlo) the covariance matrix of the gradient estimate conditional on $\theta = \hat{\theta}_k$ (this conditional covariance takes account of the randomness in the perturbations and $\boldsymbol{V}$). For the non-CRN case, the two $\boldsymbol{V}$ processes (in the two required values of $\boldsymbol{Z}$) should be independent of each other. Contrast the two covariance matrices; is one of the covariance matrices smaller in the matrix sense (see Appendix A)?

(b) Repeat part (a) except for setting $c_k = 0.1$.

(c) Comment on the relative performance of CRN and non-CRN for the two c_k values in parts (a) and (b). Include comments about the implications for CRNs in optimization.

14.8 In Example 14.8, show that $L(\theta) = \theta^T\theta + \sum_{i=1}^{10} \lambda_i/(\lambda_i + t_i)$. (Hint: It is useful in this derivation to recall that the inverse transform method is used to generate the exponentially distributed $\mathcal{V}_i$.)

14.9 For $n = 1500$ iterations in the setting of Example 14.8, compare CRN and non-CRN when they are run with the *same* gain sequence values (versus the different γ in the example). Set $a = 0.7$, $c = 0.5$, $\alpha = 1$, and $\gamma = 0.49$, as in the CRN case of Example 14.8.

(a) Establish that these are valid gain values for non-CRN SPSA (i.e., condition B.1″ and other gain conditions in Theorem 7.2 in Section 7.4 hold).

(b) From 50 replications, calculate the mean terminal loss values and the normalized θ estimates (as in Table 14.3) for the CRN and non-CRN cases. Compare the terminal loss values with the unmatched pairs two-sample t-test.

14.10 Consider the partial CRN setting of (14.17). Assume that V_{k_1} is independent of $\{V_{k_2}^{(+)}, V_{k_2}^{(-)}\}$ and that the conditions of Proposition 14.2 apply to the

synchronized input $Q_1(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k, V)$ and $Q_1(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k, V)$. Show that the partial CRN form leads to a variance reduction in the SPSA gradient approximation (versus the case with no CRNs). (Hint: Some of the same arguments in Exercise 14.5 apply here; it is sufficient to simply reference those arguments if you completed Exercise 14.5.)

14.11 For the indifference zone approach with K options, identify a bound to α such that it never makes sense to choose α larger than this bound. This bound should be less than 1.

14.12 Carry out a simulated indifference zone selection process under independent sampling and the following problem settings: $K = 5$, $\alpha = 0.05$, $\delta = 0.2$, $n_0 = 40$, and $h = 3.264$. For all k, suppose that the simulation outputs are distributed according to: $y_k(\boldsymbol{\theta}_1) \sim N(0, 1)$, $y_k(\boldsymbol{\theta}_2) \sim N(0.2, 1)$, $y_k(\boldsymbol{\theta}_3) \sim N(0.2, 1)$, $y_k(\boldsymbol{\theta}_4) \sim N(0.2, 1)$, and $y_k(\boldsymbol{\theta}_5) \sim N(0.4, 1)$.

14.13 Consider the setting and goals of Exercise 14.12 with the exception of using CRN-based sampling. In the simulated indifference zone selection process, assume that $\text{cov}[y_k(\boldsymbol{\theta}_i), y_k(\boldsymbol{\theta}_j)] = 0.5$ for all k, i, and j. The required critical value is $t^{(\alpha)}_{K-1,(K-1)(n_0-1)} = t^{(0.05)}_{4,156} = 2.17$. If you also did Exercise 14.12, contrast the sample sizes needed for the independent and CRN sampling implementations.