REFERENCES

1. J. S. Banks, J. S. Carson II, B. L. Nelson, and D. M. Nicol. *Discrete-Event System Simulation*. Prentice-Hall, Englewood Cliffs, NJ, 4th edition, 2004.
2. G. S. Fishman. *Discrete Event Simulation: Modeling, Programming, and Analysis*. Springer-Verlag, New York, 2001.
3. J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. John Wiley & Sons, New York, 1964.
4. M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*, volume I: Basics. John Wiley & Sons, New York, 1986.
5. A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 3rd edition, 2000.
6. I. Mitrani. *Simulation Techniques for Discrete Event Systems*. Cambridge University Press, Cambridge, UK, 1982.
7. T. J. Naylor, J. L. Balintfy, D. S. Burdick, and K. Chu. *Computer Simulation Techniques*. John Wiley & Sons, New York, 1966.
8. R. Y. Rubinstein and B. Melamed. *Modern Simulation and Modeling*. John Wiley & Sons, New York, 1998.

CHAPTER 4

STATISTICAL ANALYSIS OF DISCRETE-EVENT SYSTEMS

4.1 INTRODUCTION

An essential part of a simulation study is the statistical analysis of the output data, that is, the data obtained from the simulation model. In this chapter we present several important statistical techniques applied to different types of simulation models. As explained in the previous chapter, simulation models can generally be divided into *static* and *dynamic* models. In both types the behavior of the system is described by the *system state*, which, for all practical purposes, can be thought of as a finite-dimensional random vector $\mathbf{X}$ containing all the information about the system. In static models, the system state does not depend on time. The simulation of such models involves the repeated generation of the system state, and can be implemented using the algorithms in Chapter 2. In dynamic models the system state *does* depend on time, for example, $\mathbf{X}_t$ at time t. The behavior of the system is described by a discrete- or continuous-time stochastic process $\{\mathbf{X}_t\}$.

The rest of this chapter is organized as follows. Section 4.2 gives a brief introduction to point estimation and confidence intervals. Section 4.3 treats the statistical analysis of the output data from static models. Section 4.4 discusses the difference between finite-horizon and steady-state simulation for dynamic models. In Section 4.4.2 we consider steady-state simulation in more detail. Two popular methods for estimating steady-state performance measures — the batch means and regenerative methods — are discussed in Sections 4.4.2.1 and 4.4.2.2, respectively. Finally, in Section 4.5 we present the bootstrap technique.

Simulation and the Monte Carlo Method, Third Edition. By R. Y. Rubinstein and D. P. Kroese

4.2 ESTIMATORS AND CONFIDENCE INTERVALS

Suppose that the objective of a simulation study is to estimate an unknown quantity ℓ based on an estimator $\widehat{\ell}$, which is a function of the data produced by the simulation.

The common situation is when ℓ is the expectation of an output variable Y of the simulation. Suppose repeated runs of the simulation experiment produce independent copies $Y_1, \ldots, Y_N$ of Y. A commonsense estimator of ℓ is then the *sample mean*

$$\widehat{\ell} = \bar{Y} = \frac{1}{N} \sum_{i=1}^{N} Y_i \; . \tag{4.1}$$

This estimator is *unbiased*, in the sense that $\mathbb{E}[\widehat{\ell}] = \ell$. Moreover, by the law of large numbers $\widehat{\ell}$ converges to ℓ with probability 1 as $N \to \infty$. Notice that an estimator is viewed as a random variable. A particular outcome or observation of an estimator is called an *estimate* (a number), often denoted by the same letter.

In order to specify how *accurate* a particular estimate $\widehat{\ell}$ is, that is, how close it is to the actual unknown parameter ℓ, one needs to provide not only a point estimate $\widehat{\ell}$ but a confidence interval as well. To do so for the sample mean (4.1), observe that by the central limit theorem the estimator $\bar{Y}$ has approximately a $\mathsf{N}(\ell, \sigma^2/N)$ distribution, where σ^2 is the variance of Y — assuming $\sigma^2 < \infty$. Usually, σ^2 is unknown, but it can be estimated with the *sample variance*

$$S^2 = \frac{1}{N-1} \sum_{i=1}^{N} (Y_i - \bar{Y})^2 \; , \tag{4.2}$$

which (by the law of large numbers) tends to σ^2 as $N \to \infty$. Consequently, for large N, we see that $(\bar{Y} - \ell)\sqrt{N}/S$ is approximately $\mathsf{N}(0,1)$ distributed. Thus, if z_γ denotes the γ-quantile of the $\mathsf{N}(0,1)$ distribution (this is the number such that $\Phi(z_\gamma) = \gamma$, where Φ denotes the standard normal cdf; for example $z_{0.95} = 1.645$, since $\Phi(1.645) = 0.95$), then

$$\mathbb{P}\left(-z_{1-\alpha/2} \leqslant \frac{(\bar{Y} - \ell)\sqrt{N}}{S} \leqslant z_{1-\alpha/2} \right) \approx 1 - \alpha,$$

which after rearranging gives

$$\mathbb{P}\left(\bar{Y} - z_{1-\alpha/2} \frac{S}{\sqrt{N}} \leqslant \ell \leqslant \bar{Y} + z_{1-\alpha/2} \frac{S}{\sqrt{N}} \right) \approx 1 - \alpha \; .$$

In other words, an approximate $(1-\alpha)100\%$ *confidence interval* for ℓ is

$$\left(\bar{Y} \pm z_{1-\alpha/2} \frac{S}{\sqrt{N}} \right) \; , \tag{4.3}$$

where the notation $(a \pm b)$ is shorthand for the interval $(a - b, a + b)$.

Remark 4.2.1 The interpretation of a confidence interval requires some care. It is important to note that (4.3) is a *stochastic* confidence interval that contains ℓ with

probability approximately $1-\alpha$. After observing outcomes $y_1, \ldots, y_N$ of the random variables $Y_1, \ldots, Y_N$, we are able to construct a *numerical* confidence interval by replacing the $\{Y_i\}$ with the $\{y_i\}$ in (4.3). However, we can no longer claim that such an interval contains ℓ with probability approximately $1-\alpha$. This is because ℓ is a *number*, so it either lies in the numerical confidence interval with probability 1 or 0. The interpretation of a 95% numerical confidence interval such as $(1.53, 1.58)$ is thus that it is a particular outcome of a random interval that contains ℓ in 95% of the cases. If we pick at random a ball from an urn with 95 white and 5 black balls *but don't look*, we can be quite confident that the ball in our hand is in fact white. This is how confident we should be that the interval $(1.53, 1.58)$ contains ℓ.

It is common practice in simulation to use and report the *absolute and relative* widths of the confidence interval (4.3), defined as

$$w_a = 2\, z_{1-\alpha/2} \frac{S}{\sqrt{N}} \tag{4.4}$$

and

$$w_r = \frac{w_a}{\bar{Y}} \;, \tag{4.5}$$

respectively, provided that $\bar{Y} > 0$. The absolute and relative widths may be used as stopping rules (criteria) to control the length of a simulation run. The relative width is particularly useful when ℓ is very small. For example, if $\ell \approx 10^{-10}$, reporting a result such as $w_a = 0.05$ is almost meaningless, while in contrast, reporting $w_r = 0.05$ is quite meaningful. Another important quantity is the *relative error* (RE) of an estimator $\widehat{\ell}$, defined as

$$\mathrm{RE} = \frac{\sqrt{\mathrm{Var}(\widehat{\ell})}}{\mathbb{E}[\widehat{\ell}]} \;, \tag{4.6}$$

which, in the case that $\widehat{\ell} = \bar{Y}$, is equal to $\sigma/(\ell\sqrt{N})$. Note that this is equal to w_r divided by $2z_{1-\alpha/2}$ and can be estimated as $S/(\widehat{\ell}\sqrt{N})$.

EXAMPLE 4.1 Estimation of Rare-Event Probabilities

Consider estimation of the tail probability $\ell = \mathbb{P}(X \geqslant \gamma)$ of some random variable X for a *large* number γ. If ℓ is very small, then the event $\{X \geqslant \gamma\}$ is called a *rare event* and the probability $\mathbb{P}(X \geqslant \gamma)$ is called a *rare-event probability*.

We may attempt to estimate ℓ via (4.1) as

$$\widehat{\ell} = \frac{1}{N} \sum_{i=1}^{N} I_{\{X_i \geqslant \gamma\}} \;, \tag{4.7}$$

which involves drawing a random sample $X_1, \ldots, X_N$ from the pdf of X and defining the indicators $Y_i = I_{\{X_i \geqslant \gamma\}}$, $i = 1, \ldots, N$. The estimator $\widehat{\ell} = \bar{Y}$ thus defined is called the *crude Monte Carlo* (CMC) estimator. For small ℓ the relative error of the CMC estimator is given by

$$\kappa = \frac{\sqrt{\mathrm{Var}(\widehat{\ell})}}{\mathbb{E}[\widehat{\ell}]} = \sqrt{\frac{1-\ell}{N\,\ell}} \approx \sqrt{\frac{1}{N\,\ell}} \;. \tag{4.8}$$

As a numerical example, suppose that $\ell = 10^{-6}$. In order to estimate ℓ accurately with relative error (say) $\kappa = 0.01$, we need to choose a sample size

$$N \approx \frac{1}{\kappa^2 \ell} = 10^{10} .$$

This shows that estimating small probabilities via CMC estimators is computationally meaningless.

4.3 STATIC SIMULATION MODELS

As mentioned in Chapter 3, in a static simulation model the system state does not depend on time. Suppose that we want to determine the expectation

$$\ell = \mathbb{E}[H(\mathbf{X})] = \int H(\mathbf{x})\, f(\mathbf{x})\, \mathrm{d}\mathbf{x} , \tag{4.9}$$

where $\mathbf{X}$ is a random vector with pdf f, and $H(\mathbf{x})$ is a real-valued function called the *performance* function. We assume that ℓ cannot be evaluated analytically and we need to resort to simulation. The situation is exactly as described in Section 4.2, with $Y = H(\mathbf{X})$, and ℓ can be estimated with the sample mean

$$\widehat{\ell} = N^{-1} \sum_{i=1}^{N} H(\mathbf{X}_i) , \tag{4.10}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is a *random sample* from f; that is, the $\{\mathbf{X}_i\}$ are independent replications of $\mathbf{X} \sim f$.

The following algorithm summarizes how to estimate the expected system performance, $\ell = \mathbb{E}[H(\mathbf{X})]$, and how to calculate the corresponding confidence interval:

Algorithm 4.3.1: Point Estimate and Confidence Interval (Static Model)

input : Simulation method for $\mathbf{X} \sim f$, performance function H, sample size N, confidence level $1 - \alpha$.
output: Point estimate and $(1 - \alpha)$-confidence interval for $\ell = \mathbb{E}[H(\mathbf{X})]$.
1 Simulate N replications, $\mathbf{X}_1, \ldots, \mathbf{X}_N$, of $\mathbf{X}$.
2 Let $Y_i \leftarrow H(\mathbf{X}_i),\ i = 1, \ldots, N$.
3 Calculate the point estimate and a confidence interval of ℓ from (4.1) and (4.3), respectively.

We conclude with two examples where static simulation is used.

EXAMPLE 4.2 Reliability Model

Consider a system that consists of n components. The operational state of each component $i = 1, \ldots, n$ is represented by $X_i \sim \mathsf{Ber}(p_i)$, where $X_i = 1$ means that the component is working and $X_i = 0$ means that it has failed. Note that the probability that component i is working — its *reliability* — is p_i. The failure behavior of the system is thus represented by the binary random vector $\mathbf{X} = (X_1, \ldots, X_n)$, where it is usually assumed that the $\{X_i\}$ are independent. Suppose that the operational state of the system, say Y, is either

functioning or failed, depending on the operational states of the components. In other words, we assume that there exists a function $H : \mathscr{X} \to \{0, 1\}$ such that

$$Y = H(\mathbf{X}) \,,$$

where $\mathscr{X} = \{0, 1\}^n$ is the set of all binary vectors of length n.

The function H is called the *structure function* and often can be represented by a graph. In particular, the graph in Figure 4.1 depicts a *bridge network* with five components (links). For this particular model the system works (i.e., $H(\mathbf{X}) = 1$) if the black terminal nodes are connected by working links. The structure function is equal to (see Problem 4.2)

$$H(\mathbf{x}) = 1 - (1 - x_1\, x_4)\,(1 - x_2\, x_5)\,(1 - x_1\, x_3\, x_5)\,(1 - x_2\, x_3\, x_4) \,. \tag{4.11}$$

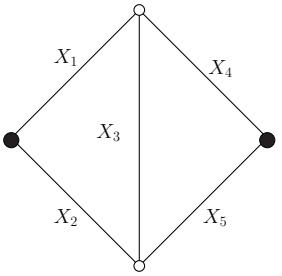

Figure 4.1: A bridge network.

Suppose that we are interested in the reliability ℓ of the general n-component system. We have

$$\begin{aligned} \ell = \mathbb{P}(Y = 1) = \mathbb{E}[H(\mathbf{X})] &= \sum_{\mathbf{x} \in \mathscr{X}} H(\mathbf{x})\, \mathbb{P}(\mathbf{X} = \mathbf{x}) \\ &= \sum_{\mathbf{x} \in \mathscr{X}} H(\mathbf{x}) \prod_{i=1}^{n} \left[p_i^{x_i}\, (1 - p_i)^{1 - x_i} \right] \,. \end{aligned} \tag{4.12}$$

For complex systems with a large number of components and with little structure, it is very time-consuming to compute the system reliability ℓ via (4.12), since this requires the evaluation of $\mathbb{P}(\mathbf{X} = \mathbf{x})$ and $H(\mathbf{x})$ for 2^n vectors $\mathbf{x}$. However, simulation of $\mathbf{X}$ and estimation of ℓ via (4.10) can still be a viable approach, even for large systems, provided that $H(\mathbf{X})$ is readily evaluated. In practice one needs substantially fewer than 2^n samples to estimate ℓ accurately.

EXAMPLE 4.3 Stochastic PERT Network

The *program evaluation and review technique* (PERT) is a frequently used tool for project management. Typically, a project consists of many activities, some of which can be performed in parallel while others can only be performed after certain preceding activities have been finished. In particular, each activity

has a list of *predecessors* that must be completed before it can start. A PERT network is a directed graph where the arcs represent the activities and the vertices represent specific milestones. A milestone is completed when all activities pointing to that milestone are completed. Before an activity can begin, the milestone from which the activity originates must be completed. An example of a precedence list of activities is given in Table 4.3; its PERT graph is given in Figure 4.2.

Table 4.1: Precedence ordering of activities.

Activity	1	2	3	4	5	6	7	8	9	10	11	12
Predecessor(s)	-	-	1	1	2	2	3	3	4,6	5,8	7	9,10

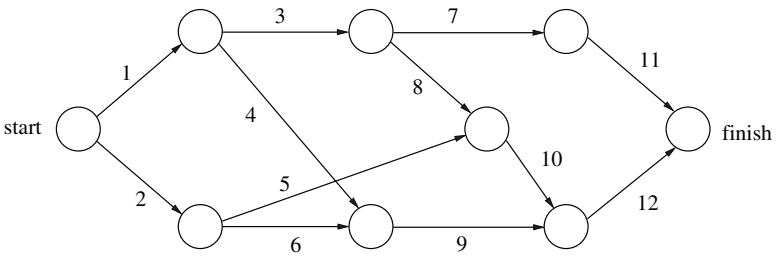

Figure 4.2: A stochastic PERT network.

Suppose that each activity i takes a random time X_i to complete. An important quantity for PERT networks is the maximal project duration, that is, the length of the longest path from start to finish — the so-called *critical path*. Suppose that we are interested in the expected maximal project duration, say ℓ. Letting $\mathbf{X}$ be the vector of activity lengths and $H(\mathbf{X})$ be the length of the critical path, we have

$$\ell = \mathbb{E}[H(\mathbf{X})] = \mathbb{E}\left[\max_{j=1,\ldots,p} \sum_{i \in \mathscr{P}_j} X_i\right] , \tag{4.13}$$

where $\mathscr{P}_j$ is the j-th complete path from start to finish and p is the number of such paths.

4.4 DYNAMIC SIMULATION MODELS

Dynamic simulation models deal with systems that evolve over time. Our goal is (as for static models) to estimate the expected system performance, where the state of the system is now described by a stochastic process $\{\mathbf{X}_t\}$, which may have a continuous or discrete time parameter. For simplicity, we mainly consider the case where $\mathbf{X}_t$ is a scalar random variable; we then write X_t instead of $\mathbf{X}_t$.

We make a distinction between *finite-horizon* and *steady-state* simulation. In finite-horizon simulation, measurements of system performance are defined relative to a specified interval of simulation time $[0, T]$ (where T may be a random variable),

while in steady-state simulation, performance measures are defined in terms of certain limiting measures as the time horizon (simulation length) goes to infinity.

The following illustrative example offers further insight into finite-horizon and steady-state simulations. Suppose that the state X_t represents the number of customers in a stable $M/M/1$ queue (see Example 1.13 on Page 27). Let

$$F_{t,m}(x) = \mathbb{P}(X_t \leqslant x \,|\, X_0 = m) \tag{4.14}$$

be the cdf of X_t given the initial state $X_0 = m$ (m customers are initially present). $F_{t,m}$ is called the *finite-horizon distribution* of X_t given that $X_0 = m$.

We say that the process $\{X_t\}$ *settles into steady state* (equivalently, that *steady state exists*) if for all m

$$\lim_{t\to\infty} F_{t,m}(x) = F(x) \equiv \mathbb{P}(X \leqslant x) \tag{4.15}$$

for some random variable X. In other words, *steady state* implies that, as $t \to \infty$, the transient cdf, $F_{t,m}(x)$ (which generally depends on t and m), approaches a steady-state cdf, $F(x)$, which *does not depend* on the initial state, m. The stochastic process, $\{X_t\}$, is said to *converge in distribution* to a random variable $X \sim F$. Such an X can be interpreted as the random state of the system when observed far away in the future. The operational meaning of *steady state* is that after some period of time the transient cdf $F_{t,m}(x)$ comes close to its limiting (steady-state) cdf $F(x)$. It is important to realize that this does *not* mean that at any point in time the realizations of $\{X_t\}$ generated from the simulation run become independent or constant. The situation is illustrated in Figure 4.3, where the dashed curve indicates the expectation of X_t.

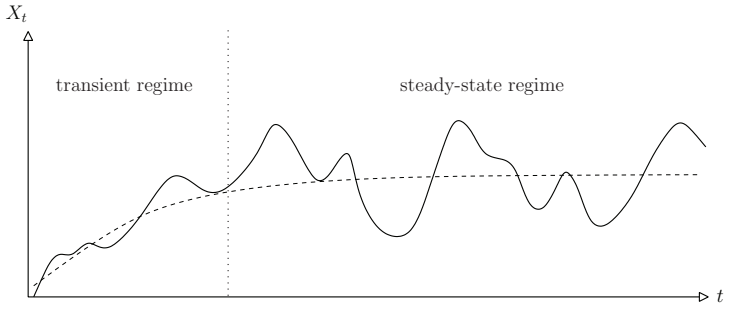

Figure 4.3: The state process for a dynamic simulation model.

The exact distributions (transient and steady-state) are usually available only for simple Markovian models such as the $M/M/1$ queue. For non-Markovian models, usually neither the distributions (transient and steady-state) nor even the associated moments are available via analytical methods. For performance analysis of such models one must resort to simulation.

Note that for some stochastic models, only finite-horizon simulation is feasible, since the steady-state regime either does not exist or the finite-horizon period is so long that the steady-state analysis is computationally prohibitive (e.g., [10]).

4.4.1 Finite-Horizon Simulation

The statistical analysis for finite-horizon simulation models is basically the same as that for static models. To illustrate the procedure, let us suppose that $\{X_t, t \geqslant 0\}$ is a continuous-time process for which we wish to estimate the expected average value,

$$\ell(T, m) = \mathbb{E}\left[T^{-1} \int_0^T X_t \, \mathrm{d}t\right] , \tag{4.16}$$

as a function of the time horizon T and the initial state $X_0 = m$. (For a discrete-time process $\{X_t, t = 1, 2, \ldots\}$, the integral $\int_0^T X_t \, \mathrm{d}t$ is replaced by the sum $\sum_{t=1}^T X_t$.) For example, if X_t represents the number of customers in a queueing system at time t, then $\ell(T, m)$ is the average number of customers in the system during the time interval $[0, T]$, given $X_0 = m$.

Assume now that N independent replications are performed, each starting at state $X_0 = m$. Then the point estimator and the $(1 - \alpha)$ 100% confidence interval for $\ell(T, m)$ can be written, as in the static case (see (4.10) and (4.3)):

$$\widehat{\ell}(T, m) = N^{-1} \sum_{i=1}^N Y_i \tag{4.17}$$

and

$$\left(\widehat{\ell}(T, m) \pm z_{1-\alpha/2} S N^{-1/2}\right) , \tag{4.18}$$

respectively, where $Y_i = T^{-1} \int_0^T X_{ti} \, \mathrm{d}t$, X_{ti} is the observation at time t from the i-th replication and S^2 is the sample variance of $\{Y_i\}$. The algorithm for estimating the finite-horizon performance, $\ell(T, m)$, is thus:

Algorithm 4.4.1: Point Estimate and Confidence Interval (Finite Horizon)

input : Simulation method for the process $\{X_t, t \geqslant 0\}$, time horizon T, initial state m, sample size N, confidence level $1 - \alpha$.
output: Point estimate and $(1 - \alpha)$-confidence interval for the expected average value $\ell(T, m)$.

1 Simulate N replications of the process $\{X_t, t \leqslant T\}$, starting each replication from the initial state $X_0 = m$.
2 Calculate the point estimator and the confidence interval of $\ell(T, m)$ from (4.17) and (4.18), respectively.

If, instead of the expected average number of customers, we want to estimate the expected *maximum* number of customers in the system during an interval $(0, T]$, the only change required is to replace $Y_i = T^{-1} \int_0^T X_{ti} \, \mathrm{d}t$ with $Y_i = \max_{0 \leqslant t \leqslant T} X_{ti}$. In the same way, we can estimate other performance measures for this system, such as the probability that the maximum number of customers during $(0, T]$ exceeds some level γ or the expected average period of time that the first k customers spend in the system.

4.4.2 Steady-State Simulation

Steady-state simulation is used for systems that exhibit some form of stationary or long-run behavior. Loosely speaking, we view the system as having started in the

infinite past, so that any information about initial conditions and starting times becomes irrelevant. The more precise notion is that the system state is described by a *stationary process*; see also Section 1.13.

EXAMPLE 4.4 $M/M/1$ Queue

Consider the birth-and-death process $\{X_t, t \geqslant 0\}$ describing the number of customers in the $M/M/1$ queue; see Example 1.13. When the traffic intensity $\varrho = \lambda/\mu$ is less than 1, this Markov jump process has a limiting distribution,

$$\lim_{t\to\infty} \mathbb{P}(X_t = k) = (1-\varrho)\varrho^k, \quad k = 0, 1, 2, \ldots,$$

which is also its stationary distribution. When X_0 is distributed according to this limiting distribution, the process $\{X_t, t \geqslant 0\}$ is stationary: it behaves as if it has been going on for an infinite period of time. In particular, the distribution of X_t does not depend on t. A similar result holds for the Markov process $\{Y_n, n = 1, 2, \ldots\}$, describing the number of customers in the system as seen by the n-th arriving customer. It can be shown that under the condition $\varrho < 1$ it has the *same* limiting distribution as $\{X_t, t \geqslant 0\}$. Note that for the $M/M/1$ queue the steady-state expected performance measures are available analytically, while for the $GI/G/1$ queue, to be discussed in Example 4.5, one needs to resort to simulation.

Special care must be taken when making inferences concerning steady-state performances. The reason is that the output data are typically correlated; consequently, the statistical analysis used above, based on independent observations, is no longer applicable.

In order to cancel the effects of the time dependence and the initial distribution, it is common practice to discard the data that are collected during the nonstationary or transient part of the simulation. However, it is not always clear when the process will reach stationarity. If the process is regenerative, then the regenerative method, discussed in Section 4.4.2.2, avoids this transience problem altogether.

From now on, we assume that $\{X_t\}$ is a stationary process. Suppose that we wish to estimate the steady-state expected value $\ell = \mathbb{E}[X_t]$, for example, the expected steady-state queue length, or the expected steady-state sojourn time of a customers in a queue. Then ℓ can be estimated as either

$$\widehat{\ell} = T^{-1} \sum_{t=1}^{T} X_t$$

or

$$\widehat{\ell} = T^{-1} \int_0^T X_t \, \mathrm{d}t \,,$$

respectively, depending on whether $\{X_t\}$ is a discrete-time or continuous-time process.

For concreteness, consider the discrete case. The variance of $\widehat{\ell}$ (see Problem 1.15) is given by

$$\mathrm{Var}(\widehat{\ell}) = \frac{1}{T^2} \left(\sum_{t=1}^{T} \mathrm{Var}(X_t) + 2 \sum_{s=1}^{T-1} \sum_{t=s+1}^{T} \mathrm{Cov}(X_s, X_t) \right) . \tag{4.19}$$

Since $\{X_t\}$ is stationary, we have $\mathrm{Cov}(X_s, X_t) = \mathbb{E}[X_s X_t] - \ell^2 = R(t-s)$, where R defines the *covariance function* of the stationary process. Note that $R(0) = \mathrm{Var}(X_t)$. As a consequence, we can write (4.19) as

$$T \,\mathrm{Var}(\widehat{\ell}) = R(0) + 2\sum_{t=1}^{T-1}\left(1 - \frac{t}{T}\right) R(t) \,. \tag{4.20}$$

Similarly, if $\{X_t\}$ is a continuous-time process, the sum in (4.20) is replaced with the corresponding integral (from $t = 0$ to T), while all other data remain the same. In many applications $R(t)$ decreases rapidly with t, so that only the first few terms in the sum (4.20) are relevant. These covariances, say $R(0), R(1), \ldots, R(K)$, can be estimated via their (unbiased) sample averages:

$$\widehat{R}(k) = \frac{1}{T-k-1}\sum_{t=1}^{T-k}(X_t - \widehat{\ell}\,)(X_{t+k} - \widehat{\ell}\,), \quad k = 0, 1, \ldots, K \,.$$

Thus, for large T the variance of $\widehat{\ell}$ can be estimated as $\tilde{S}^2/T$, where

$$\tilde{S}^2 = \widehat{R}(0) + 2\sum_{t=1}^{K}\widehat{R}(t) \,.$$

To obtain confidence intervals, we use again the central limit theorem; that is, the cdf of $\sqrt{T}(\widehat{\ell} - \ell)$ converges to the cdf of the normal distribution with expectation 0 and variance $\sigma^2 = \lim_{T\to\infty} T\,\mathrm{Var}(\widehat{\ell}\,)$ — the so-called *asymptotic variance* of $\widehat{\ell}$. Using $\tilde{S}^2$ as an estimator for σ^2, we find that an approximate $(1-\alpha)100\%$ confidence interval for ℓ is given by

$$\left(\widehat{\ell} \pm z_{1-\alpha/2}\frac{\tilde{S}}{\sqrt{T}}\right) \,. \tag{4.21}$$

Below we consider two popular methods for estimating steady-state parameters: the *batch means* and *regenerative* methods.

4.4.2.1 Batch Means Method The batch means method is most widely used by simulation practitioners to estimate steady-state parameters from a single simulation run, say of length M. The initial K observations, corresponding to the transient part of the run (called *burn-in*), are deleted, and the remaining $M - K$ observations are divided into N batches, each of length

$$T = \frac{M-K}{N} \,.$$

The deletion serves to eliminate or reduce the initial bias, so that the remaining observations $\{X_t, t > K\}$ are statistically more typical of the steady state.

Suppose we want to estimate the expected steady-state performance $\ell = \mathbb{E}[X_t]$, assuming that the process is stationary for $t > K$. We assume, for simplicity, that $\{X_t\}$ is a discrete-time process. Let X_{ti} denote the t-th observation from the i-th batch. The sample mean of the i-th batch of length T is given by

$$Y_i = \frac{1}{T}\sum_{t=1}^{T} X_{ti}, \quad i = 1, \ldots, N \,.$$

Therefore, the sample mean $\widehat{\ell}$ of ℓ is

$$\widehat{\ell} = \frac{1}{M-K} \sum_{t=K+1}^{M} X_t = \frac{1}{N} \sum_{i=1}^{N} Y_i \ . \tag{4.22}$$

The procedure is illustrated in Figure 4.4.

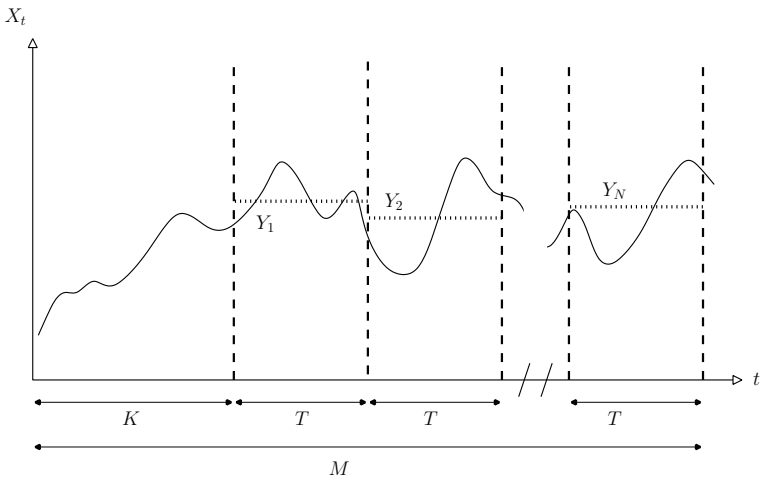

Figure 4.4: Illustration of the batch means procedure.

In order to ensure approximate independence between the batches, their size, T, should be large enough. In order for the central limit theorem to hold approximately, the number of batches, N, should typically be chosen in the range 20–30. In such a case, an approximate confidence interval for ℓ is given by (4.3), where S is the sample standard deviation of the $\{Y_i\}$. In the case where the batch means do exhibit some dependence, we can apply formula (4.21) as an alternative to (4.3).

Next, we discuss briefly how to choose K. In general, this is a very difficult task, since very few analytic results are available. The following queueing example provides some hints on how K should be increased as the traffic intensity in the queue increases.

Let $\{X_t, t \geqslant 0\}$ be the queue length process (not including the customer in service) in an $M/M/1$ queue, and assume that we start the simulation at time zero with an empty queue. It is shown in [1, 2] that in order to be within 1% of the steady-state mean, the length of the initial portion to be deleted, K, should be on the order of $8/(\mu(1-\varrho)^2)$, where $1/\mu$ is the expected service time. Thus, for $\varrho = 0.5,\ 0.8,\ 0.9$, and $0.95,\ K$ equals 32, 200, 800, and 3200 expected service times, respectively.

In general, one can use the following simple rule of thumb.

1. Define the following moving average A_k of length T:

$$A_k = \frac{1}{T} \sum_{t=k+1}^{T+k} X_t \ .$$

2. Calculate A_k for different values of k, say $k = 0, m, 2m, \ldots, rm, \ldots$, where m is fixed, say $m = 10$.

3. Find r such that $A_{rm} \approx A_{(r+1)m} \approx \cdots \approx A_{(r+s)m}$, while $A_{(r-s)m} \not\approx A_{(r-s+1)m} \not\approx \cdots \not\approx A_{rm}$, where $r \geq s$ and $s = 5$, for example.

4. Deliver $K = r\,m$.

The batch means algorithm is as follows:

Algorithm 4.4.2: Batch Means Method

input : Simulation method for $\{X_t, t \geqslant 0\}$, run length M, burn-in period K, number of batches N, confidence level $1 - \alpha$.

output: Point estimate and $(1 - \alpha)$-confidence interval for the expected steady-state performance ℓ.

1 Make a single simulation run of length M and delete the first K observations corresponding to the burn-in period.

2 Divide the remaining $M - K$ observations into N batches, each of length $T = (M - K)/N$.

3 Calculate the point estimator and the confidence interval for ℓ from (4.22) and (4.3), respectively.

EXAMPLE 4.5 $GI/G/1$ Queue

The $GI/G/1$ queueing model is a generalization of the $M/M/1$ model discussed in Examples 1.13 and 4.4. The only differences are that (1) the interarrival times each have a general cdf F and (2) the service times each have a general cdf G. Let us consider the process $\{Z_n, n = 1, 2, \ldots\}$ describing the number of people in a $GI/G/1$ queue as seen by the n-th arriving customer. Figure 4.5 gives a realization of the batch means procedure for estimating the steady-state queue length. In this example the first $K = 100$ observations are thrown away, leaving $N = 9$ batches, each of size $T = 100$. The batch means are indicated by thick lines.

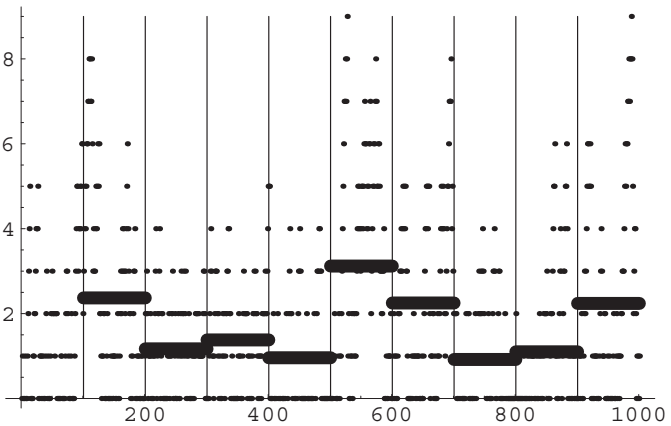

Figure 4.5: The batch means for the process $\{Z_n, n = 1, 2, \ldots\}$.

Remark 4.4.1 (Replication-Deletion Method) In the replication-deletion method, N independent runs are carried out, rather than a single simulation run as in the batch means method. From each replication, one deletes K initial observations corresponding to the finite-horizon simulation and then calculates the point estimator and the confidence interval for ℓ via (4.22) and (4.3), respectively, exactly as in the batch means approach. Note that the confidence interval obtained with the replication-deletion method is unbiased, whereas the one obtained by the batch means method is slightly biased. However, the former requires deletion from *each* replication, as compared to *a single* deletion in the latter. For this reason, the former is not as popular as the latter. For more details on the replication-deletion method, see [10].

4.4.2.2 The Regenerative Method A stochastic process $\{X_t\}$ is called *regenerative* if there exist random time points $T_0 < T_1 < T_2 < \ldots$ such that at each time point the process restarts probabilistically. More precisely, the process $\{X_t\}$ can be split into iid replicas during intervals, called *cycles*, of lengths $\tau_i = T_i - T_{i-1},\ i = 1, 2, \ldots$.

■ **EXAMPLE 4.6 Markov Chain**

The standard example of a regenerative process is a Markov chain. Assume that the chain starts from state i. Let $T_0 < T_1 < T_2 < \ldots$ denote the times that it visits state j. Note that at each random time T_n the Markov chain starts afresh, independently of the past. We say that the Markov process *regenerates* itself. For example, consider a two-state Markov chain with transition matrix

$$P = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}. \tag{4.23}$$

Assume that all four transition probabilities p_{ij} are strictly positive and that, starting from state $i = 1$, we obtain the following sample trajectory:

$$(x_0, x_1, x_2, \ldots, x_{10}) = (1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1)\ .$$

It is readily seen that the transition probabilities corresponding to the sample trajectory above are

$$p_{12},\ p_{22},\ p_{22},\ p_{21},\ p_{12},\ p_{21},\ p_{11},\ p_{12},\ p_{22},\ p_{21}\ .$$

Taking $j = 1$ as the regenerative state, the trajectory contains four cycles with the following transitions:

$$1 \to 2 \to 2 \to 2 \to 1; \quad 1 \to 2 \to 1; \quad 1 \to 1; \quad 1 \to 2 \to 2 \to 1\ ,$$

and the corresponding cycle lengths are $\tau_1 = 4,\ \tau_2 = 2,\ \tau_3 = 1,\ \tau_4 = 3$.

■ **EXAMPLE 4.7 $GI/G/1$ Queue (Continued)**

Another classic example of a regenerative process is the process $\{X_t, t \geqslant 0\}$ describing the number of customers in the $GI/G/1$ system, where the regeneration times $T_0 < T_1 < T_2 < \ldots$ correspond to customers arriving at an empty system (see also Example 4.5, where a related discrete-time process is

considered). Observe that at each time T_i the process starts afresh, independently of the past; in other words, the process regenerates itself. Figure 4.6 illustrates a typical sample path of the process $\{X_t, t \geqslant 0\}$. Note that here $T_0 = 0$; that is, at time 0 a customer arrives at an empty system.

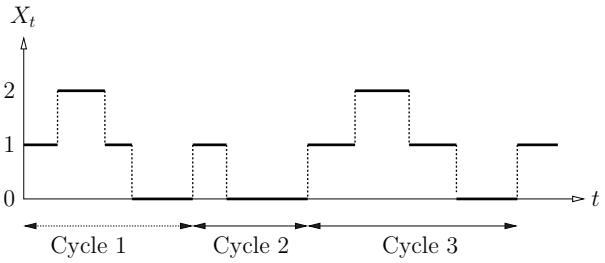

Figure 4.6: A sample path of the process $\{X_t, t \geqslant 0\}$, describing the number of customers in a $GI/G/1$ queue.

EXAMPLE 4.8 (s, S) Policy Inventory Model

Consider a continuous-review, single-commodity inventory model supplying external demands and receiving stock from a production facility. When demand occurs, it is either filled or back-ordered (to be satisfied by delayed deliveries). At time t, the *net inventory* (on-hand inventory minus back orders) is N_t, and the *inventory position* (net inventory plus on-order inventory) is X_t. The control policy is an (s, S) policy that operates on the inventory position. Specifically, at any time t when a demand D is received that would reduce the inventory position to less than s (i.e., $X_{t-} - D < s$, where X_{t-} denotes the inventory position just before t), an order of size $S - (X_{t-} - D)$ is placed, which brings the inventory position immediately back to S. Otherwise, no action is taken. The order arrives r time units after it is placed (r is called the *lead* time). Clearly, $X_t = N_t$ if $r = 0$. Both inventory processes are illustrated in Figure 4.7. The dots in the graph of the inventory position (below the s-line) represent what the inventory position would have been if no order was placed.

Let D_i and A_i be the size of the i-th demand and the length of the i-th inter-demand time, respectively. We assume that both $\{D_i\}$ and $\{A_i\}$ are iid sequences, with common cdfs F and G, respectively. In addition, the sequences are assumed to be independent of each other. Under the back-order policy and the assumptions above, both the inventory position process $\{X_t\}$ and the net inventory process $\{N_t\}$ are regenerative. In particular, each process regenerates when it is raised to S. For example, each time an order is placed, the inventory position process regenerates. It is readily seen that the sample path of $\{X_t\}$ in Figure 4.7 contains three regenerative cycles, while the sample path of $\{N_t\}$ contains only two, which occur after the second and third lead times. Note that during these times no order has been placed.

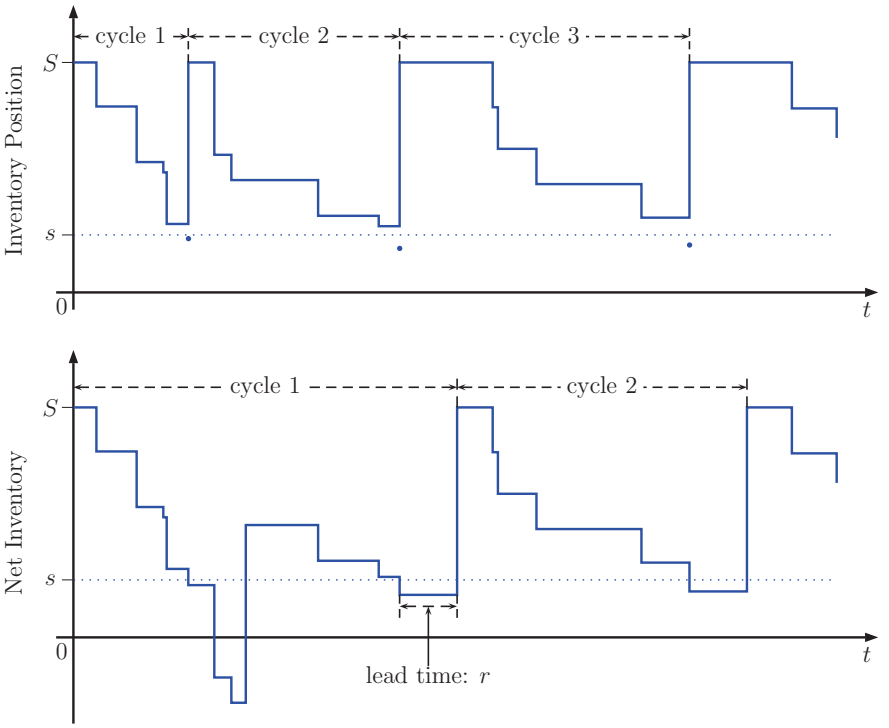

Figure 4.7: Sample paths for the two inventory processes.

The main strengths of the concept of regenerative processes are that the existence of limiting distributions is guaranteed under very mild conditions and the behavior of the limiting distribution depends only on the behavior of the process during a typical cycle.

Let $\{X_t\}$ be a regenerative process with regeneration times $T_0, T_1, T_2, \ldots$. Let $\tau_i = T_i - T_{i-1}$, $i = 1, 2, \ldots$ be the cycle lengths. Depending on whether $\{X_t\}$ is a discrete-time or continuous-time process, define, for some real-valued function H,

$$R_i = \sum_{t=T_{i-1}}^{T_i - 1} H(X_t) \tag{4.24}$$

or

$$R_i = \int_{T_{i-1}}^{T_i} H(X_t)\, \mathrm{d}t\ , \tag{4.25}$$

respectively, for $i = 1, 2, \ldots$. We assume, for simplicity, that $T_0 = 0$. We also assume that in the discrete case the cycle lengths are not always a multiple of some integer greater than 1. We can view R_i as the reward (or, alternatively, the cost) accrued during the i-th cycle. Let $\tau = T_1$ be the length of the first regeneration cycle, and let $R = R_1$ be the first reward.

The following properties of regenerative processes will be needed later on (see, e.g., [3]):

(a) If $\{X_t\}$ is regenerative, then the process $\{H(X_t)\}$ is regenerative as well.

(b) If $\mathbb{E}[\tau] < \infty$, then, under mild conditions, the process $\{X_t\}$ has a limiting (or steady-state) distribution, in the sense that there exists a random variable X, such that

$$\lim_{t\to\infty} \mathbb{P}(X_t \leqslant x) = \mathbb{P}(X \leqslant x) \; .$$

In the discrete case, no extra condition is required. In the continuous case, a sufficient condition is that the sample paths of the process are right-continuous and that the cycle length distribution is *non-lattice* — that is, the distribution does not concentrate all its probability mass at points $n\delta$, $n \in \mathbb{N}$, for some $\delta > 0$.

(c) If the conditions in (b) hold, the steady-state expected value, $\ell = \mathbb{E}[H(X)]$, is given by

$$\ell = \mathbb{E}[H(X)] = \frac{\mathbb{E}[R]}{\mathbb{E}[\tau]}. \tag{4.26}$$

(d) (R_i, τ_i), $i = 1, 2, \ldots$, is a sequence of iid random vectors.

Note that property (a) states that the behavior patterns of the system (or any measurable function thereof) during distinct cycles are statistically iid, while property (d) states that rewards and cycle lengths are jointly iid for distinct cycles. Formula (4.26) is fundamental to regenerative simulation. For typical non-Markovian queueing models, the quantity ℓ (the steady-state expected performance) is unknown and must be evaluated via regenerative simulation.

To obtain a point estimate of ℓ, one generates N regenerative cycles, calculates the iid sequence of two-dimensional random vectors (R_i, τ_i), $i = 1, \ldots, N$, and finally estimates ℓ by the *ratio* estimator

$$\widehat{\ell} = \frac{\widehat{R}}{\widehat{\tau}} \; , \tag{4.27}$$

where $\widehat{R} = N^{-1} \sum_{i=1}^{N} R_i$ and $\widehat{\tau} = N^{-1} \sum_{i=1}^{N} \tau_i$. Note that the estimator $\widehat{\ell}$ is biased; that is, $\mathbb{E}[\widehat{\ell}] \neq \ell$. However, $\widehat{\ell}$ is *strongly consistent*, that is, it converges to ℓ with probability 1 as $N \to \infty$. This follows directly from the fact that, by the law of large numbers, $\widehat{R}$ and $\widehat{\tau}$ converge with probability 1 to $\mathbb{E}[R]$ and $\mathbb{E}[\tau]$, respectively.

The *advantages* of the regenerative simulation method are:

(a) No deletion of transient data is necessary.

(b) It is asymptotically exact.

(b) It is easy to understand and implement.

The *disadvantages* of the regenerative simulation method are:

(a) For many practical cases, the output process, $\{X_t\}$, is either nonregenerative or its regeneration points are difficult to identify. Moreover, in complex systems (e.g., large queueing networks), checking for the occurrence of regeneration points could be computationally expensive.

(b) The estimator $\widehat{\ell}$ is biased.

(c) The regenerative cycles can be very long.

Next, we will establish a confidence interval for ℓ. Let $Z_i = R_i - \ell\tau_i$. It is readily seen that the Z_i are iid random variables, like the random vectors (R_i, τ_i). Letting $\widehat{R}$ and $\widehat{\tau}$ be defined as before, the central limit theorem ensures that

$$\frac{N^{1/2}\left(\widehat{R} - \ell\widehat{\tau}\right)}{\sigma} = \frac{N^{1/2}\left(\widehat{\ell} - \ell\right)}{\sigma/\widehat{\tau}}$$

converges in distribution to the standard normal distribution as $N \to \infty$, where

$$\sigma^2 = \mathrm{Var}(Z) = \mathrm{Var}(R) - 2\ell\,\mathrm{Cov}(R,\tau) + \ell^2\,\mathrm{Var}(\tau)\ . \tag{4.28}$$

Therefore, a $(1-\alpha)100\%$ confidence interval for $\ell = \mathbb{E}[R]/\mathbb{E}[\tau]$ is

$$\left(\widehat{\ell} \pm \frac{z_{1-\alpha/2}\,S}{\widehat{\tau}\,N^{1/2}}\right)\ , \tag{4.29}$$

where

$$S^2 = S_{11} - 2\,\widehat{\ell}\,S_{12} + \widehat{\ell}^{\,2} S_{22} \tag{4.30}$$

is the estimator of σ^2 based on replacing the unknown quantities in (4.28) with their unbiased estimators. That is,

$$S_{11} = \frac{1}{N-1}\sum_{i=1}^{N}(R_i - \widehat{R})^2, \quad S_{22} = \frac{1}{N-1}\sum_{i=1}^{N}(\tau_i - \widehat{\tau}\,)^2$$

and

$$S_{12} = \frac{1}{N-1}\sum_{i=1}^{N}(R_i - \widehat{R}\,)\,(\tau_i - \widehat{\tau}\,)\ .$$

Note that (4.29) differs from the standard confidence interval, say (4.3), by having an additional term $\widehat{\tau}$.

The algorithm for estimating the $(1-\alpha)100\%$ confidence interval for ℓ is as follows:

Algorithm 4.4.3: Regenerative Simulation Method

input : Simulation method for the process $\{X_t\}$, performance function H, number of regenerations N, confidence level $1-\alpha$.

output: Point estimate and $(1-\alpha)$-confidence interval for the expected steady-state performance $\ell = \mathbb{E}[H(X)]$.

1 Simulate N regenerative cycles of the process $\{X_t\}$.

2 Compute the sequence $\{(R_i, \tau_i), i = 1, \ldots, N\}$.

3 Calculate the point estimator $\widehat{\ell}$ and the confidence interval of ℓ from (4.27) and (4.29), respectively.

Note that if one uses two independent simulations of length N, one for estimating $\mathbb{E}[R]$ and the other for estimating $\mathbb{E}[\tau]$, then clearly $S^2 = S_{11} + \widehat{\ell}^{\,2} S_{22}$, since $\mathrm{Cov}(R,\tau) = 0$.

Remark 4.4.2 If the reward in each cycle is of the form (4.24) or (4.25), then $\ell = \mathbb{E}[H(X)]$ can be viewed as both the expected steady-state performance and the long-run average performance. This last interpretation is valid even if the reward in each cycle is not of the form (4.24)–(4.25) as long as the $\{(\tau_i, R_i)\}$ are iid. In that case,

$$\ell = \lim_{t\to\infty} \frac{\sum_{i=0}^{N_t-1} R_i}{t} = \frac{\mathbb{E}[R]}{\mathbb{E}[\tau]} , \tag{4.31}$$

where N_t is the number of regenerations in $[0, t]$.

■ EXAMPLE 4.9 Markov Chain: Example 4.6 (Continued)

Consider again the two-state Markov chain with the transition matrix

$$P = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}.$$

Assume, as in Example 4.6, that we start from 1 and obtain the following sample trajectory: $(x_0, x_1, x_2, \ldots, x_{10}) = (1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1)$, which has four cycles with lengths $\tau_1 = 4$, $\tau_2 = 2$, $\tau_3 = 1$, $\tau_4 = 3$ and corresponding transitions $(p_{12}, p_{22}, p_{22}, p_{21}), (p_{12}, p_{21}), (p_{11}), (p_{12}, p_{22}, p_{21})$. In addition, assume that each transition from i to j incurs a cost (or, alternatively, a reward) c_{ij} and that the related cost matrix is

$$C = (c_{ij}) = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} .$$

Note that the cost in each cycle is not of the form (4.24) (however, see Problem 4.14) but is given as

$$R_i = \sum_{t=T_{i-1}}^{T_i-1} c_{X_t, X_{t+1}}, \quad i = 1, 2, \ldots .$$

We illustrate the estimation procedure for the long-run average cost ℓ. First, observe that $R_1 = 1 + 3 + 3 + 2 = 9$, $R_2 = 3$, $R_3 = 0$, and $R_4 = 6$. It follows that $\widehat{R} = 4.5$. Since $\widehat{\tau} = 2.5$, the point estimate of ℓ is $\widehat{\ell} = 1.80$. Moreover, $S_{11} = 15, S_{22} = 5/3, S_{12} = 5$, and $S^2 = 2.4$. This gives a 95% confidence interval for ℓ of $(1.20, 2.40)$.

■ EXAMPLE 4.10 Example 4.7 (Continued)

Consider the sample path in Figure 4.6 of the process $\{X_t, t \geqslant 0\}$ describing the number of customers in the $GI/G/1$ system. The corresponding sample path data are given in Table 4.2.

Table 4.2: Sample path data for the $GI/G/1$ queueing process.

$t \in$ interval	X_t	$t \in$ interval	X_t	$t \in$ interval	X_t
$[0.00, 0.80)$	1	$[3.91, 4.84)$	1	$[6.72, 7.92)$	1
$[0.80, 1.93)$	2	$[4.84, 6.72)$	0	$[7.92, 9.07)$	2
$[1.93, 2.56)$	1			$[9.07, 10.15)$	1
$[2.56, 3.91)$	0			$[10.15, 11.61)$	0
Cycle 1		**Cycle 2**		**Cycle 3**	

Notice that the figure and table reveal three complete cycles with the following pairs: $(R_1, \tau_1) = (3.69, 3.91)$, $(R_2, \tau_2) = (0.93, 2.81)$, and $(R_3, \tau_3) = (4.58, 4.89)$. The resultant statistics are (rounded) $\widehat{\ell} = 0.79$, $S_{11} = 3.62$, $S_{22} = 1.08$, $S_{12} = 1.92$, $S^2 = 1.26$, and the 95% confidence interval is (0.79 ± 0.32).

■ **EXAMPLE 4.11 Example 4.8 (Continued)**

Let $\{X_t, t \geqslant 0\}$ be the inventory position process described in Example 4.8. Table 4.3 presents the data corresponding to the sample path in Figure 4.7 for a case where $s = 10$, $S = 40$, and $r = 1$.

Table 4.3: Data for the inventory position process, $\{X_t\}$, with $s = 10$ and $S = 40$. The boxes indicate the regeneration times.

t	X_t	t	X_t	t	X_t
$\boxed{0.00}$	40.00	$\boxed{5.99}$	40.00	$\boxed{9.67}$	40.00
1.79	32.34	6.41	33.91	11.29	32.20
3.60	22.67	6.45	23.93	11.38	24.97
5.56	20.88	6.74	19.53	12.05	18.84
5.62	11.90	8.25	13.32	13.88	13.00
		9.31	10.51	$\boxed{14.71}$	40.00

Based on the data in Table 4.3, we illustrate the derivation of the point estimator and the 95% confidence interval for the steady-state quantity $\ell = \mathbb{P}(X < 30) = \mathbb{E}[I_{\{X<30\}}]$, that is, the probability that the inventory position is less than 30. Table 4.3 shows three complete cycles with the following pairs: $(R_1, \tau_1) = (2.39, 5.99)$, $(R_2, \tau_2) = (3.22, 3.68)$, and $(R_3, \tau_3) = (3.33, 5.04)$, where $R_i = \int_{T_{i-1}}^{T_i} I_{\{X_t<30\}}\, \mathrm{d}t$. The resulting statistics are (rounded) $\widehat{\ell} = 0.61$, $S_{11} = 0.26$, $S_{22} = 1.35$, $S_{12} = -0.44$, and $S^2 = 1.30$, which gives a 95% confidence interval (0.61 ± 0.26).

4.5 BOOTSTRAP METHOD

Suppose that we estimate a number ℓ via some estimator $H = H(\mathbf{X})$, where $\mathbf{X} = (X_1, \ldots, X_n)$, and the $\{X_i\}$ form a random sample from some unknown distribution F. It is assumed that H does not depend on the order of the $\{X_i\}$. To assess the quality (e.g., accuracy) of the estimator H, we could draw independent replications $\mathbf{X}_1, \ldots, \mathbf{X}_N$ of $\mathbf{X}$ and find sample estimates for quantities such as the variance of the estimator

$$\mathrm{Var}(H) = \mathbb{E}[H^2] - (\mathbb{E}[H])^2,$$

the *bias* of the estimator

$$\text{Bias } = \mathbb{E}[H] - \ell\,,$$

and the expected quadratic error, or *mean square error* (MSE)

$$\text{MSE } = \mathbb{E}\left[(H-\ell)^2\right] \ .$$

However, it may be too time-consuming, or simply not feasible, to obtain such replications. An alternative is to *resample* the original data. Specifically, given an outcome $(x_1, \ldots, x_n)$ of $\mathbf{X}$, we draw a random sample $X_1^*, \ldots, X_n^*$ not from F but from an approximation to this distribution. The best estimate that we have about F on the grounds of $\{x_i\}$ is the *empirical distribution*, F_n, which assigns probability mass $1/n$ to each point $x_i, i = 1, \ldots, n$. In the one-dimensional case, the cdf of the empirical distribution is thus given by

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{\{x_i \leqslant x\}} \ .$$

Drawing from this distribution is trivial: for each j, draw $U \sim \mathsf{U}[0,1]$, let $J = \lfloor U\, n \rfloor + 1$, and return $X_j^* = x_J$. Note that if the $\{x_i\}$ are all different, vector $\mathbf{X}^* = (X_1^*, \ldots, X_n^*)$ can take n^n different values.

The rationale behind the resampling idea is that the empirical distribution F_n is close to the actual distribution F and gets closer as n gets larger. Hence, any quantities depending on F, such as $\mathbb{E}_F[h(H)]$, where h is a function, can be approximated by $\mathbb{E}_{F_n}[h(H)]$. The latter is usually still difficult to evaluate, but it can be simply estimated via Monte Carlo simulation as

$$\frac{1}{B}\sum_{i=1}^{B} h(H_i^*)\ ,$$

where $H_1^*, \ldots, H_B^*$ are independent copies of $H^* = H(\mathbf{X}^*)$. This seemingly self-referent procedure is called *bootstrapping* — alluding to Baron von Münchhausen, who pulled himself out of a swamp by his own bootstraps. As an example, the bootstrap estimate of the expectation of H is

$$\widehat{\mathbb{E}[H]} = \overline{H}^* = \frac{1}{B}\sum_{i=1}^{B} H_i^*\ ,$$

which is simply the sample mean of $\{H_i^*\}$. Similarly, the bootstrap estimate for $\mathrm{Var}(H)$ is the sample variance

$$\widehat{\mathrm{Var}(H)} = \frac{1}{B-1}\sum_{i=1}^{B}(H_i^* - \overline{H}^*)^2\ . \tag{4.32}$$

Perhaps of more interest are the bootstrap estimators for the bias and MSE, respectively $\overline{H}^* - H$ and

$$\frac{1}{B}\sum_{i=1}^{B}(H_i^* - H)^2 \,.$$

Note that for these estimators the unknown quantity ℓ is replaced with the original estimator H. Confidence intervals can be constructed in the same fashion. We discuss two variants: the *normal* method and the *percentile* method. In the normal method, a $(1-\alpha)100\%$ confidence interval for ℓ is given by

$$(H \pm z_{1-\alpha/2} S^*) \,,$$

where S^* is the bootstrap estimate of the standard deviation of H, that is, the square root of (4.32). In the percentile method, the upper and lower bounds of the $(1-\alpha)100\%$ confidence interval for ℓ are given by the $1-\alpha/2$ and $\alpha/2$ quantiles of H, which in turn are estimated via the corresponding sample quantiles of the bootstrap sample $\{H_i^*\}$.

PROBLEMS

4.1 We wish to estimate $\ell = \int_{-2}^{2} \mathrm{e}^{-x^2/2}\,\mathrm{d}x = \int H(x) f(x)\,\mathrm{d}x$ via Monte Carlo simulation using two different approaches: (A) defining $H(x) = 4\,\mathrm{e}^{-x^2/2}$ and f the pdf of the $\mathsf{U}[-2,2]$ distribution and (B) defining $H(x) = \sqrt{2\pi}\, I_{\{-2 \leqslant x \leqslant 2\}}$ and f the pdf of the $\mathsf{N}(0,1)$ distribution.

- **a)** For both cases, estimate ℓ via the estimator $\widehat{\ell}$ in (4.10). Use a sample size of $N = 100$.
- **b)** For both cases, estimate the relative error of $\widehat{\ell}$, using $N = 100$.
- **c)** Give a 95% confidence interval for ℓ for both cases, using $N = 100$.
- **d)** From b), assess how large N should be such that the relative width of the confidence interval is less than 0.001, and carry out the simulation with this N. Compare the result with the true value of ℓ.

4.2 Prove that the structure function of the bridge system in Figure 4.1 is given by (4.11).

4.3 Consider the bridge system in Figure 4.1. Suppose that all link reliabilities are p. Show that the reliability of the system is $p^2(2 + 2\,p - 5\,p^2 + 2\,p^3)$.

4.4 Estimate the reliability of the bridge system in Figure 4.1 via (4.10) if the link reliabilities are $(p_1, \ldots, p_5) = (0.7,\, 0.6,\, 0.5,\, 0.4,\, 0.3)$. Choose a sample size such that the estimate has a relative error of about 0.01.

4.5 Consider the following sample performance:

$$H(\mathbf{X}) = \min\{X_1 + X_2,\; X_1 + X_4 + X_5,\; X_3 + X_4\}.$$

Assume that the random variables $X_i,\ i = 1, \ldots, 5$ are iid with common distribution

(a) $\mathsf{Gamma}(\lambda_i, \beta_i)$, where $\lambda_i = i$ and $\beta_i = i$.

(b) $\mathsf{Ber}(p_i)$, where $p_i = 1/2i$.

Run a computer simulation with $N = 1000$ replications, and find point estimates and 95% confidence intervals for $\ell = \mathbb{E}[H(\mathbf{X})]$.

4.6 Consider the precedence ordering of activities in Table 4.4. Suppose that durations of the activities (when actually started) are independent of each other, and all have exponential distributions with parameters 1.1, 2.3, 1.5, 2.9, 0.7, and 1.5, for activities 1, ..., 6, respectively.

Table 4.4: Precedence ordering of activities.

Activity	1	2	3	4	5	6
Predecessor(s)	-	-	1	2,3	2,3	5

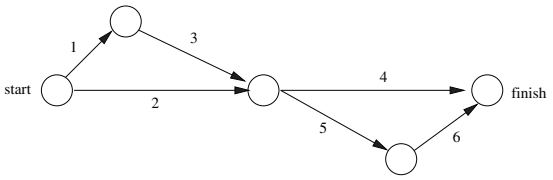

Figure 4.8: The PERT network corresponding to Table 4.4.

a) Verify that the corresponding PERT graph is given by Figure 4.8.
b) Identify the four possible paths from start to finish.
c) Estimate the expected length of the critical path in (4.13) with a relative error of less than 5%.

4.7 Let $\{X_t, t = 0, 1, 2, \ldots\}$ be a random walk on the positive integers; see Example 1.11. Suppose that $p = 0.55$ and $q = 0.45$. Let $X_0 = 0$. Let Y be the maximum position reached after 100 transitions. Estimate the probability that $Y \geqslant 15$ and give a 95% confidence interval for this probability based on 1000 replications of Y.

4.8 Consider the $M/M/1$ queue. Let X_t be the number of customers in the system at time $t \geqslant 0$. Run a computer simulation of the process $\{X_t, t \geqslant 0\}$ with $\lambda = 1$ and $\mu = 2$, starting with an empty system. Let X denote the steady-state number of people in the system. Find point estimates and confidence intervals for $\ell = \mathbb{E}[X]$, using the batch means and regenerative methods as follows:

a) For the batch means method run the system for a simulation time of 10,000, discard the observations in the interval $[0, 100]$, and use $N = 30$ batches.
b) For the regenerative method, run the system for the same amount of simulation time (10,000) and take as regeneration points the times where an arriving customer finds the system empty.
c) For both methods, find the requisite simulation time that ensures a relative width of the confidence interval not exceeding 5%.

4.9 Let Z_n be the number of customers in an $M/M/1$ queueing system, as seen by the n-th arriving customer, $n = 1, 2, \ldots$. Suppose that the service rate is $\mu = 1$ and the arrival rate is $\lambda = 0.6$. Let Z be the steady-state queue length (as seen by an arriving customer far away in the future). Note that $Z_n = X_{T_n-}$, with X_t

as in Problem 4.8, and T_n is the arrival epoch of the n-th customer. Here, "T_n-" denotes the time just before T_n.

a) Verify that $\ell = \mathbb{E}[Z] = 1.5$.

b) Explain how to generate $\{Z_n, n = 1, 2, \ldots\}$ using a random walk on the positive integers, as in Problem 4.7.

c) Find the point estimate of ℓ and a 95% confidence interval for ℓ using the batch means method. Use a sample size of 10^4 customers and $N = 30$ batches, throwing away the first $K = 100$ observations.

d) Do the same as in c) using the regenerative method instead.

e) Assess the minimum length of the simulation run in order to obtain a 95% confidence interval with an absolute width w_a not exceeding 5%.

f) Repeat c), d), and e) with $\varrho = 0.8$ and discuss c), d), and e) as $\varrho \to 1$.

4.10 Table 4.5 displays a realization of a Markov chain, $\{X_t, t = 0, 1, 2, \ldots\}$, with state space $\{0, 1, 2, 3\}$ starting at 0. Let X be distributed according to the limiting distribution of this chain (assuming it has one).

Table 4.5: A realization of the Markov chain.

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X_t	0	3	0	1	2	1	0	2	0	1	0	1	0	2	0

Find the point estimator, $\widehat{\ell}$, and the 95% confidence interval for $\ell = \mathbb{E}[X]$ using the regenerative method.

4.11 Let W_n be the *waiting time* of the n-th customer in a $GI/G/1$ queue, that is, the total time the customer spends waiting in the queue (thus excluding the service time). The waiting time process $\{W_n, n = 1, 2, \ldots\}$ follows the following well-known *Lindley equation*:

$$W_{n+1} = \max\{W_n + S_n - A_{n+1},\, 0\}, \quad n = 1, 2, \ldots, \tag{4.33}$$

where A_{n+1} is the interval between the n-th and $(n+1)$-st arrivals, S_n is the service time of the n-th customer, and $W_1 = 0$ (the first customer does not have to wait and is served immediately).

a) Explain why the Lindley equation holds.

b) Find the point estimate and the 95% confidence interval for the expected waiting time for the 4-th customer in an $M/M/1$ queue with $\varrho = 0.5$, ($\lambda = 1$), starting with an empty system. Use $N = 5000$ replications.

c) Find point estimates and confidence intervals for the expected average waiting time for customers $21, \ldots, 70$ in the same system as in b). Use $N = 5000$ replications. Note that the point estimate and confidence interval required are for the following parameter:

$$\ell = \mathbb{E}\left[\frac{1}{50}\sum_{n=21}^{70} W_n\right].$$

4.12 Run a computer simulation of 1000 regenerative cycles of the (s, S) policy inventory model (see Example 4.8), where demands arrive according to a Poisson

process with rate 2 (i.e., $A \sim \mathsf{Exp}(2)$) and the size of each demand follows a Poisson distribution with mean 2 (i.e., $D \sim \mathsf{Poi}(2)$). Take $s = 1$, $S = 6$, lead time $r = 2$, and initial value $X_0 = 4$. Find point estimates and confidence intervals for the quantity $\ell = \mathbb{P}(2 \leqslant X \leqslant 4)$, where X is the steady-state inventory position.

4.13 Simulate the Markov chain $\{X_n\}$ in Example 4.9, using $p_{11} = 1/3$ and $p_{22} = 3/4$ for 1000 regeneration cycles. Obtain a confidence interval for the long-run average cost.

4.14 Consider Example 4.9 again, with $p_{11} = 1/3$ and $p_{22} = 3/4$. Define $Y_i = (X_i, X_{i+1})$ and $H(Y_i) = c_{X_i, X_{i+1}}$, $i = 0, 1, \ldots$. Show that $\{Y_i\}$ is a regenerative process. Find the corresponding limiting/steady-state distribution and calculate $\ell = \mathbb{E}[H(Y)]$, where Y is distributed according to this limiting distribution. Check if ℓ is contained in the confidence interval obtained in Problem 4.13.

4.15 Consider the tandem queue of Section 3.4.1. Let X_t and Y_t denote the number of customers in the first and second queues at time t, including those who are possibly being served. Is $\{(X_t, Y_t), t \geqslant 0\}$ a regenerative process? If so, specify the regeneration times.

4.16 Consider the machine repair problem in Problem 3.5, with three machines and two repair facilities. Each repair facility can take only one failed machine. Suppose that the lifetimes are $\mathsf{Exp}(1/10)$ distributed and the repair times are $\mathsf{U}(0, 8)$ distributed. Let ℓ be the limiting probability that all machines are out of order.

- **a)** Estimate ℓ via the regenerative estimator $\widehat{\ell}$ in (4.27) using 1000 regeneration cycles. Compute the 95% confidence interval (4.30).
- **b)** Estimate the bias and MSE of $\widehat{\ell}$ using the bootstrap method with a sample size of $B = 300$. [Hint: The original data are $\mathbf{X} = (X_1, \ldots, X_{100})$, where $X_i = (R_i, \tau_i)$, $i = 1, \ldots, 100$. Resample from these data using the empirical distribution.]
- **c)** Compute 95% bootstrap confidence intervals for ℓ using the normal and percentile methods with $B = 1000$ bootstrap samples.

Further Reading

The regenerative method in a simulation context was introduced and developed by Crane and Iglehart [4, 5]. A more complete treatment of regenerative processes is given in [3]. Fishman [7] treats the statistical analysis of simulation data in great detail. Gross and Harris [8] is a classical reference on queueing systems. Efron and Tibshirani [6] gives the defining introduction to the bootstrap method. A modern introduction to statistical modeling and computation can be found in [9].

REFERENCES

1. J. Abate and W. Whitt. Transient behavior of regulated Brownian motion, I: Starting at the origin. *Advances in Applied Probability*, 19:560–598, 1987.
2. J. Abate and W. Whitt. Transient behavior of regulated Brownian motion, II: Nonzero initial conditions. *Advances in Applied Probability*, 19:599–631, 1987.
3. S. Asmussen. *Applied Probability and Queues.* John Wiley & Sons, New York, 1987.

4. M. A. Crane and D. L. Iglehart. Simulating stable stochastic systems, I: General multiserver queues. *Journal of the ACM*, 21:103–113, 1974.

5. M. A. Crane and D. L. Iglehart. Simulating stable stochastic systems, II: Markov chains. *Journal of the ACM*, 21:114–123, 1974.

6. B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1994.

7. G. S. Fishman. *Monte Carlo: Concepts, Algorithms and Applications*. Springer-Verlag, New York, 1996.

8. D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. John Wiley & Sons, New York, 2nd edition, 1985.

9. D. P. Kroese and J. C. C. Chan. *Statistical Modeling and Computation*. Springer, New York, 2014.

10. A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 3rd edition, 2000.

CHAPTER 5

CONTROLLING THE VARIANCE

5.1 INTRODUCTION

This chapter treats basic theoretical and practical aspects of *variance reduction* techniques. Variance reduction can be viewed as a means of utilizing known information about the model in order to obtain more accurate estimators of its performance. Generally, the more we know about the system, the more effective is the variance reduction. One way of gaining this information is through a pilot simulation run of the model. Results from this first-stage simulation can then be used to formulate variance reduction techniques that will subsequently improve the accuracy of the estimators in the second simulation stage. Two of the most effective techniques for variance reduction are *importance sampling* and *conditional Monte Carlo*. Other well-known techniques that can provide moderate variance reduction include the use of common and antithetic variables, control variables, and stratification. The splitting method, discussed in Chapter 9, is another powerful approach to variance reduction.

The chapter is organized as follows. We start, in Sections 5.2–5.5, with common and antithetic variables, control variables, conditional Monte Carlo, and stratified sampling. Section 5.6 introduces the multilevel Monte Carlo method for the estimation of performance measures of diffusion processes. Most of our attention, from Section 5.7 on, is focused on *importance sampling* and *likelihood ratio* techniques. Using importance sampling, one can often achieve substantial (sometimes dramatic) variance reduction, in particular when estimating rare-event probabilities. In Sec-

Simulation and the Monte Carlo Method, Third Edition. By R. Y. Rubinstein and D. P. Kroese

tion 5.7 we present two alternative importance sampling-based techniques, called the *variance minimization* and *cross-entropy* methods. Sections 5.8–5.10 discuss how importance sampling can be carried out sequentially/dynamically. Section 5.11 presents a simple, convenient, and unifying way of constructing efficient importance sampling estimators: the so-called *transform likelihood ratio* (TLR) method. Finally, in Section 5.12 we present the *screening* method for variance reduction, which can also be seen as a dimension-reduction technique. The aim of this method is to identify (screen out) the most important (bottleneck) parameters of the simulated system to be used in an importance sampling estimation procedure.

5.2 COMMON AND ANTITHETIC RANDOM VARIABLES

To motivate the use of common and antithetic random variables in simulation, let us consider a simple example. Let X and Y be random variables with known cdfs, F and G, respectively. Suppose that we need to estimate $\ell = \mathbb{E}[X-Y]$ via simulation. The simplest unbiased estimator for ℓ is $X-Y$. We can simulate X and Y via the IT method:

$$\begin{aligned} X &= F^{-1}(U_1)\,, \quad U_1 \sim \mathsf{U}(0,1)\,, \\ Y &= G^{-1}(U_2)\,, \quad U_2 \sim \mathsf{U}(0,1)\,. \end{aligned} \tag{5.1}$$

It is important to note that X and Y (or U_1 and U_2) *need not be independent*. In fact, since

$$\mathrm{Var}(X-Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) - 2\,\mathrm{Cov}(X,Y) \tag{5.2}$$

and since the marginal cdfs of X and Y have been prescribed, it follows that the variance of $X-Y$ can be minimized by maximizing the covariance in (5.2). We say that *common random variables* are used in (5.1) if $U_2 = U_1$ and *antithetic random variables* are used if $U_2 = 1-U_1$. Since both F^{-1} and G^{-1} are nondecreasing functions, in using common random variables, we clearly have

$$\mathrm{Cov}\left(F^{-1}(U),\ G^{-1}(U)\right) \geqslant 0$$

for $U \sim \mathsf{U}(0,1)$. Consequently, variance reduction is achieved, in the sense that the estimator $F^{-1}(U) - G^{-1}(U)$ has a smaller variance than the *crude Monte Carlo* (CMC) estimator $X-Y$, where X and Y are independent, with cdfs F and G, respectively. In fact, it is well known (e.g., see [44]) that using common random variables maximizes the covariance between X and Y, so that $\mathrm{Var}(X-Y)$ is *minimized*. Similarly, $\mathrm{Var}(X+Y)$ is minimized when antithetic random variables are used.

Now consider minimal variance estimation of $\mathbb{E}[H_1(X) - H_2(Y)]$, where X and Y are unidimensional variables with known marginal cdfs, F and G, respectively, and H_1 and H_2 are real-valued monotone functions. Mathematically, the problem can be formulated as follows:

> Within the set of all two-dimensional joint cdfs of (X,Y), find a joint cdf, F^*, that minimizes $\mathrm{Var}(H_1(X) - H_2(Y))$, subject to X and Y having the prescribed cdfs F and G, respectively.

This problem has been solved by Gal, Rubinstein, and Ziv [14], who proved that if H_1 and H_2 are monotonic in the *same* direction, then the use of common random variables leads to optimal variance reduction, that is,

$$\min_{F^*} \mathrm{Var}(H_1(X) - H_2(Y)) = \mathrm{Var}\left(H_1[F^{-1}(U)] - H_2[G^{-1}(U)]\right)\ . \tag{5.3}$$

The proof of (5.3) uses the fact that if $H(u)$ is a monotonic function, then $H(F^{-1}(U))$ is monotonic as well, since $F^{-1}(u)$ is. By symmetry, if H_1 and H_2 are monotonic in *opposite* directions, then the use of antithetic random variables (i.e., $U_2 = 1 - U_1$) yields optimal variance reduction.

This result can be further generalized by considering minimal variance estimation of

$$\mathbb{E}[H_1(\mathbf{X}) - H_2(\mathbf{Y})] \,, \tag{5.4}$$

where $\mathbf{X} = (X_1, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, \ldots, Y_n)$ are random vectors with $X_i \sim F_i$ and $Y_i \sim G_i$, $i = 1, \ldots, n$, and the functions H_1 and H_2 are real-valued and monotone in each component of $\mathbf{X}$ and $\mathbf{Y}$. If the pairs $\{(X_i, Y_i)\}$ are independent and H_1 and H_2 are monotonic in the same direction (for each component), then the use of common random variables again leads to minimal variance. That is, we take $X_i = F_i^{-1}(U_i)$ and $Y_i = G_i^{-1}(U_i)$, $i = 1, \ldots, n$, where $U_1, \ldots, U_n$ are independent $\mathsf{U}(0,1)$-distributed random variables, or, symbolically,

$$\mathbf{X} = F^{-1}(\mathbf{U}), \quad \mathbf{Y} = G^{-1}(\mathbf{U}) \,. \tag{5.5}$$

Similarly, if H_1 and H_2 are monotonic in opposite directions, then using antithetic random variables is optimal. Last, if H_1 and H_2 are monotonically increasing with respect to some components and monotonically decreasing with respect to others, then minimal variance is obtained by using the appropriate combination of common and antithetic random variables.

We now describe one of the main applications of antithetic random variables. We want to estimate

$$\ell = \mathbb{E}[H(\mathbf{X})] \,,$$

where $\mathbf{X} \sim F$ is a random vector with independent components and the sample performance function, $H(\mathbf{x})$, is monotonic in each component of $\mathbf{x}$. An example of such a function is given below.

EXAMPLE 5.1 Stochastic Shortest Path

Consider the undirected graph in Figure 5.1, depicting a so-called *bridge network*.

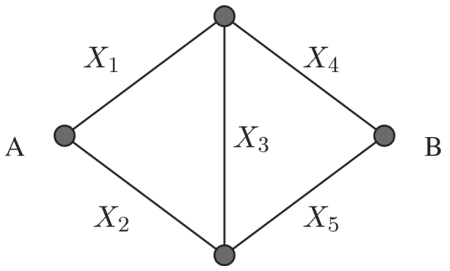

Figure 5.1: Determine the shortest path from A to B in a bridge network.

Our objective is to estimate the expected length ℓ of the shortest path between nodes (vertices) A and B, where the lengths of the links (edges) are

random variables $X_1, \ldots, X_5$. We have $\ell = \mathbb{E}[H(\mathbf{X})]$, where

$$H(\mathbf{X}) = \min\{X_1 + X_4,\ X_1 + X_3 + X_5,\ X_2 + X_3 + X_4,\ X_2 + X_5\}\,. \tag{5.6}$$

Note that $H(\mathbf{x})$ is nondecreasing in each component of the vector $\mathbf{x}$.

Similarly, the length of the shortest path $H(\mathbf{X})$ in an arbitrary network with random edge lengths $\{X_i\}$ can be written as

$$H(\mathbf{X}) = \min_{j=1,\ldots,p} \sum_{i \in \mathscr{P}_j} X_i\,, \tag{5.7}$$

where $\mathscr{P}_j$ is the j-th complete path from the source to the sink of the network and p is the number of complete paths in the network. The sample performance is nondecreasing in each of the components.

An unbiased estimator of $\ell = \mathbb{E}[H(\mathbf{X})]$ is the CMC estimator, given by

$$\widehat{\ell} = \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k)\,, \tag{5.8}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is an iid sample from the (multidimensional) cdf F. An alternative unbiased estimator of ℓ, for N even, is

$$\widehat{\ell}^{(a)} = \frac{1}{N} \sum_{k=1}^{N/2} \left\{ H(\mathbf{X}_k) + H(\mathbf{X}_k^{(a)}) \right\}, \tag{5.9}$$

where $\mathbf{X}_k = F^{-1}(\mathbf{U}_k)$ and $\mathbf{X}_k^{(a)} = F^{-1}(\mathbf{1} - \mathbf{U}_k)$, using notation similar to (5.5). The estimator $\widehat{\ell}^{(a)}$ is called the *antithetic estimator* of ℓ. Since $H(\mathbf{X}) + H(\mathbf{X}^{(a)})$ is a particular case of $H_1(\mathbf{X}) - H_2(\mathbf{Y})$ in (5.4) (with $H_2(\mathbf{Y})$ replaced by $-H(\mathbf{X}^{(a)})$), we immediately obtain that $\mathrm{Var}(\widehat{\ell}^{(a)}) \leqslant \mathrm{Var}(\widehat{\ell})$. That is, the antithetic estimator, $\widehat{\ell}^{(a)}$, is more accurate than the CMC estimator, $\widehat{\ell}$.

To compare the efficiencies of $\widehat{\ell}$ and $\widehat{\ell}^{(a)}$, we can consider their *relative time variance*,

$$\varepsilon = \frac{T^{(a)}\,\mathrm{Var}(\widehat{\ell}^{(a)})}{T\,\mathrm{Var}(\widehat{\ell})}\,, \tag{5.10}$$

where $T^{(a)}$ and T are the CPU times required to calculate the estimators $\widehat{\ell}^{(a)}$ and $\widehat{\ell}$, respectively. Note that

$$\begin{aligned}\mathrm{Var}(\widehat{\ell}^{(a)}) &= \frac{N/2}{N^2}\left(\mathrm{Var}(H(\mathbf{X})) + \mathrm{Var}(H(\mathbf{X}^{(a)})) + 2\,\mathrm{Cov}[H(\mathbf{X}), H(\mathbf{X}^{(a)})]\right)\\ &= \mathrm{Var}(\widehat{\ell}) + \mathrm{Cov}(H(\mathbf{X}), H(\mathbf{X}^{(a)}))/N\,.\end{aligned}$$

Also, $T^{(a)} \leqslant T$, since the antithetic estimator, $\widehat{\ell}^{(a)}$, needs only *half* as many random numbers as its CMC counterpart, $\widehat{\ell}$. Neglecting this time advantage, the efficiency measure (5.10) reduces to

$$\varepsilon = \frac{\mathrm{Var}(\widehat{\ell}^{(a)})}{\mathrm{Var}(\widehat{\ell})} = 1 + \frac{\mathrm{Cov}[H(\mathbf{X}), H(\mathbf{X}^{(a)})]}{\mathrm{Var}(H(\mathbf{X}))}\,, \tag{5.11}$$

where the covariance is negative and can be estimated via the corresponding sample covariance.

The use of common/antithetic random variables for the case of dependent components of $\mathbf{X}$ and $\mathbf{Y}$ for strictly monotonic functions, H_1 and H_2, is presented in Rubinstein, Samorodnitsky, and Shaked [39].

EXAMPLE 5.2 Stochastic Shortest Path (Continued)

We estimate the expected length of the shortest path for the bridge network in Example 5.1 for the case where each link has an exponential weight with parameter 1. Taking a sample size of $N = 10{,}000$ obtains the CMC estimate $\widehat{\ell} = 1.159$ with an estimated variance of $5.6 \cdot 10^{-5}$, whereas the antithetic estimate is $\widehat{\ell} = 1.164$ with an estimated variance of $2.8 \cdot 10^{-5}$. Therefore, the efficiency ε of the estimator $\widehat{\ell}^{(a)}$ relative to the CMC estimator $\widehat{\ell}$ is about 2.0.

EXAMPLE 5.3 Lindley's Equation

Consider Lindley's equation for the waiting time of the $(n+1)$-st customer in a $GI/G/1$ queue :

$$W_{n+1} = \max\{W_n + U_n, 0\}, \quad W_1 = 0 \, .$$

See also (4.33). Here $U_n = S_n - A_{n+1}$, where S_n is the service time of the n-th customer, and A_{n+1} is the interarrival time between the n-th and $(n+1)$-st customer. Since W_n is a monotonic function of each component $A_2, \ldots, A_n$ and $S_1, \ldots, S_{n-1}$, we can obtain variance reduction by using antithetic random variables.

5.3 CONTROL VARIABLES

The *control variables* method is a widely used variance reduction technique. We first consider the one-dimensional case. Let X be an unbiased estimator of μ, to be obtained from a simulation run. A random variable C is called a *control variable* for X if it is correlated with X and its expectation, r, is known. The control variable C is used to construct an unbiased estimator of μ with a variance smaller than that of X. This estimator,

$$X_\alpha = X - \alpha(C - r) \, , \tag{5.12}$$

where α is a scalar parameter, is called the *linear control variable*. The variance of X_α is given by

$$\mathrm{Var}(X_\alpha) = \mathrm{Var}(X) - 2\alpha \, \mathrm{Cov}(X, C) + \alpha^2 \, \mathrm{Var}(C)$$

(see, e.g., Problem 1.15). Consequently, the value α^* that minimizes $\mathrm{Var}(X_\alpha)$ is

$$\alpha^* = \frac{\mathrm{Cov}(X, C)}{\mathrm{Var}(C)} \, . \tag{5.13}$$

Typically, α^* is estimated from the corresponding sample covariance and variance. Using α^*, we can write the minimal variance as

$$\mathrm{Var}(X_{\alpha^*}) = (1 - \varrho_{XC}^2)\mathrm{Var}(X) \, , \tag{5.14}$$

where ϱ_{XC} denotes the correlation coefficient of X and C. Notice that the larger $|\varrho_{XC}|$ is, the greater is the variance reduction.

Formulas (5.12)–(5.14) can be easily extended to the case of multiple control variables. To see this, let $\mathbf{C} = (C_1, \ldots, C_m)^\top$ be a (column) vector of m control variables with known mean vector $\mathbf{r} = \mathbb{E}[\mathbf{C}] = (r_1, \ldots, r_m)^\top$, where $r_i = \mathbb{E}[C_i]$. Then the vector version of (5.12) can be written as

$$X_\alpha = X - \boldsymbol{\alpha}^\top (\mathbf{C} - \mathbf{r}) \,, \tag{5.15}$$

where $\boldsymbol{\alpha}$ is an m-dimensional vector of parameters. The value $\boldsymbol{\alpha}^*$ that minimizes $\mathrm{Var}(X_{\boldsymbol{\alpha}})$ is given by

$$\boldsymbol{\alpha}^* = \Sigma_C^{-1} \boldsymbol{\sigma}_{XC} \,, \tag{5.16}$$

where Σ_C denotes the $m \times m$ covariance matrix of $\mathbf{C}$ and $\boldsymbol{\sigma}_{XC}$ denotes the $m \times 1$ vector whose i-th component is the covariance of X and C_i, $i = 1, \ldots, m$. The corresponding minimal variance evaluates to

$$\mathrm{Var}(X_{\alpha^*}) = (1 - R_{XC}^2)\mathrm{Var}(X) \,, \tag{5.17}$$

where

$$R_{XC}^2 = (\boldsymbol{\sigma}_{XC})^\top \Sigma_C^{-1} \boldsymbol{\sigma}_{XC} / \mathrm{Var}(X)$$

is the square of the so-called *multiple correlation coefficient* of X and $\mathbf{C}$. Again the larger $|R_{XC}|$ is, the greater is the variance reduction. The case where $\mathbf{X}$ is a vector with dependent components and the vector $\boldsymbol{\alpha}$ is replaced by a corresponding matrix is treated in Rubinstein and Marcus [36].

The following examples illustrate various applications of the control variables method.

EXAMPLE 5.4 Stochastic Shortest Path (Continued)

Consider again the stochastic shortest path estimation problem for the bridge network in Example 5.1. Among the control variables we can use are the lengths of the paths $\mathscr{P}_j$, $j = 1, \ldots, 4$, that is, any (or all) of

$$\begin{aligned} C_1 &= X_1 + X_4 \\ C_2 &= X_1 + X_3 + X_5 \\ C_3 &= X_2 + X_3 + X_4 \\ C_4 &= X_2 + X_5 \,. \end{aligned}$$

The expectations of the $\{C_i\}$ are easy to calculate, and each C_i is positively correlated with the length of the shortest path $H(\mathbf{X}) = \min\{C_1, \ldots, C_4\}$.

EXAMPLE 5.5 Lindley's Equation (Continued)

Consider Lindley's equation for the waiting time process $\{W_n, n = 1, 2, \ldots\}$ in the $GI/G/1$ queue; see Example 5.3. As a control variable for W_n we can take C_n, defined by the recurrence relation

$$C_{n+1} = C_n + U_n, \qquad C_1 = 0 \,,$$

where $U_n = S_n - A_{n+1}$, as in the waiting time process. Obviously, C_n and W_n are highly correlated. Moreover, the expectation $r_n = \mathbb{E}[C_n]$ is known. It is $r_n = (n-1)\,(\mathbb{E}[S] - \mathbb{E}[A])$, where $\mathbb{E}[S]$ and $\mathbb{E}[A]$ are the expected service and interarrival times, respectively. The corresponding linear control process is

$$Y_n = W_n - \alpha(C_n - r_n)\ .$$

■ **EXAMPLE 5.6 Queueing Networks**

Now we return to the estimation of the expected steady-state performance $\ell = \mathbb{E}[X]$ in a queueing network. For example, suppose that X is the steady-state number of customers in the system. As a linear control random process, one may take

$$Y_t = X_t - \alpha(C_t - r_t)\ ,$$

where X_t is the number of customers in the original system, and C_t is the number of customers in an auxiliary *Markovian* network for which the steady-state distribution is known. The latter network must be synchronized in time with the original network.

In order to produce high correlations between the two processes, $\{X_t\}$ and $\{C_t\}$, it is desirable that both networks have similar topologies and similar loads. In addition, they must use a common stream of random numbers for generating the input variables. Expressions for the expected steady-state performance $r = \mathbb{E}[C]$, such as the expected number in the system in a Markovian network, may be found in [19].

5.4 CONDITIONAL MONTE CARLO

Let

$$\ell = \mathbb{E}[H(\mathbf{X})] = \int H(\mathbf{x})\, f(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{5.18}$$

be some expected performance measure of a computer simulation model, where $\mathbf{X}$ is the input random variable (vector) with a pdf $f(\mathbf{x})$ and $H(\mathbf{X})$ is the sample performance measure (output random variable). Suppose that there is a random variable (or vector), $\mathbf{Y} \sim g(\mathbf{y})$, such that the conditional expectation $\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y} = \mathbf{y}]$ can be computed analytically. Since, by (1.11),

$$\ell = \mathbb{E}[H(\mathbf{X})] = \mathbb{E}[\,\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y}]\,]\ , \tag{5.19}$$

it follows that $\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y}]$ is an unbiased estimator of ℓ. Furthermore, it is readily seen that

$$\mathrm{Var}(\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y}]) \leqslant \mathrm{Var}(H(\mathbf{X}))\ , \tag{5.20}$$

so using the random variable $\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y}]$, instead of $H(\mathbf{X})$, leads to variance reduction. Thus conditioning *always* leads to variance reduction. To prove (5.20), we use the property (see Problem 5.7) that for any pair of random variables (U, V),

$$\mathrm{Var}(U) = \mathbb{E}[\,\mathrm{Var}(U \,|\, V)] + \mathrm{Var}(\,\mathbb{E}[U \,|\, V]\,)\ . \tag{5.21}$$

Since both terms on the right-hand side are nonnegative, (5.20) immediately follows. The conditional Monte Carlo idea is sometimes referred to as *Rao-Blackwellization*. The conditional Monte Carlo algorithm is given next.

Algorithm 5.4.1: Conditional Monte Carlo

input : Method to generate $\mathbf{Y} \sim g$, performance function H, sample size N.
output: Estimator $\widehat{\ell}_c$ of $\ell = \mathbb{E}[H(\mathbf{X})]$.

1 Generate an iid sample $\mathbf{Y}_1, \ldots \mathbf{Y}_N$ from g.
2 Calculate $\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y}_k],\ k = 1, \ldots, N$ analytically.
3 Set $\widehat{\ell}_c \leftarrow \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y}_k]$.
4 **return** $\widehat{\ell}_c$

Algorithm 5.4.1 requires that a random variable $\mathbf{Y}$ be found, such that $\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y} = \mathbf{y}]$ is known analytically for all $\mathbf{y}$. Moreover, for Algorithm 5.4.1 to be of practical use, the following conditions must be met:

(a) $\mathbf{Y}$ should be easy to generate.

(b) $\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y} = \mathbf{y}]$ should be readily computable for all values of $\mathbf{y}$.

(c) $\mathbb{E}[\mathrm{Var}(H(\mathbf{X}) \,|\, \mathbf{Y})]$ should be large relative to $\mathrm{Var}(\,\mathbb{E}[H(\mathbf{X}) \,|\, \mathbf{Y}]\,)$.

EXAMPLE 5.7 Random Sums

Consider the estimation of

$$\ell = \mathbb{P}(S_R \leqslant x) = \mathbb{E}[I_{\{S_R \leqslant x\}}] \,,$$

where

$$S_R = \sum_{i=1}^{R} X_i \,,$$

R is a random variable with a given distribution and the $\{X_i\}$ are iid with $X_i \sim F$ and independent of R. Let F^r be the cdf of the random variable S_r for fixed $R = r$. Noting that

$$F^r(x) = \mathbb{P}\left(\sum_{i=1}^{r} X_i \leqslant x\right) = \mathbb{E}\left[F\left(x - \sum_{i=2}^{r} X_i\right)\right],$$

we obtain

$$\ell = \mathbb{E}\left[\,\mathbb{E}\left[I_{\{S_R \leqslant x\}} \,\middle|\, \sum_{i=2}^{R} X_i\right]\right] = \mathbb{E}\left[F\left(x - \sum_{i=2}^{R} X_i\right)\right].$$

Thus, we can take the following estimator of ℓ based on conditioning:

$$\widehat{\ell}_c = \frac{1}{N} \sum_{k=1}^{N} F\left(x - \sum_{i=2}^{R_k} X_{ki}\right). \tag{5.22}$$

5.4.1 Variance Reduction for Reliability Models

Next, we present two variance reduction techniques for reliability models based on conditioning. As in Example 4.2 on page 110, we are given an unreliable system of n components, each of which can be either functioning or failed, with a structure function H that determines the state of the system (working or failed) as a function of the states of the components. The component states $X_1, \ldots, X_n$ are assumed to be independent, with reliabilities $\{p_i\}$ and unreliabilities $\{q_i\}$, where $q_i = 1 - p_i$. The probability of system failure — the unreliability of the system — is thus $\overline{r} = \mathbb{P}(H(\mathbf{X}) = 0)$. In typical applications the unreliability is very small and is difficult to estimate via CMC.

5.4.1.1 Permutation Monte Carlo Permutation Monte Carlo is a conditional Monte Carlo technique for network reliability estimation (see Elperin et al. [12]). Here the components are unreliable links in a network, such as in Example 4.2. The system state $H(\mathbf{X})$ is the indicator of the event that certain preselected nodes are connected by functioning links. Suppose that we need to estimate the system's unreliability $\overline{r} = \mathbb{P}(H(\mathbf{X}) = 0)$.

To apply the conditional Monte Carlo idea, we view the static network as a snapshot of a *dynamic* network at time $t = 1$. In this dynamic system, the links are repaired independently of each other with an exponential repair time rate of $\mu_e = -\ln(q_e), e = 1, \ldots, n$. At time $t = 0$ all links are failed. The state of the links at time t is given by the vector $\mathbf{X}_t$. Note that $\{\mathbf{X}_t, t \geqslant 0\}$ is a Markov jump process with state space $\{0, 1\}^n$. Since the probability of each link e being operational at time $t = 1$ is p_e, the reliability of the dynamic network at time $t = 1$ is exactly the same as the reliability of the original network.

Let Π denote the *order* in which the links become operational, and let $S_0, S_0 + S_1, \ldots, S_0 + \cdots + S_{n-1}$ be the times at which those links are constructed. Π is a random variable that takes values in the space of permutations of the set of links $\mathcal{E} = \{1, \ldots, n\}$ — hence, the name *permutation Monte Carlo*. For any permutation $\pi = (e_1, e_2, \ldots, e_n)$, define $\mathcal{E}_0 = \mathcal{E}$ and $\mathcal{E}_i = \mathcal{E}_{i-1} \setminus \{e_i\}$, $1 \leqslant i \leqslant n - 1$. Thus $\mathcal{E}_i$ corresponds to the set of links that are still failed after i links have been repaired. Let $b = b(\pi)$ be the number of repairs required (in the order defined by π) to bring the network up. This is called the *critical number* or *construction anchor* for π.

From the theory of Markov jump processes (see Section 1.13.5), it follows that

$$\mathbb{P}(\Pi = \pi) = \prod_{i=1}^{n} \frac{\mu_{e_i}}{\lambda_{i-1}} \,, \tag{5.23}$$

where $\lambda_i = \sum_{e \in \mathcal{E}_i} \mu_e$. More important, conditional on Π the sojourn times $S_0, \ldots, S_{n-1}$ are independent and each S_i is exponentially distributed with parameter λ_i, $i = 0, \ldots, n - 1$. By conditioning on Π, we have

$$\overline{r} = \sum_{\pi} \mathbb{P}[H(\mathbf{X}_1) = 0 \,|\, \Pi = \pi] \, \mathbb{P}[\Pi = \pi] = \mathbb{E}[g(\Pi)], \tag{5.24}$$

with

$$g(\pi) = \mathbb{P}[H(\mathbf{X}_1) = 0 \,|\, \Pi = \pi]. \tag{5.25}$$

From the definitions of S_i and b, we see that $g(\pi)$ is equal to the probability that the sum of b independent exponential random variables with rates $\lambda_i, i = 0, 1, \ldots, b-1$

exceeds 1. This can be computed exactly, for example, by using convolutions. Specifically, we have

$$g(\pi) = 1 - F_0 \star \cdots \star F_{b-1}(1)\,,$$

where F_i is the cdf of the $\mathsf{Exp}(\lambda_i)$ distribution, and $\star$ means convolution; that is,

$$F \star G(t) = \int_0^t F(t-x)\,\mathrm{d}G(x)\ .$$

Alternatively, it can be shown (e.g., see [30]) that

$$g(\pi) = (1,0,\ldots,0)\,\mathrm{e}^{A}\,(1,\ldots,1)^\top, \tag{5.26}$$

where A is the matrix with diagonal elements $-\lambda_0,\ldots,-\lambda_{b-1}$ and upper-diagonal elements $\lambda_0,\ldots,\lambda_{b-2}$ and 0 elsewhere. Here e^A is defined as the *matrix exponential* $\sum_{k=0}^{\infty} A^k/k!$.

Let $\Pi_1,\ldots,\Pi_N$ be iid random permutations, each distributed according to Π; then

$$\widehat{\bar{r}} = \frac{1}{N}\sum_{k=1}^{N} g(\Pi_k) \tag{5.27}$$

is an unbiased estimator for $\bar{r}$. This leads to the following algorithm for estimating the unreliability $\bar{r}$:

Algorithm 5.4.2: Permutation Monte Carlo

input : Structure function H, component unreliabilities $\{q_i\}$, sample size N.
output: Estimator $\widehat{\bar{r}}$ of the system unreliability $\bar{r}$.

for $k = 1$ **to** N **do**
  Draw a random permutation Π according to (5.23). A simple way, similar to Algorithm 2.10.1, is to draw $Y_e \sim \mathsf{Exp}(\mu_e)$, $e = 1,\ldots,n$ independently and return Π as the indices of the (increasing) ordered values.
  Determine the critical number b and the rates $\lambda_i, i = 1,\ldots,b-1$.
  Evaluate the conditional probability $g(\Pi)$ exactly, for example, via (5.26).
Deliver (5.27) as the estimator for $\bar{r}$.
return $\widehat{\bar{r}}$

5.4.1.2 Conditioning Using Minimal Cuts The second method used to estimate unreliability efficiently, developed by Ross [33], employs the concept of a minimal cut. A state vector $\mathbf{x}$ is called a *cut vector* if $H(\mathbf{x}) = 0$. If in addition $H(\mathbf{y}) = 1$ for all $\mathbf{y} > \mathbf{x}$, then $\mathbf{x}$ is called the *minimal cut vector*. Note that $\mathbf{y} > \mathbf{x}$ means that $y_i \geq x_i,\ i = 1,\ldots,n$, with $y_i > x_i$ for some i. If $\mathbf{x}$ is a minimal cut vector, the set $C = \{i : x_i = 0\}$ is called a *minimal cut set*. That is, a minimal cut set is a minimal set of components whose *failure* ensures the failure of the system. If $C_1,\ldots,C_m$ denote all the minimal cut sets, the system is functioning if and only if at least one component in each of the cut sets is functioning. It follows that $H(\mathbf{x})$ can be written as

$$H(\mathbf{x}) = \prod_{j=1}^{m} \max_{i\in C_j} x_i = \prod_{j=1}^{m} \left(1 - \prod_{i\in C_j} (1-x_i)\right). \tag{5.28}$$

To proceed, we need the following proposition, which is adapted from [33].

Proposition 5.4.1 Let $Y_1, \ldots, Y_m$ be Bernoulli random variables (possibly dependent) with success parameters $a_1, \ldots, a_m$. Define $S = \sum_{j=1}^m Y_j$ and let $a = \mathbb{E}[S] = \sum_{j=1}^m a_j$. Let J be a discrete uniform random variable on $\{1, \ldots, m\}$ independent of $Y_1, \ldots, Y_m$. Last, let R be any random variable that is independent of J. Then

$$\mathbb{P}(J = j \,|\, Y_J = 1) = \frac{a_j}{a}, \quad j = 1, \ldots, m \,, \tag{5.29}$$

and

$$\mathbb{E}[SR] = \mathbb{E}[S]\, \mathbb{E}[R \,|\, Y_J = 1] \,. \tag{5.30}$$

Proof: To derive formula (5.29) write, using Bayes' formula

$$\mathbb{P}(J = j \,|\, Y_J = 1) = \frac{\mathbb{P}(Y_J = 1 \,|\, J = j)\, \mathbb{P}(J = j)}{\sum_{i=1}^m \mathbb{P}(Y_J = 1 \,|\, J = i)\, \mathbb{P}(J = i)} \,.$$

Taking into account that $\mathbb{P}(Y_J = 1 \,|\, J = j) = \mathbb{P}(Y_j = 1 \,|\, J = j) = \mathbb{P}(Y_j = 1) = a_j$, the result follows. To prove (5.30), we write

$$\begin{aligned} \mathbb{E}[SR] &= \sum_{j=1}^m \mathbb{E}[R\, Y_j] = \sum_{j=1}^m \mathbb{E}[R \,|\, Y_j = 1]\, \mathbb{P}(Y_j = 1) \\ &= a \sum_{j=1}^m \mathbb{E}[R \,|\, Y_j = 1]\, \frac{a_j}{a} \,. \end{aligned}$$

Since $a = \mathbb{E}[S]$ and, by (5.29), $\{a_j/a\}$ is the conditional distribution of J given $Y_J = 1$, (5.30) follows. □

We will apply Proposition 5.4.1 to the estimation of the unreliability $\bar{r} = \mathbb{P}(H(\mathbf{X}) = 0)$. Let $Y_j = \prod_{i \in C_j} (1 - X_i)$, $j = 1, \ldots, m$, where, as before, $\{C_j\}$ denotes the collection of minimal cut sets. Thus, Y_j is the indicator of the event that all components in C_j are *failed*. Note that $Y_j \sim \mathsf{Ber}(a_j)$, with

$$a_j = \prod_{i \in C_j} q_i \,. \tag{5.31}$$

Let $S = \sum_{j=1}^m Y_j$ and $a = \mathbb{E}[S] = \sum_{j=1}^m a_j$. By (5.28) we have $\bar{r} = \mathbb{P}(S > 0)$, and by (5.30) it follows that

$$\bar{r} = \mathbb{E}[S]\, \mathbb{E}\left[\frac{I_{\{S>0\}}}{S} \,\middle|\, Y_J = 1 \right] = \mathbb{E}\left[\frac{a}{S} \,\middle|\, Y_J = 1 \right],$$

where conditional on $Y_J = 1$ the random variable J takes the value j with probability a_j/a for $j = 1, \ldots, m$. This leads to the following algorithm for estimating the unreliability $\bar{r}$:

Algorithm 5.4.3: Conditioning via Minimal Cuts

input : Minimal cut sets $C_1, \ldots, C_m$, component reliabilities $\{p_i\}$, sample size N.

output: Estimator $\widehat{\bar{r}}$ of the system unreliability $\bar{r}$.

for $k = 1$ **to** N **do**
- Simulate random variable J according to $\mathbb{P}(J = j) = a_j/a,\ j = 1, \ldots, m$.
- Set X_i equal to 0 for all $i \in C_J$ and generate the values of all other $X_i,\ i \notin C_J$ from their corresponding $\mathsf{Ber}(p_i)$ distributions.
- Let S_k be the number of minimal cut sets that have all their components failed (note that $S_k \geqslant 1$).

Set $\widehat{\bar{r}} \leftarrow N^{-1} \sum_{k=1}^{N} a/S_k$ as an estimator of $\bar{r} = \mathbb{P}(S > 0)$.

return $\widehat{\bar{r}}$

It is readily seen that when a, the mean number of failed minimal cuts, is very small, the resulting estimator $\frac{a}{S}$ will have a very small variance. In addition, we could apply importance sampling to the conditional estimator $\frac{a}{S}$ to further reduce the variance.

5.5 STRATIFIED SAMPLING

Stratified sampling is closely related to both the composition method of Section 2.3.3 and the conditional Monte Carlo method discussed in the previous section. As always, we wish to estimate

$$\ell = \mathbb{E}[H(\mathbf{X})] = \int H(\mathbf{x}) f(\mathbf{x})\, \mathrm{d}\mathbf{x}\,.$$

Suppose that $\mathbf{X}$ can be generated via the composition method. Thus, we assume that there exists a random variable Y taking values in $\{1, \ldots, m\}$, say, with known probabilities $\{p_i,\ i = 1, \ldots, m\}$, and we assume that it is easy to sample from the conditional distribution of $\mathbf{X}$ given Y. The events $\{Y = i\}, i = 1, \ldots, m$ form disjoint subregions, or *strata* (singular: stratum), of the sample space Ω, hence the name *stratification*. We use the conditioning formula (1.11) and write

$$\ell = \mathbb{E}[\mathbb{E}[H(\mathbf{X}) \,|\, Y]] = \sum_{i=1}^{m} p_i\, \mathbb{E}[H(\mathbf{X}) \,|\, Y = i]\,. \tag{5.32}$$

This representation suggests that we can estimate ℓ via the following *stratified sampling estimator*

$$\widehat{\ell}^{\,s} = \sum_{i=1}^{m} p_i\, \frac{1}{N_i} \sum_{j=1}^{N_i} H(\mathbf{X}_{ij})\,, \tag{5.33}$$

where $\mathbf{X}_{ij}$ is the j-th observation from the conditional distribution of $\mathbf{X}$ given $Y = i$. Here N_i is the sample size assigned to the i-th stratum. The variance of the stratified sampling estimator is given by

$$\mathrm{Var}\left(\widehat{\ell}^{\,s}\right) = \sum_{i=1}^{m} \frac{p_i^2}{N_i} \mathrm{Var}(H(\mathbf{X}) \,|\, Y = i) = \sum_{i=1}^{m} \frac{p_i^2\, \sigma_i^2}{N_i}\,, \tag{5.34}$$

where $\sigma_i^2 = \mathrm{Var}(H(\mathbf{X}) \,|\, Y = i)$.

How the strata should be chosen depends very much on the problem at hand. However, for a given particular choice of the strata, the sample sizes $\{N_i\}$ can be obtained in an optimal manner, as given in the next theorem.

Theorem 5.5.1 (Stratified Sampling) *Assuming that a maximum number of N samples can be collected, that is, $\sum_{i=1}^m N_i = N$, the optimal value of N_i is given by*

$$N_i^* = N \, \frac{p_i \, \sigma_i}{\sum_{j=1}^m p_j \, \sigma_j} \,, \tag{5.35}$$

which gives a minimal variance of

$$Var(\widehat{\ell}^{*s}) = \frac{1}{N} \left[\sum_{i=1}^m p_i \, \sigma_i \right]^2 \,. \tag{5.36}$$

Proof: The proof is straightforward and uses Lagrange multipliers; it is left as an exercise to the reader (see Problem 5.10). □

Theorem 5.5.1 asserts that the minimal variance of $\widehat{\ell}^s$ is attained for sample sizes N_i that are proportional to $p_i \, \sigma_i$. A difficulty is that although the probabilities p_i are assumed to be known, the standard deviations $\{\sigma_i\}$ are usually unknown. In practice, one would estimate the $\{\sigma_i\}$ from "pilot" runs and then proceed to estimate the optimal sample sizes, N_i^*, from (5.35).

A simple stratification procedure that can achieve variance reduction without requiring prior knowledge of σ_i^2 and $H(\mathbf{X})$, is presented next.

Proposition 5.5.1 *Let the sample sizes N_i be proportional to p_i, that is, $N_i = p_i \, N, \; i = 1, \ldots m$. Then*

$$Var(\widehat{\ell}^s) \leqslant Var(\widehat{\ell}) \,.$$

Proof: Substituting $N_i = p_i \, N$ in (5.34) yields $\mathrm{Var}(\widehat{\ell}^s) = \frac{1}{N} \sum_{i=1}^m p_i \, \sigma_i^2$. The result now follows from

$$N \, \mathrm{Var}(\widehat{\ell}) = \mathrm{Var}(H(\mathbf{X})) \geqslant \mathbb{E}[\mathrm{Var}(H(\mathbf{X}) \,|\, Y)] = \sum_{i=1}^m p_i \, \sigma_i^2 = N \, \mathrm{Var}(\widehat{\ell}^s),$$

where we have used (5.21) in the inequality. □

Proposition 5.5.1 states that the estimator $\widehat{\ell}^s$ is more accurate than the CMC estimator $\widehat{\ell}$. It effects stratification by favoring those events $\{Y = i\}$ whose probabilities p_i are largest. Intuitively, this cannot, in most cases, be an optimal assignment, since information on σ_i^2 and $H(\mathbf{X})$ is ignored.

In the special case of equal weights ($p_i = 1/m$ and $N_i = N/m$), the estimator (5.33) reduces to

$$\widehat{\ell}^s = \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^{N/m} H(\mathbf{X}_{ij}) \,, \tag{5.37}$$

and the method is known as the *systematic sampling method* (e.g., see Cochran [9]).

5.6 MULTILEVEL MONTE CARLO

When estimating the expected value $\ell = \mathbb{E}[Y]$ of a functional $Y = H(\mathbf{X})$ of a diffusion process $\mathbf{X} = \{X_t\}$ (see Section 2.9) by simulation, there are typically two sources of error: estimation error and bias. If an iid samples $Y_1, \ldots, Y_N$ can be simulated *exactly* from the distribution of Y, then their sample mean is an estimator of $\mathbb{E}[Y]$, and the estimation error, as usual, is expressed in terms of the sample variance of the $\{Y_i\}$. However, it is often not possible to obtain exact copies of Y because Y could depend on the whole (continuous) path of the diffusion process. For example, $\mathbf{X}$ could be a Wiener process on the interval $[0, 1]$ and Y its maximum value. Simulating the whole process is not feasible, but it is easy to simulate the process at any grid points $0, 1/n, 2/n, \ldots, n/n$, via Algorithm 2.8.1 and approximate $Y = \max_{0 \leqslant t \leqslant 1} X_t$ with $\widetilde{Y} = \max_{i=0,\ldots,n} X_{i/n}$. However, since $\mathbb{E}[\widetilde{Y}] < \mathbb{E}[Y]$, this introduces a bias. This bias goes to zero as the grid gets finer; that is, as $n \to \infty$. A similar bias is introduced when simulating a stochastic differential equation via Euler's method; see Algorithm 2.9.1.

Thus, to obtain a small overall error, both the number of grid points, $n+1$, and the sample size, N, have to be large enough. However, taking both n and N large, say $N = 10^6$ and $n = 10^4$, may lead to very slow simulations. The multilevel Monte Carlo method of Giles [16] significantly reduces the computational effort required to obtain both small bias and estimation error, by simulating the process at *multiple* grids. As such, the method resembles the multigrid methods in numerical analysis [43].

To explain the multilevel methodology, we consider first the two-level case, which uses a fine and a coarse grid. Let $\mathbf{X}^{(2)}$ be an approximation of $\mathbf{X}$ simulated on the fine grid, and let $Y^{(2)} = H(\mathbf{X}^{(2)})$ be the corresponding performance. Using the *same* path we can generate an approximation $\mathbf{X}^{(1)}$ of $\mathbf{X}$ on the coarse grid. For example, Figure 5.2 shows two simulated paths, $\mathbf{X}^{(2)}$ (solid, black) and $\mathbf{X}^{(1)}$ (dashed, red), evaluated at a fine grid and a coarse grid. The coarse grid is a subset of the fine grid, and $\mathbf{X}^{(1)}$ is simply taken as the subvector of $\mathbf{X}^{(2)}$ evaluated at the coarse grid points. Let $Y^{(1)}$ be the corresponding performance.

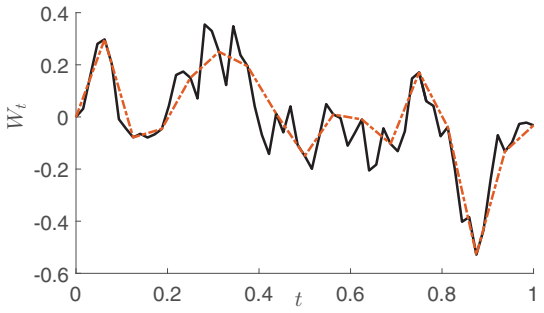

Figure 5.2: The solid (black) line represents an approximation of the Wiener process evaluated at a grid with step size 1/64. The dashed (red) path shows the coarsened version on a grid with step size 1/16. The performances $Y^{(1)}$ and $Y^{(2)}$ could, for example, be the maximum heights of the paths.

Obviously, $Y^{(2)}$ and $Y^{(1)}$ are highly positively correlated. Hence, the latter could be used as a *control variable* for the former. This suggests the control variable estimator

$$\frac{1}{N_2}\sum_{i=1}^{N_2}\left\{Y_i^{(2)} - \alpha\left(Y_i^{(1)} - \mathbb{E}\left[Y^{(1)}\right]\right)\right\},$$

where $\{(Y_i^{(1)}, Y_i^{(2)})\}$ are N_2 independent copies of $(Y^{(1)}, Y^{(2)})$. Note that any value for α will lead to variance reduction if the expectation $\mathbb{E}[Y^{(1)}]$ is known. Unfortunately, the expectation is not known. However, it can be replaced with an estimator

$$\frac{1}{N_1}\sum_{i=1}^{N_1}\widetilde{Y}_i^{(1)}$$

evaluated at the coarser level, and independent of $\{(Y_i^{(1)}, Y_i^{(2)})\}$. If, further, α is set to 1, then the estimator for $\ell = \mathbb{E}[Y]$ reduces to

$$\frac{1}{N_1}\sum_{i=1}^{N_1}\widetilde{Y}_i^{(1)} + \frac{1}{N_2}\sum_{i=1}^{N_2}\left(Y_i^{(2)} - Y_i^{(1)}\right) \stackrel{\text{def}}{=} Z_1 + Z_2\ ,$$

where Z_1 and Z_2 are independent. The first term, Z_1, estimates the expected performance at the coarse level, $\mathbb{E}[Y^{(1)}]$, and Z_2 estimates the expected bias between the fine and coarse level performances; that is, $\mathbb{E}[Y^{(2)}] - \mathbb{E}[Y^{(1)}]$. Since, the difference between $\mathbb{E}[Y^{(2)}]$ and $\mathbb{E}[Y]$ may still be significant, it may be advantageous to use $K > 2$ grids, with step sizes $1/n_1 > \cdots > 1/n_K$, say. This leads to the K-level estimator

$$\widehat{\ell} = Z_1 + Z_2 + \cdots + Z_K,$$

where $Z_1, \ldots, Z_K$ are independent of each other, with

$$Z_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} Y_i^{(1)}$$

and

$$Z_k = \frac{1}{N_k}\sum_{i=1}^{N_k}\left(Y_i^{(k)} - Y_i^{(k-1)}\right), \quad k = 2, \ldots, K\ ,$$

where each $Y_i^{(k)}$ is distributed as $Y^{(k)} = H(\mathbf{X}^{(k)})$, and $\mathbf{X}^{(k)}$ is a path obtained using the k-th level grid. Note that both Z_k and Z_{k-1} contain variables $\{Y_i^{(k-1)}\}$. It is important to realize that these are obtained from *different* and independent simulations. For the 2-level case we emphasized this difference by using $Y_i^{(1)}$ and $\widetilde{Y}_i^{(1)}$, respectively.

For simplicity, we assume that the process $\mathbf{X}$ is to be simulated on the interval $[0, 1]$ and that the k-level grid points are $0, 1/n_k, 2/n_k, \ldots, 1$, with $n_k = M^k$, where M is a small integer, such as $M = 4$. It is useful to take the sample size $\{N_k\}$ proportional to the step sizes $\{1/n_k\}$; but see also Remark 5.6.1. This leads to the following multilevel Monte Carlo algorithm:

Algorithm 5.6.1: Multilevel Monte Carlo

input : Mesh factor M, number of levels K, simulation effort N.
output: Estimator $\widehat{\ell}$ of $\ell = \mathbb{E}[H(\mathbf{X})]$.

for $k = 1$ **to** K **do**
  $n_k \leftarrow M^k$
  $N_k \leftarrow \lceil N/n_k \rceil$
for $i = 1$ **to** N_1 **do**
  Generate $\mathbf{X}_i^{(1)}$ `// the i-th path at the coarsest level`
  $Y_i^{(1)} \leftarrow H(\mathbf{X}_i^{(1)})$
$Z_1 \leftarrow \sum_{i=1}^{N_1} Y_i^{(1)}/N_1$
for $k = K$ **to** 2 **do**
  for $i = 1$ **to** N_k **do**
    Generate $\mathbf{X}_i^{(k)}$ `// the i-th path at level k`
    Coarsen $\mathbf{X}_i^{(k)}$ to $\mathbf{X}_i^{(k-1)}$
    $Y_i^{(k)} \leftarrow H(\mathbf{X}_i^{(k)})$
    $Y_i^{(k-1)} \leftarrow H(\mathbf{X}_i^{(k-1)})$
  $Z_k \leftarrow \sum_{i=1}^{N_k} \left(Y_i^{(k)} - Y_i^{(k-1)} \right) / N_k$
$\widehat{\ell} \leftarrow \sum_{k=1}^{K} Z_k$
return $\widehat{\ell}$

EXAMPLE 5.8

We estimate the expected maximum of a Wiener process in the interval [0,1] via Algorithm 5.6.1, using the following parameters: $M = 4$, $K = 10$, and $N = 10^6$. Figure 5.2 depicts typical pairs of paths that are generated at level $k = 3$ of the algorithm. The solid (black) path is generated on a grid with step size $1/4^3 = 1/64$ and the dashed (red) path is the coarsened version, on the subgrid with step size $1/4^2 = 1/16$.

Typical outcomes of $Z_1, \ldots, Z_{10}$ are $0.557, 0.1094, 0.0629, 0.0326, 0.0178$, $0.0086, 0.00429, 0.0020, 0.0023$, and 0, which suggests that $K = 10$ is high enough to eliminate most of the bias. Adding the $\{Z_k\}$ gives the estimate $\widehat{\ell} = 0.7967$, which is close the exact value $\ell = \sqrt{2/\pi} \approx 0.7978$ (e.g., [24]). The simulation time was 1.5 seconds. Note that the majority of paths (250,000 out of a total of 333,337) are simulated on the coarsest grid, with $M + 1 = 5$ points, whereas the finest grid has only 1 simulation. In contrast, simulating 333,337 paths at the highest resolution (1,048,577 points) would take a very long time.

Remark 5.6.1 (Choice of Parameters) The optimal choice for the number of levels K, grid factor M, and simulation effort N is largely problem dependent. Commonly used values for M are $M = 4$ and $M = 2$. For a given M, the number of levels K should be chosen large enough such that $\mathbb{E}[Y^{(K)}]$ is close enough to $\mathbb{E}[Y]$. This can be assessed by investigating how fast $\{Z_k, k = 2, 3, \ldots, K\}$ converges to zero. The parameter N regulates the overall simulation effort. By increasing N, the accuracy of the estimator is improved. In [16] an asymptotic complexity analysis is given for the multilevel Monte Carlo estimator $\widehat{\ell}$ as $K \to \infty$. The analysis suggests that the optimal sample size N_k should be chosen proportional to $\sqrt{V_k/n_k}$, where V_k is the variance of Z_k. The $\{V_k\}$ could be estimated via a pilot run.

5.7 IMPORTANCE SAMPLING

The most fundamental variance reduction technique is *importance sampling*. As we will see below, importance sampling often leads to a dramatic variance reduction (sometimes on the order of millions, in particular when estimating rare event probabilities), while with all of the above-mentioned variance reduction techniques only a moderate reduction, typically up to 10-fold, can be achieved. Importance sampling involves choosing a sampling distribution that favors important samples. Let, as before,

$$\ell = \mathbb{E}_f[H(\mathbf{X})] = \int H(\mathbf{x})\, f(\mathbf{x})\, \mathrm{d}\mathbf{x}\,, \tag{5.38}$$

where H is the sample performance and f is the probability density of $\mathbf{X}$. For reasons that will become clear shortly, we add a subscript f to the expectation to indicate that it is taken with respect to the density f.

Let g be another probability density such that $H\,f$ is *dominated* by g. That is, $g(\mathbf{x}) = 0 \;\Rightarrow\; H(\mathbf{x})\, f(\mathbf{x}) = 0$. Using the density g, we can represent ℓ as

$$\ell = \int H(\mathbf{x})\, \frac{f(\mathbf{x})}{g(\mathbf{x})}\, g(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathbb{E}_g\left[H(\mathbf{X})\, \frac{f(\mathbf{X})}{g(\mathbf{X})}\right], \tag{5.39}$$

where the subscript g means that the expectation is taken with respect to g. Such a density is called the *importance sampling* density, *proposal* density, or *instrumental* density (as we use g as an instrument to obtain information about ℓ). Consequently, if $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is a *random sample* from g, that is, $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are iid random vectors with density g, then

$$\widehat{\ell} = \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k)\, \frac{f(\mathbf{X}_k)}{g(\mathbf{X}_k)} \tag{5.40}$$

is an unbiased estimator of ℓ. This estimator is called the *importance sampling estimator*. The ratio of densities,

$$W(\mathbf{x}) = \frac{f(\mathbf{x})}{g(\mathbf{x})}\,, \tag{5.41}$$

is called the *likelihood ratio*. For this reason the importance sampling estimator is also called the *likelihood ratio estimator*. In the particular case where there is no change of measure, that is, $g = f$, we have $W = 1$, and the likelihood ratio estimator in (5.40) reduces to the usual CMC estimator.

5.7.1 Weighted Samples

The likelihood ratios need only be known *up to a constant*, that is, $W(\mathbf{X}) = c\, w(\mathbf{X})$ for some known function $w(\cdot)$. Since $\mathbb{E}_g[W(\mathbf{X})] = 1$, we can write $\ell = \mathbb{E}_g[H(\mathbf{X})\, W(\mathbf{X})]$ as

$$\ell = \frac{\mathbb{E}_g[H(\mathbf{X})\, W(\mathbf{X})]}{\mathbb{E}_g[W(\mathbf{X})]}\,.$$

This suggests, as an alternative to the standard likelihood ratio estimator (5.41), the following *weighted sample estimator*:

$$\widehat{\ell}_w = \frac{\sum_{k=1}^{N} H(\mathbf{X}_k)\, w_k}{\sum_{k=1}^{N} w_k}\,. \tag{5.42}$$

Here, the $\{w_k\}$, with $w_k = w(\mathbf{X}_k)$, are interpreted as *weights* of the random sample $\{\mathbf{X}_k\}$, and the sequence $\{(\mathbf{X}_k, w_k)\}$ is called a *weighted (random) sample* from $g(\mathbf{x})$. Similar to the regenerative ratio estimator in Chapter 4, the weighted sample estimator (5.42) introduces some bias, which tends to 0 as N increases. Loosely speaking, we may view the weighted sample $\{(\mathbf{X}_k, w_k)\}$ as a representation of $f(\mathbf{x})$ in the sense that $\ell = \mathbb{E}_f[H(\mathbf{X})] \approx \widehat{\ell}_w$ for any function $H(\cdot)$.

5.7.2 Variance Minimization Method

Since the choice of the importance sampling density g is crucially linked to the variance of the estimator $\widehat{\ell}$ in (5.40), we consider next the problem of minimizing the variance of $\widehat{\ell}$ with respect to g, that is,

$$\min_g \mathrm{Var}_g \left(H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right). \tag{5.43}$$

It is not difficult to prove (e.g., see Rubinstein and Melamed [37] and Problem 5.14) that the solution of the problem (5.43) is

$$g^*(\mathbf{x}) = \frac{|H(\mathbf{x})|\, f(\mathbf{x})}{\int |H(\mathbf{x})|\, f(\mathbf{x})\, \mathrm{d}\mathbf{x}}\,. \tag{5.44}$$

In particular, if $H(\mathbf{x}) \geqslant 0$ — which we will assume from now on — then

$$g^*(\mathbf{x}) = \frac{H(\mathbf{x})\, f(\mathbf{x})}{\ell} \tag{5.45}$$

and

$$\mathrm{Var}_{g^*}(\widehat{\ell}) = \mathrm{Var}_{g^*}(H(\mathbf{X})W(\mathbf{X})) = \mathrm{Var}_{g^*}(\ell) = 0\,.$$

The density g^* as per (5.44) and (5.45) is called the *optimal importance sampling density*.

■ **EXAMPLE 5.9**

Let $X \sim \mathsf{Exp}(u^{-1})$ and $H(X) = I_{\{X \geqslant \gamma\}}$ for some $\gamma > 0$. Let f denote the pdf of X. Consider the estimation of

$$\ell = \mathbb{E}_f[H(X)] = \int_\gamma^\infty u^{-1}\, \mathrm{e}^{-x\,u^{-1}}\, \mathrm{d}x = \mathrm{e}^{-\gamma\,u^{-1}}.$$

We have

$$g^*(x) = H(x)\, f(x)\, \ell^{-1} = I_{\{x \geqslant \gamma\}}\, u^{-1}\, \mathrm{e}^{-x\,u^{-1}}\, \mathrm{e}^{\gamma u^{-1}} = I_{\{x \geqslant \gamma\}}\, u^{-1}\, \mathrm{e}^{-(x-\gamma)\,u^{-1}}.$$

Thus, the optimal importance sampling distribution of X is the *shifted* exponential distribution. Note that Hf is dominated by g^* but f itself is not dominated by g^*. Since g^* is optimal, the likelihood ratio estimator $\widehat{\ell}$ is constant. Namely, with $N = 1$,

$$\widehat{\ell} = H(X)\, W(X) = \frac{H(X) f(X)}{H(X) f(X)/\ell} = \ell\,.$$

It is important to realize that, although (5.40) is an unbiased estimator for *any* pdf g dominating $H\,f$, not all such pdfs are appropriate. One of the main rules for choosing a good importance sampling pdf is that the estimator (5.40) should have finite variance. This is equivalent to the requirement that

$$\mathbb{E}_g\left[H^2(\mathbf{X})\frac{f^2(\mathbf{X})}{g^2(\mathbf{X})}\right] = \mathbb{E}_f\left[H^2(\mathbf{X})\frac{f(\mathbf{X})}{g(\mathbf{X})}\right] < \infty\,. \tag{5.46}$$

This suggests that g should not have a "lighter tail" than f and that, preferably, the likelihood ratio, f/g, should be bounded.

In general, implementation of the optimal importance sampling density g^* as per (5.44) and (5.45) is problematic. The main difficulty lies in the fact that, to derive $g^*(\mathbf{x})$, we need to know ℓ. But ℓ is precisely the quantity we want to estimate from the simulation!

In most simulation studies the situation is even worse, since the analytical expression for the sample performance H is unknown in advance. To overcome this difficulty, we could perform a pilot run with the underlying model, obtain a sample $H(\mathbf{X}_1),\ldots,H(\mathbf{X}_N)$, and then use it to estimate g^*. It is important to note that sampling from such an artificially constructed density can be a very complicated and time-consuming task, especially when g is a high-dimensional density.

Remark 5.7.1 (Degeneracy of the Likelihood Ratio Estimator) The likelihood ratio estimator $\widehat{\ell}$ in (5.40) suffers from a form of degeneracy in the sense that the distribution of $W(\mathbf{X})$ under the importance sampling density g may become increasingly skewed as the dimensionality n of $\mathbf{X}$ increases. That is, $W(\mathbf{X})$ may take values close to 0 with high probability, but may also take very large values with a small though significant probability. As a consequence, the variance of $W(\mathbf{X})$ under g may become very large for large n. As an example of this degeneracy, assume, for simplicity, that the components in $\mathbf{X}$ are iid, under both f and g. Hence, both $f(\mathbf{x})$ and $g(\mathbf{x})$ are the products of their marginal pdfs. Suppose that the marginal pdfs of each component X_i are f_1 and g_1, respectively. We can then write $W(\mathbf{X})$ as

$$W(\mathbf{X}) = \exp\sum_{i=1}^{n}\ln\frac{f_1(X_i)}{g_1(X_i)}\,. \tag{5.47}$$

Using the law of large numbers, the random variable $\sum_{i=1}^{n}\ln\left(f_1(X_i)/g_1(X_i)\right)$ is approximately equal to $n\,\mathbb{E}_{g_1}[\ln\left(f_1(X)/g_1(X)\right)]$ for large n. Hence,

$$W(\mathbf{X}) \approx \exp\left\{-n\,\mathbb{E}_{g_1}\left[\ln\left(\frac{g_1(X)}{f_1(X)}\right)\right]\right\}. \tag{5.48}$$

Since $\mathbb{E}_{g_1}\left[\ln(g_1(\mathbf{X})/f_1(\mathbf{X}))\right]$ is nonnegative (see page 31), the likelihood ratio $W(\mathbf{X})$ tends to 0 as $n\to\infty$. However, by definition, the expectation of $W(\mathbf{X})$ under g is always 1. This indicates that the distribution of $W(\mathbf{X})$ becomes increasingly skewed when n gets large. Several methods have been introduced to prevent this degeneracy. Examples are the heuristics of Doucet et al. [11], Liu [27], and Robert and Casella [32] and the so-called screening method. The last will be presented in Sections 5.12 and 8.2.2 and can be considered as a dimension-reduction technique.

When the pdf f belongs to some parametric family of distributions, it is often convenient to choose the importance sampling distribution from the *same* family.

In particular, suppose that $f(\cdot) = f(\cdot; \mathbf{u})$ belongs to the family

$$\mathscr{F} = \{f(\cdot; \mathbf{v}),\ \mathbf{v} \in \mathscr{V}\} .$$

Then the problem of finding an optimal importance sampling density in this class reduces to the following *parametric* minimization problem:

$$\min_{\mathbf{v} \in \mathscr{V}} \mathrm{Var}_{\mathbf{v}} \left(H(\mathbf{X})\, W(\mathbf{X}; \mathbf{u}, \mathbf{v})\right), \tag{5.49}$$

where $W(\mathbf{X}; \mathbf{u}, \mathbf{v}) = f(\mathbf{X}; \mathbf{u})/f(\mathbf{X}; \mathbf{v})$. We will call the vector $\mathbf{v}$ the *reference parameter vector* or *tilting vector*. Since under $f(\cdot; \mathbf{v})$ the expectation $\ell = \mathbb{E}_{\mathbf{v}}[H(\mathbf{X})\, W(\mathbf{X}; \mathbf{u}, \mathbf{v})]$ is constant, the optimal solution of (5.49) coincides with that of

$$\min_{\mathbf{v} \in \mathscr{V}} V(\mathbf{v}) , \tag{5.50}$$

where

$$V(\mathbf{v}) = \mathbb{E}_{\mathbf{v}}[H^2(\mathbf{X})\, W^2(\mathbf{X}; \mathbf{u}, \mathbf{v})] = \mathbb{E}_{\mathbf{u}}[H^2(\mathbf{X})\, W(\mathbf{X}; \mathbf{u}, \mathbf{v})] . \tag{5.51}$$

We will call either of the equivalent problems (5.49) and (5.50) the *variance minimization* (VM) problem, and we will call the parameter vector ${}_*\mathbf{v}$ that minimizes programs (5.49)–(5.50) the *optimal VM reference parameter vector*. We refer to $\mathbf{u}$ as the *nominal* parameter.

The sample average version of (5.50)–(5.51) is

$$\min_{\mathbf{v} \in \mathscr{V}} \widehat{V}(\mathbf{v}) , \tag{5.52}$$

where

$$\widehat{V}(\mathbf{v}) = \frac{1}{N} \sum_{k=1}^{N} [H^2(\mathbf{X}_k)\, W(\mathbf{X}_k; \mathbf{u}, \mathbf{v})] , \tag{5.53}$$

and the sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is from $f(\mathbf{x}; \mathbf{u})$. Note that as soon as the sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is available, the function $\widehat{V}(\mathbf{v})$ becomes a deterministic one.

Since in typical applications both functions $V(\mathbf{v})$ and $\widehat{V}(\mathbf{v})$ are convex and differentiable with respect to $\mathbf{v}$, and since one can typically interchange the expectation and differentiation operators (see Rubinstein and Shapiro [38]), the solutions of programs (5.50)–(5.51) and (5.52)–(5.53) can be obtained by solving (with respect to $\mathbf{v}$) the following system of equations:

$$\mathbb{E}_{\mathbf{u}}[H^2(\mathbf{X})\, \nabla W(\mathbf{X}; \mathbf{u}, \mathbf{v})] = \mathbf{0} \tag{5.54}$$

and

$$\frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k)\, \nabla W(\mathbf{X}_k; \mathbf{u}, \mathbf{v}) = \mathbf{0} , \tag{5.55}$$

respectively, where

$$\nabla W(\mathbf{X}; \mathbf{u}, \mathbf{v}) = \nabla \frac{f(\mathbf{X}; \mathbf{u})}{f(\mathbf{X}; \mathbf{v})} = [\nabla \ln f(\mathbf{X}; \mathbf{v})]\ W(\mathbf{X}; \mathbf{u}, \mathbf{v}) ,$$

the gradient is with respect to $\mathbf{v}$ and the function $\nabla \ln f(\mathbf{x}; \mathbf{v})$ is the score function; see (1.57). Note that the system of nonlinear equations (5.55) is typically solved using numerical methods.

EXAMPLE 5.10

Consider estimating $\ell = \mathbb{E}[X]$, where $X \sim \mathsf{Exp}(u^{-1})$. Choosing $f(x;v) = v^{-1}\exp(-xv^{-1}), x \geqslant 0$ as the importance sampling pdf, the program (5.50) reduces to

$$\min_v V(v) = \min_v \frac{v}{u^2}\int_0^\infty x^2\,\mathrm{e}^{-(2u^{-1}-v^{-1})x}\,\mathrm{d}x = \min_{v\geqslant u/2}\frac{2uv^4}{(2v-u)^3}\ .$$

The optimal reference parameter ${}_*v$ is given by

$${}_*v = 2\,u\ .$$

We see that ${}_*v$ is exactly two times larger than u. Solving the sample average version (5.55) (numerically), one should find that, for large N, its optimal solution ${}_*\widehat{v}$ will be close to the true parameter ${}_*v$.

EXAMPLE 5.11 Example 5.9 (Continued)

Consider again estimating $\ell = \mathbb{P}_u(X \geqslant \gamma) = \exp(-\gamma u^{-1})$. In this case, using the family $\{f(x;v), v > 0\}$ defined by $f(x;v) = v^{-1}\exp(xv^{-1}), x \geqslant 0$, we can reduce the program (5.50) to

$$\min_v V(v) = \min_v \frac{v}{u^2}\int_\gamma^\infty \mathrm{e}^{-(2u^{-1}-v^{-1})x}\,\mathrm{d}x = \min_{v\geqslant u/2}\frac{v^2}{u}\,\frac{\mathrm{e}^{-\gamma(2u^{-1}-v^{-1})}}{(2v-u)}\ .$$

The optimal reference parameter ${}_*v$ is given by

$${}_*v = \frac{1}{2}\left\{\gamma + u + \sqrt{\gamma^2+u^2}\right\} = \gamma + \frac{u}{2} + \mathcal{O}((u/\gamma)^2)\ ,$$

where $\mathcal{O}(x^2)$ is a function of x such that

$$\lim_{x\to 0}\frac{\mathcal{O}(x^2)}{x^2} = \text{constant}\ .$$

We see that for $\gamma \gg u$, ${}_*v$ is approximately equal to γ.

It is important to note that in this case the sample version (5.55) (or (5.52) – (5.53)) is meaningful only for small γ, in particular for those γ for which ℓ is *not a rare-event probability*, say where $\ell > 10^{-4}$. For very small ℓ, a tremendously large sample N is needed (because of the indicator function $I_{\{X\geqslant\gamma\}}$), and thus the importance sampling estimator $\widehat{\ell}$ is useless. We will discuss the estimation of rare-event probabilities in more detail in Chapter 8.

Observe that the VM problem (5.50) can also be written as

$$\min_{\mathbf{v}\in\mathscr{V}} V(\mathbf{v}) = \min_{\mathbf{v}\in\mathscr{V}} \mathbb{E}_{\mathbf{w}}\left[H^2(\mathbf{X})\,W(\mathbf{X};\mathbf{u},\mathbf{v})\,W(\mathbf{X};\mathbf{u},\mathbf{w})\right], \tag{5.56}$$

where $\mathbf{w}$ is an arbitrary reference parameter. Note that (5.56) is obtained from (5.51) by multiplying and dividing the integrand by $f(\mathbf{x};\mathbf{w})$. We now replace the expected value in (5.56) by its sample (stochastic) counterpart and then take the

optimal solution of the associated Monte Carlo program as an estimator of $_*\mathbf{v}$. Specifically, the stochastic counterpart of (5.56) is

$$\min_{\mathbf{v}\in\mathscr{V}} \widehat{V}(\mathbf{v}) = \min_{\mathbf{v}\in\mathscr{V}} \frac{1}{N}\sum_{k=1}^{N} H^2(\mathbf{X}_k)\, W(\mathbf{X}_k\,;\mathbf{u},\mathbf{v})\, W(\mathbf{X}_k\,;\mathbf{u},\mathbf{w})\,, \tag{5.57}$$

where $\mathbf{X}_1,\ldots,\mathbf{X}_N$ is an iid sample from $f(\cdot;\mathbf{w})$ and $\mathbf{w}$ is an appropriately chosen *trial* parameter. Solving the stochastic program (5.57) thus yields an estimate, say $\widehat{{}_*\mathbf{v}}$, of $_*\mathbf{v}$. In some cases it may be useful to *iterate* this procedure, that is, use $\widehat{{}_*\mathbf{v}}$ as a trial vector in (5.57), to obtain a better estimate.

Once the reference parameter $\mathbf{v} = \widehat{{}_*\mathbf{v}}$ is determined, ℓ is estimated via the likelihood ratio estimator

$$\widehat{\ell} = \frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_k)\, W(\mathbf{X}_k;\mathbf{u},\mathbf{v})\,, \tag{5.58}$$

where $\mathbf{X}_1,\ldots,\mathbf{X}_N$ is a random sample from $f(\cdot;\mathbf{v})$. Typically, the sample size N in (5.58) is larger than that used for estimating the reference parameter. We call (5.58) the *standard likelihood ratio* (SLR) estimator.

5.7.3 Cross-Entropy Method

An alternative approach for choosing an "optimal" reference parameter vector in (5.58) is based on the Kullback–Leibler cross-entropy, or simply *cross-entropy* (CE), mentioned in (1.53). For clarity, we repeat that the CE distance between two pdfs g and h is given (in the continuous case) by

$$\begin{aligned}\mathscr{D}(g,h) &= \mathbb{E}_g\left[\ln\frac{g(\mathbf{X})}{h(\mathbf{X})}\right] = \int g(\mathbf{x})\ln\frac{g(\mathbf{x})}{h(\mathbf{x})}\,\mathrm{d}\mathbf{x}\\ &= \int g(\mathbf{x})\ln g(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int g(\mathbf{x})\ln h(\mathbf{x})\,\mathrm{d}\mathbf{x}\,.\end{aligned} \tag{5.59}$$

Recall that $\mathscr{D}(g,h)\geqslant 0$, with equality if and only if $g=h$.

The general idea is to choose the importance sampling density, say h, such that the CE distance between the optimal importance sampling density g^* in (5.44) and h is minimal. We call this the *CE optimal pdf*. Thus, this pdf solves the following *functional* optimization program:

$$\min_h \mathscr{D}\left(g^*,h\right).$$

If we optimize over all densities h, then it is immediate from $\mathscr{D}(g^*,h)\geqslant 0$ that the CE optimal pdf coincides with the VM optimal pdf g^*.

As with the VM approach in (5.49) and (5.50), we will restrict ourselves to the parametric family of densities $\{f(\cdot;\mathbf{v}),\mathbf{v}\in\mathscr{V}\}$ that contains the "nominal" density $f(\cdot;\mathbf{u})$. The CE method now aims to solve the *parametric* optimization problem

$$\min_{\mathbf{v}} \mathscr{D}\left(g^*, f(\cdot;\mathbf{v})\right).$$

Since the first term on the right-hand side of (5.59) does not depend on $\mathbf{v}$, minimizing the Kullback–Leibler distance between g^* and $f(\cdot;\mathbf{v})$ is equivalent to

maximizing with respect to $\mathbf{v}$,

$$\int H(\mathbf{x})\, f(\mathbf{x};\mathbf{u})\, \ln f(\mathbf{x};\mathbf{v})\, \mathrm{d}\mathbf{x} = \mathbb{E}_{\mathbf{u}}\left[H(\mathbf{X})\, \ln f(\mathbf{X};\mathbf{v})\right],$$

where we have assumed that $H(\mathbf{x})$ is nonnegative. Arguing as in (5.50), we find that the CE optimal reference parameter vector $\mathbf{v}^*$ can be obtained from the solution of the following simple program:

$$\max_{\mathbf{v}} D(\mathbf{v}) = \max_{\mathbf{v}} \mathbb{E}_{\mathbf{u}}\left[H(\mathbf{X}) \ln f(\mathbf{X};\mathbf{v})\right]. \tag{5.60}$$

Since typically $D(\mathbf{v})$ is convex and differentiable with respect to $\mathbf{v}$ (see Rubinstein and Shapiro [38]), the solution to (5.60) may be obtained by solving

$$\mathbb{E}_{\mathbf{u}}\left[H(\mathbf{X})\, \nabla \ln f(\mathbf{X};\mathbf{v})\right] = \mathbf{0}\,, \tag{5.61}$$

provided that the expectation and differentiation operators can be interchanged. The sample counterpart of (5.61) is

$$\frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_k)\, \nabla \ln f(\mathbf{X}_k;\mathbf{v}) = \mathbf{0}\,. \tag{5.62}$$

By analogy to the VM program (5.50), we call (5.60) the *CE program*, and we call the parameter vector $\mathbf{v}^*$ that minimizes the program (5.63) the *optimal CE reference parameter vector*.

Arguing as in (5.56), it is readily seen that (5.60) is equivalent to the following program:

$$\max_{\mathbf{v}} D(\mathbf{v}) = \max_{\mathbf{v}}\, \mathbb{E}_{\mathbf{w}}\left[H(\mathbf{X})\, W(\mathbf{X};\mathbf{u},\mathbf{w}) \ln f(\mathbf{X};\mathbf{v})\right], \tag{5.63}$$

where $W(\mathbf{X};\mathbf{u},\mathbf{w})$ is again the likelihood ratio and $\mathbf{w}$ is an *arbitrary* tilting parameter. Similar to (5.57), we can estimate $\mathbf{v}^*$ as the solution of the stochastic program

$$\max_{\mathbf{v}} \widehat{D}(\mathbf{v}) = \max_{\mathbf{v}}\, \frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_k)\, W(\mathbf{X}_k;\mathbf{u},\mathbf{w})\, \ln f(\mathbf{X}_k;\mathbf{v})\,, \tag{5.64}$$

where $\mathbf{X}_1,\ldots,\mathbf{X}_N$ is a random sample from $f(\cdot;\mathbf{w})$. As in the VM case, we mention the possibility of *iterating* this procedure, that is, using the solution of (5.64) as a trial parameter for the next iteration.

Since in typical applications the function $\widehat{D}$ in (5.64) is convex and differentiable with respect to $\mathbf{v}$ (see [38]), the solution of (5.64) may be obtained by solving (with respect to $\mathbf{v}$) the following system of equations:

$$\frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_k)\, W(\mathbf{X}_k;\mathbf{u},\mathbf{w})\, \nabla \ln f(\mathbf{X}_k;\mathbf{v}) = \mathbf{0}\,, \tag{5.65}$$

where the gradient is with respect to $\mathbf{v}$.

Our extensive numerical studies show that for moderate dimensions n, say $n \leq 50$, the optimal solutions of the CE programs (5.63) and (5.64) (or (5.65)) and their VM counterparts (5.56) and (5.57) are typically nearly the same. However,

for high-dimensional problems ($n > 50$), we found numerically that the importance sampling estimator $\widehat{\ell}$ in (5.58) based on VM updating of $\mathbf{v}$ outperforms its CE counterpart in both variance and bias. The latter is caused by the degeneracy of W, to which, we found, CE is more sensitive.

The advantage of the CE program is that it can often be solved *analytically*. In particular, this happens when the distribution of $\mathbf{X}$ belongs to an *exponential family* of distributions; see Section A.3 of the Appendix. Specifically (see (A.16)), for a one-dimensional exponential family parameterized by the mean, the CE optimal parameter is *always*

$$v^* = \frac{\mathbb{E}_u[H(X)\,X]}{\mathbb{E}_u[H(X)]} = \frac{\mathbb{E}_w[W(X;u,w)\,H(X)\,X]}{\mathbb{E}_w[H(X)\,W(X;u,w)]}\,, \tag{5.66}$$

and the corresponding sample-based updating formula is

$$\widehat{v} = \frac{\sum_{k=1}^N H(X_k)\,W(X_k;u,w)\,X_k}{\sum_{k=1}^N H(X_k)\,W(X_k;u,w)}\,, \tag{5.67}$$

respectively, where $X_1,\ldots,X_N$ is a random sample from the density $f(\cdot;w)$ and w is an arbitrary parameter. The multidimensional version of (5.67) is

$$\widehat{v}_i = \frac{\sum_{k=1}^N H(\mathbf{X}_k)\,W(\mathbf{X}_k;\mathbf{u},\mathbf{w})\,X_{ki}}{\sum_{k=1}^N H(\mathbf{X}_k)\,W(\mathbf{X}_k;\mathbf{u},\mathbf{w})} \tag{5.68}$$

for $i = 1,\ldots,n$, where X_{ki} is the i-th component of vector $\mathbf{X}_k$ and $\mathbf{u}$ and $\mathbf{w}$ are parameter vectors.

Observe that for $\mathbf{u} = \mathbf{w}$ (no likelihood ratio term W), (5.68) reduces to

$$\widehat{v}_i = \frac{\sum_{k=1}^N H(\mathbf{X}_k)\,X_{ki}}{\sum_{k=1}^N H(\mathbf{X}_k)}\,, \tag{5.69}$$

where $\mathbf{X}_k \sim f(\mathbf{x};\mathbf{u})$.

Observe also that because of the degeneracy of W, one would always prefer the estimator (5.69) to (5.68), especially for high-dimensional problems. But as we will see below, this is not always feasible, particularly when estimating rare-event probabilities in Chapter 8.

■ **EXAMPLE 5.12 Example 5.10 (Continued)**

Consider again the estimation of $\ell = \mathbb{E}[X]$, where $X \sim \mathsf{Exp}(u^{-1})$ and $f(x;v) = v^{-1}\exp(xv^{-1}), x \geqslant 0$. Solving (5.61), we find that the optimal reference parameter v^* is equal to

$$v^* = \frac{\mathbb{E}_u[X^2]}{\mathbb{E}_u[X]} = 2\,u\,.$$

Thus, v^* is exactly the same as ${}_*v$. For the sample average of (5.61), we should find that for large N its optimal solution $\widehat{v}^*$ is close to the optimal parameter $v^* = 2u$.

■ **EXAMPLE 5.13 Example 5.11 (Continued)**

Consider again the estimation of $\ell = \mathbb{P}_u(X \geqslant \gamma) = \exp(-\gamma u^{-1})$. In this case, we readily find from (5.66) that the optimal reference parameter is $v^* = \gamma + u$. Note that similar to the VM case, for $\gamma \gg u$, the optimal reference parameter is approximately γ.

Note that in the preceding example, similar to the VM problem, the CE sample version (5.65) is meaningful only when γ is chosen such that ℓ is *not a rare-event probability*, say when $\ell > 10^{-4}$. In Chapter 8 we present a general procedure for estimating rare-event probabilities of the form $\ell = \mathbb{P}_{\mathbf{u}}(S(\mathbf{X}) \geqslant \gamma)$ for an arbitrary function $S(\mathbf{x})$ and level γ.

■ **EXAMPLE 5.14 Finite Support Discrete Distributions**

Let X be a discrete random variable with finite support, that is, X can only take a finite number of values, say $a_1, \ldots, a_m$. Let $u_i = \mathbb{P}(X = a_i), i = 1, \ldots, m$ and define $\mathbf{u} = (u_1, \ldots, u_m)$. The distribution of X is thus trivially parameterized by the vector $\mathbf{u}$. We can write the density of X as

$$f(x; \mathbf{u}) = \sum_{i=1}^{m} u_i \, I_{\{x=a_i\}} \, .$$

From the discussion at the beginning of this section, we know that the optimal CE and VM parameters *coincide*, since we optimize over *all* densities on $\{a_1, \ldots, a_m\}$. From (5.44) the VM (and CE) optimal density is given by

$$\begin{aligned} f(x; \mathbf{v}^*) &= \frac{H(x)\, f(x; \mathbf{u})}{\sum_x H(x)\, f(x; \mathbf{u})} \\ &= \frac{\sum_{i=1}^m H(a_i)\, u_i \, I_{\{x=a_i\}}}{\mathbb{E}_{\mathbf{u}}[H(X)]} \\ &= \sum_{i=1}^{m} \frac{H(a_i)\, u_i}{\mathbb{E}_{\mathbf{u}}[H(X)]} \, I_{\{x=a_i\}} \\ &= \sum_{i=1}^{m} \left(\frac{\mathbb{E}_u[H(X)\, I_{\{X=a_i\}}]}{\mathbb{E}_{\mathbf{u}}[H(X)]} \right) I_{\{x=a_i\}} \, , \end{aligned}$$

so that

$$v_i^* = \frac{\mathbb{E}_{\mathbf{u}}[H(X)\, I_{\{X=a_i\}}]}{\mathbb{E}_{\mathbf{u}}[H(X)]} = \frac{\mathbb{E}_{\mathbf{w}}[H(X)\, W(X; \mathbf{u}, \mathbf{w}) I_{\{X=a_i\}}]}{\mathbb{E}_{\mathbf{w}}[H(X)\, W(X; \mathbf{u}, \mathbf{w})]} \tag{5.70}$$

for any reference parameter $\mathbf{w}$, provided that $\mathbb{E}_{\mathbf{w}}[H(X)\, W(X; \mathbf{u}, \mathbf{w})] > 0$.

The vector $\mathbf{v}^*$ can be estimated from the stochastic counterpart of (5.70), that is, as

$$\widehat{v}_i = \frac{\sum_{k=1}^N H(X_k)\, W(X_k; \mathbf{u}, \mathbf{w}) I_{\{X_k=a_i\}}}{\sum_{k=1}^N H(X_k)\, W(X_k; \mathbf{u}, \mathbf{w})} \, , \tag{5.71}$$

where $X_1, \ldots, X_N$ is an iid sample from the density $f(\cdot; \mathbf{w})$.

A similar result holds for a random vector $\mathbf{X} = (X_1, \ldots, X_n)$, where $X_1, \ldots, X_n$ are independent discrete random variables with finite support, characterized by the parameter vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$. Because of the independence assumption, the CE problem (5.63) separates into n subproblems of the form above, and all the components of the optimal CE reference parameter $\mathbf{v}^* = (\mathbf{v}_1^*, \ldots, \mathbf{v}_n^*)$, which is now a vector of vectors, follow from (5.71). Note that in this case the optimal VM and CE reference parameters are usually not equal, since we are not optimizing the CE over all densities. See, however, Proposition 4.2 in Rubinstein and Kroese [35] for an important case where they *do* coincide and yield a zero-variance likelihood ratio estimator.

The updating rule (5.71), which involves discrete finite support distributions, and in particular the Bernoulli distribution, will be extensively used for combinatorial optimization problems later on in the book.

■ **EXAMPLE 5.15 Example 5.1 (Continued)**

Consider the bridge network in Figure 5.1, and let

$$S(\mathbf{X}) = \min\{X_1 + X_4,\ X_1 + X_3 + X_5,\ X_2 + X_3 + X_4,\ X_2 + X_5\}\,.$$

We now want to estimate the probability that the shortest path from node A to node B has a length of at least γ; that is, with $H(\mathbf{x}) = I_{\{S(\mathbf{x}) \geqslant \gamma\}}$, we want to estimate

$$\ell = \mathbb{E}[H(\mathbf{X})] = \mathbb{P}_{\mathbf{u}}(S(\mathbf{X}) \geqslant \gamma) = \mathbb{E}_{\mathbf{u}}[I_{\{S(\mathbf{X}) \geqslant \gamma\}}]\,.$$

We assume that the components $\{X_i\}$ are independent, that $X_i \sim \mathsf{Exp}(u_i^{-1})$, $i = 1, \ldots, 5$, and that γ is chosen such that $\ell \geq 10^{-2}$. Thus here the CE updating formula (5.68) and its particular case (5.69) (with $\mathbf{w} = \mathbf{u}$) apply. We will show that this yields substantial variance reduction. The likelihood ratio in this case is

$$\begin{aligned} W(\mathbf{x}; \mathbf{u}, \mathbf{v}) = \frac{f(\mathbf{x}; \mathbf{u})}{f(\mathbf{x}; \mathbf{v})} &= \frac{\prod_{i=1}^5 \frac{1}{u_i} \mathrm{e}^{-x_i/u_i}}{\prod_{i=1}^5 \frac{1}{v_i} \mathrm{e}^{-x_i/v_i}} \\ &= \exp\left(-\sum_{i=1}^5 x_i \left(\frac{1}{u_i} - \frac{1}{v_i}\right)\right) \prod_{i=1}^5 \frac{v_i}{u_i}\,. \end{aligned} \tag{5.72}$$

As a concrete example, let the *nominal* parameter vector $\mathbf{u}$ be equal to $(1, 1, 0.3,\ 0.2, 0.1)$ and let $\gamma = 1.5$. We will see that this probability ℓ is approximately 0.06.

Note that the typical length of a path from A to B is smaller than $\gamma = 1.5$; hence, using importance sampling instead of CMC should be beneficial. The idea is to estimate the optimal parameter vector $\mathbf{v}^*$ *without* using likelihood ratios, that is, using (5.69), since likelihood ratios, as in (5.68) (with quite arbitrary $\mathbf{w}$, say by guessing an initial trial vector $\mathbf{w}$), would typically make the estimator of $\mathbf{v}^*$ unstable, especially for high-dimensional problems.

Denote by $\widehat{\mathbf{v}}_1$ the CE estimator of $\mathbf{v}^*$ obtained from (5.69). We can iterate (repeat) this procedure, say for T iterations, using (5.68), and starting with $\mathbf{w} = \widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, \ldots$. Once the final reference vector $\widehat{\mathbf{v}}_T$ is obtained, we then

estimate ℓ via a *larger* sample from $f(\mathbf{x}; \widehat{\mathbf{v}}_T)$, say of size N_1, using the SLR estimator (5.58). Note, however, that for high-dimensional problems, iterating in this way could lead to an unstable final estimator $\widehat{\mathbf{v}}_T$. In short, a single iteration with (5.69) might often be the best alternative.

Table 5.1 presents the performance of the estimator (5.58), starting from $\mathbf{w} = \mathbf{u} = (1, 1, 0.3, 0.2, 0.1)$ and then iterating (5.68) three times. Note again that in the first iteration we generate a sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ from $f(\mathbf{x}; \mathbf{u})$ and then apply (5.69) to obtain an estimate $\widehat{\mathbf{v}} = (\widehat{v}_1, \ldots, \widehat{v}_5)$ of the CE optimal reference parameter vector $\mathbf{v}^*$. The sample sizes for updating $\widehat{\mathbf{v}}$ and calculating the estimator $\widehat{\ell}$ were $N = 10^3$ and $N_1 = 10^5$, respectively. In the table RE denotes the estimated relative error.

Table 5.1: Iterating the five-dimensional vector $\widehat{\mathbf{v}}$.

Iteration	$\widehat{\mathbf{v}}$					$\widehat{\ell}$	RE
0	1	1	0.3	0.2	0.1	0.0643	0.0121
1	2.4450	2.3274	0.2462	0.2113	0.1030	0.0631	0.0082
2	2.3850	2.3894	0.3136	0.2349	0.1034	0.0644	0.0079
3	2.3559	2.3902	0.3472	0.2322	0.1047	0.0646	0.0080

Note that $\widehat{\mathbf{v}}$ already converged after the first iteration, so using likelihood ratios in iterations 2 and 3 did not add anything to the quality of $\widehat{\mathbf{v}}$. It also follows from the results of Table 5.1 that CE outperforms CMC (compare the relative errors 0.008 and 0.0121 for CE and CMC, respectively). To obtain a similar relative error of 0.008 with CMC would require a sample size of approximately $2.5 \cdot 10^5$ instead of 10^5; we thus obtained a reduction by a factor of 2.5 when using the CE estimation procedure. As we will see in Chapter 8 for smaller probabilities, a variance reduction of several orders of magnitude can be achieved.

5.8 SEQUENTIAL IMPORTANCE SAMPLING

Sequential importance sampling (SIS), also called *dynamic importance sampling*, is simply importance sampling carried out in a sequential manner. To explain the SIS procedure, consider the expected performance ℓ in (5.38) and its likelihood ratio estimator $\widehat{\ell}$ in (5.40), with $f(\mathbf{x})$ the "target" and $g(\mathbf{x})$ the importance sampling, or proposal, pdf. Suppose that (1) $\mathbf{X}$ is decomposable, that is, it can be written as a vector $\mathbf{X} = (X_1, \ldots, X_n)$, where each of the X_i may be multidimensional, and (2) it is easy to sample from $g(\mathbf{x})$ sequentially. Specifically, suppose that $g(\mathbf{x})$ is of the form

$$g(\mathbf{x}) = g_1(x_1)\, g_2(x_2 \,|\, x_1) \cdots g_n(x_n \,|\, x_1, \ldots, x_{n-1})\,, \tag{5.73}$$

where it is easy to generate X_1 from density $g_1(x_1)$, and conditional on $X_1 = x_1$, the second component from density $g_2(x_2 \,|\, x_1)$, and so on, until we obtain a single random vector $\mathbf{X}$ from $g(\mathbf{x})$. Repeating this independently N times, each time sampling from $g(\mathbf{x})$, we obtain a random sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ from $g(\mathbf{x})$ and an

estimator of ℓ according to (5.40). To further simplify the notation, we abbreviate $(x_1, \ldots, x_t)$ to $\mathbf{x}_{1:t}$ for all t. In particular, $\mathbf{x}_{1:n} = \mathbf{x}$. Typically, t can be viewed as a (discrete) time parameter and $\mathbf{x}_{1:t}$ as a path or trajectory. By the product rule of probability (1.4), the target pdf $f(\mathbf{x})$ can also be written sequentially, that is,

$$f(\mathbf{x}) = f(x_1)\, f(x_2 \,|\, x_1) \cdots f(x_n \,|\, \mathbf{x}_{1:n-1}). \tag{5.74}$$

From (5.73) and (5.74) it follows that we can write the likelihood ratio in product form as

$$W(\mathbf{x}) = \frac{f(x_1)\, f(x_2 \,|\, x_1) \cdots f(x_n \,|\, \mathbf{x}_{1:n-1})}{g_1(x_1)\, g_2(x_2 \,|\, x_1) \cdots g_n(x_n \,|\, \mathbf{x}_{1:n-1})} \tag{5.75}$$

or, if $W_t(\mathbf{x}_{1:t})$ denotes the likelihood ratio up to time t, recursively as

$$W_t(\mathbf{x}_{1:t}) = u_t\, W_{t-1}(\mathbf{x}_{1:t-1}), \quad t = 1, \ldots, n\ , \tag{5.76}$$

with initial weight $W_0(\mathbf{x}_{1:0}) = 1$ and *incremental weights* $u_1 = f(x_1)/g_1(x_1)$ and

$$u_t = \frac{f(x_t \,|\, \mathbf{x}_{1:t-1})}{g_t(x_t \,|\, \mathbf{x}_{1:t-1})} = \frac{f(\mathbf{x}_{1:t})}{f(\mathbf{x}_{1:t-1})\, g_t(x_t \,|\, \mathbf{x}_{1:t-1})}\ , \quad t = 2, \ldots, n\ . \tag{5.77}$$

In order to update the likelihood ratio recursively, as in (5.77), we need to known the marginal pdfs $f(\mathbf{x}_{1:t})$. This may not be easy when f does not have a Markov structure, as it requires integrating $f(\mathbf{x})$ over all $x_{t+1}, \ldots, x_n$. Instead, we can introduce a sequence of *auxiliary* pdfs $f_1, f_2, \ldots, f_n$ that are easily evaluated and such that each $f_t(\mathbf{x}_{1:t})$ is a good approximation to $f(\mathbf{x}_{1:t})$. The terminating pdf f_n must be equal to the original f. Since

$$f(\mathbf{x}) = \frac{f_1(x_1)}{1} \frac{f_2(\mathbf{x}_{1:2})}{f_1(x_1)} \cdots \frac{f_n(\mathbf{x}_{1:n})}{f_{n-1}(\mathbf{x}_{1:n-1})}\ , \tag{5.78}$$

we have as a generalization of (5.77) the incremental updating weight

$$u_t = \frac{f_t(\mathbf{x}_{1:t})}{f_{t-1}(\mathbf{x}_{1:t-1})\, g_t(x_t \,|\, \mathbf{x}_{1:t-1})} \tag{5.79}$$

for $t = 1, \ldots, n$, where we put $f_0(\mathbf{x}_{1:0}) = 1$. Summarizing, the SIS method can be written as follows:

Algorithm 5.8.1: SIS Method

input : Sample size N, pdfs $\{f_t\}$ and $\{g_t\}$, performance function H.
output: Estimator of $\ell = \mathbb{E}\,[H\,(\mathbf{X})] = \mathbb{E}[H(\mathbf{X}_{1:n})]$.

for $k = 1$ **to** N **do**
  $X_1 \sim g_1(x_1)$
  $W_1 \leftarrow \frac{f_1(X_1)}{g_1(X_1)}$
  for $t = 2$ **to** n **do**
    $X_t \sim g_t(x_t \,|\, \mathbf{X}_{1:t-1})$ `// simulate next component`
    $W_t \leftarrow W_{t-1} \frac{f_t(\mathbf{X}_{1:t})}{f_{t-1}(\mathbf{X}_{1:t-1})\, g_t(X_t \,|\, \mathbf{X}_{1:t-1})}$
  $W^{(k)} \leftarrow W_n$
  $\mathbf{X}^{(k)} \leftarrow \mathbf{X}_{1:n}$
return $N^{-1} \sum_{k=1}^{N} H(\mathbf{X}^{(k)})\, W_n^{(k)}$

Remark 5.8.1 Note that the incremental weights u_t only need to be defined *up to a constant*, say c_t, for each t. In this case the likelihood ratio $W(\mathbf{x})$ is known up to a constant as well, say $W(\mathbf{x}) = C\, w(\mathbf{x})$, where $1/C = \mathbb{E}_g[w(\mathbf{X})]$ can be estimated via the corresponding sample mean. In other words, when the normalization constant is unknown, we can still estimate ℓ using the weighted sample estimator (5.42) rather than the likelihood ratio estimator (5.40).

■ **EXAMPLE 5.16 Random Walk on the Integers**

Consider the random walk on the integers of Example 1.10 (on Page 20), with probabilities p and q for jumping up or down, respectively. Suppose that $p < q$, so that the walk has a drift toward $-\infty$. Our goal is to estimate the rare-event probability ℓ of reaching state K before state 0, starting from state $0 < k \ll K$, where K is a large number. As an intermediate step, consider first the probability of reaching K in exactly n steps, that is, $\mathbb{P}(X_n = K) = \mathbb{E}[I_{A_n}]$, where $A_n = \{X_n = K\}$. We have

$$f(\mathbf{x}_{1:n}) = f(x_1 \,|\, k)\, f(x_2 \,|\, x_1)\, f(x_3 \,|\, x_2) \ldots f(x_n \,|\, x_{n-1}) \;,$$

where the conditional probabilities are either p (for upward jumps) or q (for downward jumps). If we simulate the random walk with *different* upward and downward probabilities, $\tilde{p}$ and $\tilde{q}$, then the importance sampling pdf $g(\mathbf{x}_{1:n})$ has the same form as $f(\mathbf{x}_{1:n})$ above. Thus the importance weight after Step t is updated via the incremental weight

$$u_t = \frac{f(x_t \,|\, x_{t-1})}{g(x_t \,|\, x_{t-1})} = \begin{cases} p/\tilde{p} & \text{if } x_t = x_{t-1} + 1 \;, \\ q/\tilde{q} & \text{if } x_t = x_{t-1} - 1 \;. \end{cases}$$

The probability $\mathbb{P}(A_n)$ can now be estimated via importance sampling as

$$\frac{1}{N} \sum_{i=1}^{N} W_{i,n}\, I_{\{X_{i,n}=K\}} \;, \tag{5.80}$$

where the paths $\mathbf{X}_{i,1:n}, i = 1, \ldots, N$, are generated via g, rather than f and $W_{i,n}$ is the likelihood ratio of the i-th such path. Returning to the estimation of ℓ, let τ be the first time that either 0 or K is reached. Writing $I_{\{X_t=K\}} = H(\mathbf{X}_{1:t})$, we have

$$\begin{aligned} \ell &= \mathbb{E}_f[I_{\{X_\tau=K\}}] = \mathbb{E}_f[H(\mathbf{X}_{1:\tau})] = \sum_{n=1}^{\infty} \mathbb{E}[H(\mathbf{X}_{1:n})\, I_{\{\tau=n\}}] \\ &= \sum_{n=1}^{\infty} \sum_{\mathbf{x}} H(\mathbf{x}_{1:n})\, I_{\{\tau=n\}}\, f(\mathbf{x}_{1:n}) \\ &= \sum_{n=1}^{\infty} \sum_{\mathbf{x}} \underbrace{\frac{f(\mathbf{x}_{1:n})}{g(\mathbf{x}_{1:n})} I_{\{x_n=K\}}}_{\tilde{H}(\mathbf{x}_{1:n})}\, I_{\{\tau=n\}}\, g(\mathbf{x}_{1:n}) \\ &= \mathbb{E}_g[\tilde{H}(\mathbf{X}_{1:\tau})] = \mathbb{E}_g[W_\tau\, I_{\{X_\tau=K\}}] \;, \end{aligned}$$

with W_τ the likelihood ratio of $\mathbf{X}_{1:\tau}$, which can be updated at each time t by multiplying with either $p/\tilde{p}$ or $q/\tilde{q}$ for upward and downward steps, respectively. Note that $I_{\{\tau=n\}}$ is indeed a function of $\mathbf{x}_n = (x_1, \ldots, x_n)$. This leads to the same estimator as (5.80) with the deterministic n replaced by the stochastic τ. It can be shown (e.g., see [5]) that choosing $\tilde{p} = q$ and $\tilde{q} = p$, that is, *interchanging* the probabilities, gives an efficient estimator for ℓ.

EXAMPLE 5.17 Counting Self-Avoiding Walks

The self-avoiding random walk, or simply *self-avoiding walk*, is a basic mathematical model for polymer chains. For simplicity, we will deal only with the two-dimensional case. Each self-avoiding walk is represented by a path $\mathbf{x} = (x_1, x_2, \ldots, x_{n-1}, x_n)$, where x_i represents the two-dimensional position of the i-th molecule of the polymer chain. The distance between adjacent molecules is fixed at 1, and the main requirement is that the chain does not self-intersect. We assume that the walk starts at the origin. An example of a self-avoiding walk walk of length 130 is given in Figure 5.3.

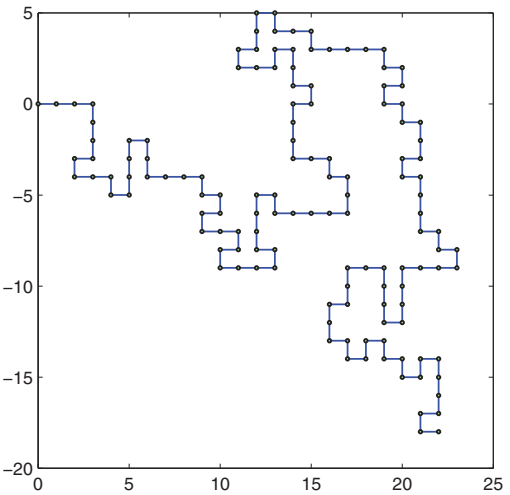

Figure 5.3: A self-avoiding random walk of length $n = 130$.

One of the main questions regarding the self-avoiding walk model is: how many self-avoiding walks are there of length n? Let $\mathscr{X}^*$ be the set of self-avoiding walks of length n. The exact number of self-avoiding walks up to $n = 72$ can be found in `http://www.ms.unimelb.edu.au/~iwan/saw/series/sqsaw.ser`. The first 20 entries are as follows:

n	$\lvert\mathscr{X}^*\rvert$	n	$\lvert\mathscr{X}^*\rvert$	n	$\lvert\mathscr{X}^*\rvert$	n	$\lvert\mathscr{X}^*\rvert$
0	1	5	284	10	44100	15	6416596
1	4	6	780	11	120292	16	17245332
2	12	7	2172	12	324932	17	46466676
3	36	8	5916	13	881500	18	124658732
4	100	9	16268	14	2374444	19	335116620

We wish to estimate $|\mathscr{X}^*|$ via Monte Carlo. The crude Monte Carlo approach is to use acceptance–rejection in the follow way:

1. Generate a random sample $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(N)}$ uniformly distributed over the set $\mathscr{X}$ of all random walks of length n. This set has $|\mathscr{X}| = 4^n$ elements. Generating the samples from $\mathscr{X}$ is easy.

2. Estimate the desired number $|\mathscr{X}^*|$ as

$$\widehat{|\mathscr{X}^*|} = |\mathscr{X}| \, \frac{1}{N} \sum_{k=1}^{N} I_{\{\mathbf{X}^{(k)} \in \mathscr{X}^*\}} \,, \tag{5.81}$$

where $I_{\{\mathbf{X}^{(k)} \in \mathscr{X}^*\}}$ denotes the indicator of the event $\{\mathbf{X}^{(k)} \in \mathscr{X}^*\}$. Note that according to (5.81) we accept the generated point $\mathbf{X}^{(k)}$ if $\mathbf{X}^{(k)} \in \mathscr{X}^*$ and reject it otherwise.

Unfortunately, for large n the event $\{\mathbf{X}^{(k)} \in \mathscr{X}^*\}$ is very rare. Acceptance–rejection is meaningless if there are no acceptable samples. Instead, we could opt to use importance sampling. In particular, let g be an importance sampling pdf defined on some set $\mathscr{X}$ and let $\mathscr{X}^* \subset \mathscr{X}$; then $|\mathscr{X}^*|$ can be written as

$$|\mathscr{X}^*| = \sum_{\mathbf{x} \in \mathscr{X}} I_{\{\mathbf{x} \in \mathscr{X}^*\}} \frac{g(\mathbf{x})}{g(\mathbf{x})} = \mathbb{E}_g \left[\frac{I_{\{\mathbf{x} \in \mathscr{X}^*\}}}{g(\mathbf{X})} \right]. \tag{5.82}$$

To estimate $|\mathscr{X}^*|$ via Monte Carlo, we draw a random sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ from g and take the estimator

$$\widehat{|\mathscr{X}^*|} = \frac{1}{N} \sum_{k=1}^{N} I_{\{\mathbf{X}^{(k)} \in \mathscr{X}^*\}} \frac{1}{g(\mathbf{X}^{(k)})} \,. \tag{5.83}$$

The best choice for g is $g^*(\mathbf{x}) = 1/|\mathscr{X}^*|, \mathbf{x} \in \mathscr{X}^*$; in words, $g^*(\mathbf{x})$ is the uniform pdf over the discrete set $\mathscr{X}^*$. Under g^* the estimator has zero variance, so that only *one sample is required*. Clearly, such g^* is infeasible. Fortunately, the SAW counting problem presents a natural *sequential* importance sampling density g. This pdf is defined by the following *one-step-look-ahead* procedure:

Algorithm 5.8.2: One-Step–Look-Ahead

input : Length of path n.
output: Self-avoiding walk of length n, or $\emptyset$ (no such path found).

Let $X_0 \leftarrow (0,0)$ and $t \leftarrow 1$.
for $t = 1$ **to** n **do**
  Let d_t be the number of neighbors of X_{t-1} that have not yet been visited.
  if $d_t > 0$ **then**
    Choose X_t with probability $1/d_t$ from its neighbors.
  else
    return $\emptyset$ `// no SAW of length` n `found`
return $X_1, \ldots, X_n$

Note that the procedure generates either a self-avoiding walk $\mathbf{x}$ of length n or $\emptyset$. Let $g(\mathbf{x})$ be the corresponding discrete pdf. Then, for any self-avoiding walk $\mathbf{x}$ of length n, we have by the product rule (1.4) that

$$g(\mathbf{x}) = \frac{1}{d_1}\frac{1}{d_2}\cdots\frac{1}{d_n} = \frac{1}{w(\mathbf{x})} ,$$

where

$$w(\mathbf{x}) = d_1 \cdots d_n . \tag{5.84}$$

The self-avoiding walk counting algorithm below now follows directly from (5.83).

Algorithm 5.8.3: Counting Self-Avoiding Walks

input : Length of path n.
output: Estimator $\widehat{|\mathscr{X}^*|}$ of the number of self-avoiding walks of length n.

Generate independently N paths $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(N)}$ via the one-step-look-ahead procedure.
For each self-avoiding walk $\mathbf{X}^{(k)}$, compute $w(\mathbf{X}^{(k)})$ as in (5.84). If $\emptyset$ is returned, set $w(\mathbf{X}^{(k)}) \leftarrow 0$.
return $\frac{1}{N}\sum_{k=1}^{N} w(\mathbf{X}^{(k)})$

The efficiency of the simple one-step-look-ahead method deteriorates rapidly as n becomes large. It becomes impractical to simulate walks of length more than 200. This is due to the fact that, if at any one step t the point x_{t-1} does not have unoccupied neighbors ($d_t = 0$), then the "weight" $w(\mathbf{x})$ is zero and contributes nothing to the final estimate of $|\mathscr{X}^*|$. This problem can occur early in the simulation, rendering any subsequent sequential buildup useless. Better-performing algorithms do not restart from scratch but reuse successful partial walks to build new walks. These methods usually split the self-avoiding partial walks into a number of copies and continue them as if they were independently built up from scratch. We refer to [27] for a discussion of these more advanced algorithms. We will revisit this example in Chapter 9, where the splitting method is used to estimate the number of SAWs.

5.9 SEQUENTIAL IMPORTANCE RESAMPLING

A common problem with sequential importance sampling (Algorithm 5.8.1) is that the distribution of the importance weight W_t becomes very skewed as t increases, resulting in a high probability of a very small weight and a small probability of a very large weight; see also Remark 5.7.1. As a consequence, most of the N samples will not contribute significantly to the final estimator $\widehat{\ell}$ in (5.41).

One way to rectify this issue is to *resample* high-weight samples. To explain the resampling procedure, we first give the parallel version of Algorithm 5.8.1. Instead of simulating all n components of $\mathbf{X} = (X_1, \ldots, X_n)$ and repeating this process N times, we can simulate N copies of the first component X_1, then simulate N copies of the second components $\mathbf{X}_{1:2} = (X_1, X_2)$, and so on. To further enhance the generality of Algorithm 5.8.1, we assume that each auxiliary pdf f_t is known *up to a normalization constant* c_t; see also Remark 5.8.1. In particular, we assume that the product $c_t\, f_t(\mathbf{x}_{1:t})$ can be explicitly evaluated, whereas c_t and $f_t(\mathbf{x}_{1:t})$ may not. If f_n is known, we can set $c_n = 1$ and use the ordinary likelihood ratio estimator (5.41) to estimate ℓ. If c_n is unknown, we must use the weighted estimator (5.42). That is, in Line 8 below return

$$\left(\sum_{k=1}^{N} W_n^{(k)}\right)^{-1} \sum_{k=1}^{N} H(\mathbf{X}_{1:n}^{(k)})\, W_n^{(k)} \,.$$

Algorithm 5.9.1: Parallel SIS

input : Sample size N, unnormalized pdfs $\{c_t f_t\}$ and pdfs $\{g_t\}$, performance function H.

output: Estimator of $\ell = \mathbb{E}\,[H\,(\mathbf{X})] = \mathbb{E}[H(\mathbf{X}_{1:n})]$.

1 **for** $k = 1$ **to** N **do**

2 $\quad X_1^{(k)} \sim g_1(x_1)$ `// simulate first component`

3 $\quad W_1^{(k)} \leftarrow \frac{c_1 f_1(X_1^{(k)})}{g_1(X_1^{(k)})}$

4 **for** $t = 2$ **to** n **do**

5 $\quad$ **for** $k = 1$ **to** N **do**

6 $\quad\quad X_t^{(k)} \sim g_t(x_t \,|\, \mathbf{X}_{1:t-1}^{(k)})$ `// simulate next component`

7 $\quad\quad W_t^{(k)} \leftarrow W_{t-1}^{(k)} \frac{c_t\, f_t(\mathbf{X}_{1:t}^{(k)})}{c_{t-1}\, f_{t-1}(\mathbf{X}_{1:t-1}^{(k)})\, g_t(X_t^{(k)} \,|\, \mathbf{X}_{1:t-1}^{(k)})}$

8 **return** $N^{-1} \sum_{k=1}^{N} H(\mathbf{X}_{1:n}^{(k)})\, W_n^{(k)}$

Note that at any stage $t = 1, \ldots, n$ the "weighted particles" $\{(\mathbf{X}_{1:t}^{(k)}, W_t^{(k)})\}_{k=1}^N$ can provide the unbiased estimator $\sum_{k=1}^N H_t(\mathbf{X}_{1:t}^{(k)}) W_t^{(k)}$ of $\mathbb{E}_{f_t}[H_t(\mathbf{X}_{1:t})]$ for any function H_t of $\mathbf{X}_{1:t}$. Let $\{\mathbf{Y}_t^{(k)}\}_{k=1}^N$ be a sample of size N chosen with replacement from $\{\mathbf{X}_{1:t}^{(k)}\}_{k=1}^N$ with probabilities proportional to $\{W_t^{(k)}\}_{k=1}^N$, and let $\overline{W}_t = N^{-1} \sum_{k=1}^N W_t^{(k)}$. Then,

$$\mathbb{E}\left[\sum_{k=1}^{N} H_t(\mathbf{Y}_t^{(k)})\,\overline{W}_t\right] = \mathbb{E}\left[\overline{W}_t \sum_{k=1}^{N} \mathbb{E}\left[H_t(\mathbf{Y}_t^{(k)}) \,|\, \mathbf{X}_{1:t}^{(1)}, \ldots, \mathbf{X}_{1:t}^{(N)}, W_t^{(1)}, \ldots, W_t^{(N)}\right]\right]$$

$$= \mathbb{E}\left[\overline{W}_t \sum_{k=1}^{N}\sum_{j=1}^{N} \frac{H_t(\mathbf{X}_{1:t}^{(j)})\,W_t^{(j)}}{N\overline{W}_t}\right] = \mathbb{E}\left[\sum_{j=1}^{N} H_t(\mathbf{X}_{1:t}^{(j)})\,W_t^{(j)}\right]. \tag{5.85}$$

This suggests that we replace the variables $\{(\mathbf{X}_{1:t}^{(k)}, W_t^{(k)})\}_{k=1}^N$ by $\{(\mathbf{Y}_t^{(k)}, \overline{W}_t)\}_{k=1}^N$ and continue the sequential importance sampling algorithm. This type of resampling is called *bootstrap resampling*. When the importance weights are all identical, this corresponds to *simple random sampling with replacement*.

Adding such a resampling step to Algorithm 5.9.1 for every t results in *sequential importance resampling* (SIR) Algorithm 5.9.2. It can be shown that the weighted estimator returned by the algorithm is asymptotically unbiased and asymptotically normal [7].

Note that the addition of a resampling step can result in a worse estimator. For example, if H is a positive function, then the optimal importance sampling density is $g \propto Hf$ and the resulting importance sampling estimator has zero variance. If a resampling step is added, then the resulting estimator can have nonzero variance.

Algorithm 5.9.2: SIR Algorithm with Bootstrap Resampling

input : Sample size N, unnormalized pdfs $\{c_t f_t\}$ and pdfs $\{g_t\}$, performance function H.

output: Estimator of $\ell = \mathbb{E}[H(\mathbf{X})] = \mathbb{E}[H(\mathbf{X}_{1:n})]$.

for $k = 1$ **to** N **do**
  $X_1^{(k)} \sim g_1(x_1)$ `// simulate first component`
  $W_1^{(k)} \leftarrow \frac{c_1 f_1(X_1^{(k)})}{g_1(X_1^{(k)})}$
for $t = 2$ **to** n **do**
  $\mathbf{Y}_{t-1}^{(1)}, \ldots, \mathbf{Y}_{t-1}^{(N)} \leftarrow$ iid samples from $\mathbf{X}_{1:t-1}^{(1)}, \ldots, \mathbf{X}_{1:t-1}^{(N)}$, with probabilities proportional to $W_{t-1}^{(1)}, \ldots, W_{t-1}^{(N)}$ `// resample`
  $\overline{W}_{t-1} \leftarrow N^{-1}\sum_{k=1}^N W_{t-1}^{(k)}$ `// compute average weight`
  for $k = 1$ **to** N **do**
    $X_t^{(k)} \sim g_t(x_t \,|\, \mathbf{Y}_{t-1}^{(k)})$ `// simulate next component`
    $\mathbf{X}_{1:t}^{(k)} \leftarrow (\mathbf{Y}_{t-1}^{(k)}, X_t^{(k)})$
    $W_t^{(k)} \leftarrow \overline{W}_{t-1} \frac{c_t f_t(\mathbf{X}_{1:t}^{(k)})}{c_{t-1} f_{t-1}(\mathbf{Y}_{t-1}^{(k)})\, g_t(X_t^{(k)} \,|\, \mathbf{Y}_{t-1}^{(k)})}$ `// update weight`
return $(\sum_{k=1}^N W_n^{(k)})^{-1} \sum_{k=1}^N H(\mathbf{X}_{1:n}^{(k)}) W_n^{(k)}$

There are various other ways in which the resampling step can be carried out. For example, in the *enrichment* method of [42] resampling is performed by making r_t copies of $\mathbf{X}_{1:t}^{(k)}$ for some integer $r_t > 0$. A natural generalization of enrichment is to split every particle $\mathbf{X}_{1:t}^{(k)}$ into a random number $R_t^{(k)}$ of copies, e.g., with some fixed expectation r_t that does not have to be an integer. The natural choice is