

CHAPTER 19

Simulation Modeling

資料來源:

Operations Research
An Introduction 9th ed
Hamdy A. Taha.

19.1 MONTE CARLO SIMULATION

A forerunner to present-day simulation is the Monte Carlo experiment, a modeling scheme that estimates stochastic or deterministic parameters based on random sampling. Examples of Monte Carlo applications include evaluation of multiple integrals, estimation of the constant π (≈ 3.14159), and matrix inversion.

This section uses an example to demonstrate the Monte Carlo technique. The objective of the example is to emphasize the statistical nature of simulation.

Example 19.1-1

We will use Monte Carlo sampling to estimate the area of the following circle:

$$(x - 1)^2 + (y - 2)^2 = 25$$

The radius of the circle is $r = 5$ cm, and its center is $(x, y) = (1, 2)$.

The procedure for estimating the area requires enclosing the circle tightly in a square whose side equals the diameter of the circle, as shown in Figure 19.1. The corner points are determined from the geometry of the square.

The estimation of the area of the circle is based on a sampling experiment that gives equal chance to selecting any point in the square. If m out of n sampled points fall within the circle, then

$$\left(\begin{array}{c} \text{Approximate} \\ \text{area of the circle} \end{array} \right) = \frac{m}{n} \left(\begin{array}{c} \text{Area of} \\ \text{the square} \end{array} \right) = \frac{m}{n} (10 \times 10)$$

To ensure that all the points in the square are equally probable, the coordinates x and y of a point in the square are represented by the following *uniform* distributions:

$$f_1(x) = \frac{1}{10}, -4 \leq x \leq 6$$

$$f_2(y) = \frac{1}{10}, -3 \leq y \leq 7$$

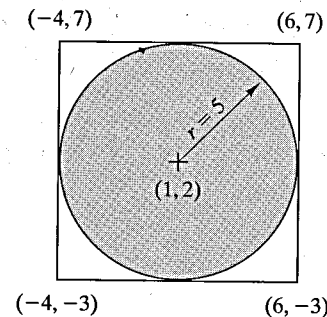


FIGURE 19.1

Monte Carlo estimation of the area of a circle

The determination of a sample (x, y) is based on the use of independent 0-1 random numbers. Table 19.1 lists a sample of such numbers which we will use in the examples in this chapter. For the purpose of general simulation, special arithmetic operations are used to generate (pseudo) 0-1 random numbers, as will be shown in Section 19.4.

A pair of 0-1 random numbers, R_1 and R_2 , can be used to generate a random point (x, y) in the square by using the following formulas:

$$x = -4 + [6 - (-4)]R_1 = -4 + 10R_1$$

$$y = -3 + [7 - (-3)]R_2 = -3 + 10R_2$$

To demonstrate the application of the procedure, consider $R_1 = .0589$ and $R_2 = .6733$.

$$x = -4 + 10R_1 = -4 + 10 \times .0589 = -3.411$$

$$y = -3 + 10R_2 = -3 + 10 \times .6733 = 3.733$$

This point falls inside the circle because

$$(-3.411 - 1)^2 + (3.733 - 2)^2 = 22.46 < 25$$

Remarks. The accuracy of the area estimate can be enhanced by using procedures from ordinary statistical experiments:

1. Increase the sample size, n .
2. Use replications, N .

TABLE 19.1 A Short List of 0-1 Random Numbers

.0589	.3529	.5869	.3455	.7900	.6307
.6733	.3646	.1281	.4871	.7698	.2346
.4799	.7676	.2867	.8111	.2871	.4220
.9486	.8931	.8216	.8912	.9534	.6991
.6139	.3919	.8261	.4291	.1394	.9745
.5933	.7876	.3866	.2302	.9025	.3428
.9341	.5199	.7125	.5954	.1605	.6037
.1782	.6358	.2108	.5423	.3567	.2569
.3473	.7472	.3575	.4208	.3070	.0546
.5644	.8954	.2926	.6975	.5513	.0305

The discussion in Example 19.1-1 poses two questions regarding the simulation experiment:

1. How large should the sample size be?
2. How many replications are needed?

There are some formulas in statistical theory for determining n and N , and they depend on the nature of the simulation experiment as well as the desired confidence level. However, as in any statistical experiment, the golden rule is that higher values of n and N mean more accurate simulation results. In the end, the sample size will depend on the cost associated with conducting the simulation experiment. Generally speaking, however, a selected sample size is considered "adequate" if it produces a relatively "small" standard deviation.

It is necessary to express the results as a confidence interval to account for the random variation in the output of the experiment. Letting \bar{A} and s be the mean and variance of N replications, then, given a confidence level α , the confidence interval for the true area A is

$$\bar{A} - \frac{s}{\sqrt{N}} t_{\frac{\alpha}{2}, N-1} \leq A \leq \bar{A} + \frac{s}{\sqrt{N}} t_{\frac{\alpha}{2}, N-1}$$

The parameter $t_{\frac{\alpha}{2}, N-1}$ is determined from the t -distribution tables given a confidence level α and $N - 1$ degrees of freedom (see the t -table in Appendix A or use *excelStatTables.xls*). Note that N equals the number of replications, which is distinct from the sample size n .

Excel Moment

The computations associated with each sample in Example 19.1-1 are voluminous. Excel template *excelCircle.xls* (with VBA macros) is used to test the effect of sample size and number of replications on the accuracy of the area estimate. The input data include the circle radius, r ; and its center (cx, cy) ; sample size, n ; number of replications, N ; and the confidence level α . The entry *Steps* in cell D4 allows executing several samples in the same run. For example, if $n = 30,000$ and *Steps* = 3, the template will automatically produce output for $n = 30,000, 60,000$, and $90,000$. New estimates are realized each time the command button **Press to Execute Monte Carlo** is clicked because Excel refreshes the seed of the random number generator.

Figure 19.2 summarizes the results for 5 replications and sample sizes of 30,000, 60,000, and 90,000. The exact area is 78.54 cm^2 , and the Monte Carlo results show that the mean estimated areas for the three sample sizes are slightly different.

Figure 16.2 gives the 95% confidence intervals for each n . For example, the confidence interval $78.452 \leq A \leq 78.68$ corresponds to $n = 90,000$, with $N = 5$, $\bar{A} = 78.566 \text{ cm}^2$, and $s = .092 \text{ cm}$, and $t_{0.025, 4} = 2.776$. In general, to realize reasonable accuracy in the estimation of the confidence interval, the value of N should be at least 5.

PROBLEM SET 19.1A

1. In Example 19.1-1, estimate the area of the circle using the first two columns of the 0-1 random numbers in Table 19.1. (For convenience, go down each column, selecting R_1 first and then R_2 .) How does this estimate compare with the ones given in Figure 19.2?

	B	C	D	E
1	Monte Carlo Estimation of the Area of a Circle			
2	Input data			
3	Nbr. Replications, N =	5	$\alpha =$	0.025
4	Sample size, n =	30,000	Steps =	3
5	Radius, r =	5		
6	Center, cx =	1		
7	Center, cy =	2		
8	Output results			
9	Exact area =	78.540		
10	Press to Execute Monte Carlo			
11	Monte Carlo Calculations:			
12		n=30000	n=60000	n=90000
13	Replication 1	78.590	78.543	78.536
14	Replication 2	78.447	78.695	78.731
15	Replication 3	78.747	78.648	78.534
16	Replication 4	78.363	78.500	78.512
17	Replication 5	78.540	78.420	78.517
18				
19	Mean =	78.537	78.561	78.566
20	Std. Deviation =	0.142	0.118	0.092
21				
22	95% lower conf. limit =	78.361	78.415	78.452
23	95% upper conf. limit =	78.714	78.708	78.680

FIGURE 19.2

Excel output of Monte Carlo estimation of the area of a circle (file *excelCircle.xls*)

2. Suppose that the equation of a circle is

$$(x - 4)^2 + (y + 3)^2 = 25$$

- Define the corresponding distributions $f(x)$ and $f(y)$, and then show how a sample point (x, y) is determined using the $(0, 1)$ random pair (R_1, R_2) .
 - Use *excelCircle.xls* to estimate the area and the associated 95% confidence interval, given $n = 100,000$ and $N = 10$.
3. Use Monte Carlo sampling to estimate the area of the lake shown in Figure 19.3. Base the estimate on the first two columns of $(0, 1)$ random numbers in Table 19.1.
4. Consider the game in which two players, Jan and Jim, take turns in tossing a fair coin. If the outcome is heads, Jim gets \$10 from Jan. Otherwise, Jan gets \$10 from Jim.
- How is the game simulated as a Monte Carlo experiment?
 - Run the experiment for 5 replications of 10 tosses each. Use the first five columns of the 0-1 random numbers in Table 19.1, with each column corresponding to one replication.

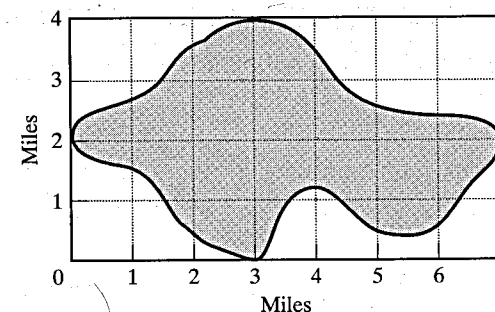


FIGURE 19.3

Lake map for Problem 3, Set 19.1A

- Establish a 95% confidence interval on Jan's winnings.
 - Compare the confidence interval in (c) with Jan's expected theoretical winnings.
5. Consider the following definite integral:

$$\int_0^1 x^2 dx$$

- Develop the Monte Carlo experiment to estimate the value of the integral.
 - Use the first four columns in Table 19.1 to evaluate the integral based on 4 replications of size 5 each. Compute a 95% confidence interval, and compare it with the exact value of the integral.
6. Simulate five wins or losses of the following game of craps: The player rolls two fair dice. If the outcome sum is 7 or 11, the player wins \$10. Otherwise, the player records the resulting sum (called *point*) and keeps on rolling the dice until the outcome sum matches the recorded *point*, in which case the player wins \$10. If a 7 is obtained prior to matching the *point*, the player loses \$10.
- *7. The lead time for receiving an order can be 1 or 2 days, with equal probabilities. The demand *per day* assumes the values 0, 1, and 2 with the respective probabilities of .2, .7, and .1. Use the random numbers in Table 19.1 (starting with column 1) to estimate the joint distribution of the demand and lead time. From the joint distribution, estimate the pdf of demand during lead time. (*Hint*: The demand during lead time assumes discrete values from 0 to 4.)
8. Consider the Buffon needle experiment. A horizontal plane is ruled with parallel lines spaced D cm apart. A needle of length d cm ($d < D$) is dropped randomly on the plane. The objective of the experiment is to determine the probability that either end of the needle touches or crosses one of the lines. Define
- $$h = \text{Perpendicular distance from the needle center to a (parallel) line}$$
- $$\theta = \text{Inclination angle of the needle with a line}$$
- Show that the needle will touch or cross a line only if

$$h \leq \frac{d}{2} \sin \theta, 0 \leq h \leq \frac{D}{2}, 0 \leq \theta \leq \pi$$
 - Design the Monte Carlo experiment, and provide an estimate of the desired probability.
 - Use Excel to obtain 4 replications of size 10 each of the desired probability. Determine a 95% confidence interval for the estimate. Assume $D = 20$ cm and $d = 10$ cm.

- (d) Prove that the theoretical probability is given by the formula

$$p = \frac{2d}{\pi D}$$

- (e) Use the result in (c) together with the formula in (d) to estimate π .
9. Design a Monte Carlo experiment for estimating the value of the constant π .
[Hint: (Area of a circle)/(Area of rectangle tightly enveloping the circle) = $\pi/4$.]

19.2 TYPES OF SIMULATION

The execution of present-day simulation is based on the idea of sampling used with the Monte Carlo method. It differs in that it deals with the study of the behavior of real systems *as a function of time*. Two distinct types of simulation models exist.

1. **Continuous models** deal with systems whose behavior changes *continuously* with time. These models usually use difference-differential equations to describe the interactions among the different elements of the system. A typical example deals with the study of world population dynamics.
2. **Discrete models** deal primarily with the study of waiting lines, with the objective of determining such measures as the average waiting time and length of the queue. These measures change only when a customer enters or leaves the system. The instants at which changes take place occur at specific discrete points in time (arrivals and departure events), giving rise to the name **discrete event simulation**.

This chapter presents the basics of discrete event simulation, including a description of the components of a simulation model, collection of simulation statistics, and the statistical aspect of the simulation experiment. The chapter also emphasizes the role of the computer and simulation languages in the execution of simulation models.

PROBLEM SET 19.2A

1. Categorize the following situations as either discrete or continuous (or a combination of both). In each case, specify the objective of developing the simulation model.
 - *(a) Orders for an item arrive randomly at a warehouse. An order that cannot be filled immediately from available stock must await the arrival of new shipments.
 - (b) World population is affected by the availability of natural resources, food production, environmental conditions, educational level, health care, and capital investments.
 - (c) Goods arrive on pallets at a receiving bay of an automated warehouse. The pallets are loaded on a lower conveyor belt and lifted through an up-elevator to an upper conveyor that moves the pallets to corridors. The corridors are served by cranes that pick up the pallets from the conveyor and place them in storage bins.
2. Explain why you would agree or disagree with the following statement: "Most discrete event simulation models can be viewed in some form or another as queuing systems consisting of *sources* from which customers arrive, *queues* where customers may wait, and *facilities* where customers are served."

19.3 ELEMENTS OF DISCRETE EVENT SIMULATION

The ultimate goal of simulation is to estimate some desirable measures of performance that describe the behavior of the simulated system. For example, in a service facility, the associated measures of performance can include the average waiting time until a customer is served, the average length of the queue, and the average utilization of the service facility. This section shows how the statistics of the simulated system are collected based on the concept of *events*.

19.3.1 Generic Definition of Events

All discrete event simulations describe, directly or indirectly, queuing situations in which customers arrive (for service), wait in a queue (if necessary), and then receive service before leaving the service facility. As such, any discrete event simulation, regardless of the complexity of the system it describes, reduces to dealing with two basic events: arrivals and departures. The following example illustrates the use of the arrival and departure events to describe a system consisting of distinct queues.

Example 19.3-1

Metalco Jobshop receives two types of jobs: regular and rush. All jobs are processed on two consecutive machines with ample buffer areas. Rush jobs always assume nonpreemptive priority over regular jobs.

This situation consists of two tandem queues representing the two machines. At first, one may be inclined to identify the events of the situation as

- A11: A regular job arrives at machine 1.
- A21: A rush job arrives at machine 1.
- D11: A regular job departs machine 1.
- D21: A rush job departs machine 1.
- A12: A regular job arrives at machine 2.
- A22: A rush job arrives at machine 2.
- D12: A regular job departs machine 2.
- D22: A rush job departs machine 2.

In reality, there are only two events: an arrival of a (new) job at the shop and a departure of a (completed) job from a machine. First notice that events $D11$ and $A12$ are actually one and the same. The same applies to $D21$ and $A22$. Next, in discrete simulation we can use one event (arrival or departure) for both types of jobs and simply "tag" the event with an **attribute** that identifies the job type as either regular or rush. (We can think of the attribute in this case as a *personal identification descriptor*, and indeed it is.) Given this reasoning, the events of the model reduce to (1) an arrival A (at the shop) and (2) a departure D (from a machine). The actions associated with the arrival event depends on the type of arriving job (rush or regular) and the availability of a machine. Similarly, the processing of the departure event will depend on the machine and the status of waiting jobs.

Having defined the basic events of a simulation model, we show how the model is executed. Figure 19.4 gives a schematic representation of typical occurrences of events on the simulation timescale. After all the actions associated with a current event have been performed, the simulation advances by "jumping" to the next chronological event. In essence, the execution of the simulation occurs at the instants at which the events occur.

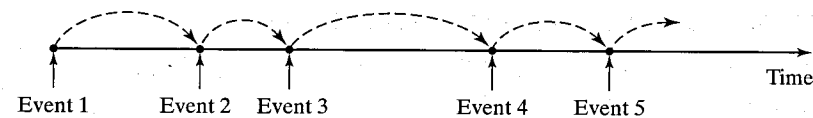


FIGURE 19.4

Example of the occurrence of simulation events on the timescale

How does the simulation determine the occurrence time of the events? The arrival events are separated by the interarrival time (the interval between successive arrivals), and the departure events are a function of the service time in the facility. These times may be deterministic (e.g., a train arriving at a station every 5 minutes) or probabilistic (e.g., the random arrival of customers at a bank). If the time between events is deterministic, the determination of their occurrence times is straightforward. If it is probabilistic, we use a special procedure to sample from the corresponding probability distribution. This point is discussed in the next section.

PROBLEM SET 19.3A

1. Identify the discrete events needed to simulate the following situation: Three types of jobs arrive from different sources. All three types are processed on a single machine, with the highest priority given to jobs from the first source followed by source 2, then source 3.
2. Jobs arrive at a constant rate at a carousel conveyor system. Two service stations are spaced equally around the carousel. If the server is idle when a job arrives at the station, the job is removed from the conveyor for processing. Otherwise, the job continues to rotate on the carousel until a server becomes available. A processed job is stored in an adjacent shipping area. Identify the discrete events needed to simulate this situation.
3. Cars arrive at a two-lane, drive-in bank, where each lane can house a maximum of four cars. If the two lanes are full, arriving cars seek service elsewhere. If at any time one lane is at least two cars longer than the other, the last car in the longer lane will jockey to the last position in the shorter lane. The bank operates the drive-in facility from 8:00 A.M. to 3:00 P.M. each work day. Define the discrete events for the situation.
- *4. The cafeteria at Elmdale Elementary provides a single-tray, fixed-menu lunch to all its pupils. Kids arrive at the dispensing window every 30 seconds. It takes 18 seconds to receive the lunch tray. Map the arrival-departure events on the time scale for the first five pupils.

19.3.2 Sampling from Probability Distributions

Randomness in simulation arises when the interval, t , between successive events is probabilistic. This section presents three methods for generating successive random samples ($t = t_1, t_2, \dots$) from a probability distribution $f(t)$:

1. Inverse method.
2. Convolution method.
3. Acceptance-rejection method.

The inverse method is particularly suited for analytically tractable probability density functions, such as the exponential and the uniform. The remaining two methods deal with more complex cases, such as the normal and the Poisson. All three methods are rooted in the use of independent and identically distributed uniform 0-1 random numbers.

This section will present the first two methods only. Details of the acceptance-rejection method can be found in the bibliography.

Inverse method. Suppose that it is desired to obtain a random sample x from the (continuous or discrete) probability density function $f(x)$. The inverse method first determines a closed-form expression of the cumulative density function $F(x) = P\{y \leq x\}$, where $0 \leq F(x) \leq 1$, for all defined values of y .

It can be proved that the random variable $z = F(x)$ is uniformly distributed in the interval $0 \leq z \leq 1$. Based on this result, a random sample from $f(x)$ is determined using the following steps (F^{-1} is the inverse of F):

Step 1. Generate a 0-1 random number, R .

Step 2. Compute the desired sample $x = F^{-1}(R)$.

Figure 19.5 illustrates the procedures for both a continuous and a discrete random distribution.

Example 19.3-2 (Exponential Distribution)

The exponential probability density function $f(t) = \lambda e^{-\lambda t}$, $t > 0$ represents the interarrival time t at a facility with a mean value of $\frac{1}{\lambda}$. The cumulative density function is

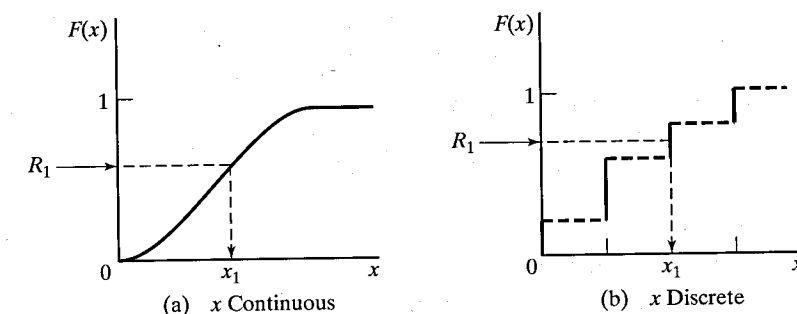
$$F(t) = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t}, t > 0$$

Setting $R = F(t)$, we can solve for t as

$$t = -\left(\frac{1}{\lambda}\right) \ln(1 - R)$$

FIGURE 19.5

Sampling from a probability distribution by the inverse method



For example, for $\lambda = 4$ customers per hour and $R = .9$, the time period until the next arrival occurs is

$$t_1 = -\left(\frac{1}{4}\right) \ln(1 - .9) = .577 \text{ hour} = 34.5 \text{ minutes}$$

Note that $\ln(1 - R)$ may be replaced with $\ln(R)$ because $1 - R$ is the complement of R .

PROBLEM SET 19.3B

- *1. In Example 19.3-2, suppose that the first customer arrives at time 0. Use the first three random numbers in column 1 of Table 19.1 to generate the arrival times of the next 3 customers, and graph the resulting events on the timescale.
- *2. *Uniform Distribution.* Suppose that the time needed to manufacture a part on a machine is described by the following uniform distribution:

$$f(t) = \frac{1}{b - a}, a \leq t \leq b$$

- Determine an expression for the sample t , given the random number R .
3. Jobs are received randomly at a one-machine shop. The time between arrivals is exponential with mean 2 hours. The time needed to manufacture a job is uniform between 1.1 and 2 hours. Assuming that the first job arrives at time 0, determine the arrival and departure time for the first five jobs using the (0, 1) random numbers in column 1 of Table 19.1.
 4. The demand for an expensive spare part of a passenger jet is 0, 1, 2, or 3 units per month with probabilities .3, .3, .2, and .2, respectively. The airline maintenance shop starts operation with a stock of 6 units and will bring the stock level back to 6 units immediately after it drops below 5 units.
- (a) Devise the procedure for sampling demand.
 - (b) How many months will elapse until the first replenishment occurs? Use successive values of R from the first column in Table 19.1.
5. In a simulation situation, TV units are inspected for possible defects. There is an 70% chance that a unit will pass inspection, in which case it is sent to packaging. Otherwise, the unit is repaired. We can represent the situation symbolically in one of two ways.

```
goto REPAIR/.3, PACKAGE/.7
goto PACKAGE/.7, REPAIR/.3
```

These two representations appear equivalent. Yet, when a given sequence of (0, 1) random numbers is applied to the two representations, different decisions (REPAIR or PACKAGE) may result. Explain why.

6. A player tosses a fair coin repeatedly until a head occurs. The associated payoff is 3^n , where n is the number of tosses until a head comes up.
 - (a) Devise the sampling procedure of the game.
 - (b) Use the random numbers in column 1 of Table 19.1 to determine the cumulative payoff after two heads occur.
7. *Triangular Distribution.* In simulation, the lack of data may make it impossible to determine the probability distribution associated with a simulation activity. In most of these situations, it may be easy to describe the desired variable by estimating its smallest, most likely, and largest values. These three values are sufficient to define a triangular distribution, which can then be used as "rough cut" estimation of the real distribution.

- (a) Develop the formula for sampling from the following triangular distribution, whose respective parameters are a , b , and c :

$$f(x) = \begin{cases} \frac{2(x - a)}{(b - a)(c - a)}, & a \leq x \leq b \\ \frac{2(c - x)}{(c - b)(c - a)}, & b \leq x \leq c \end{cases}$$

- (b) Generate three samples from a triangular distribution with parameters (1, 3, 7) using the first three random numbers in column 1 of Table 19.1.
8. Consider a probability distribution that consists of a rectangle flanked on the left and right sides by two symmetrical right triangles. The respective ranges for the triangle on the left, the rectangle, and the triangle on the right are $[a, b]$, $[b, c]$, and $[c, d]$, $a < b < c < d$. Both triangles have the same height as the rectangle.
 - (a) Develop a sampling procedure.
 - (b) Determine five samples with $(a, b, c, d) = (1, 2, 4, 6)$ using the first five random numbers in column 1 of Table 19.1.
- *9. *Geometric distribution.* Show how a random sample can be obtained from the following geometric distribution:

$$f(x) = p(1 - p)^x, x = 0, 1, 2, \dots$$

The parameter x is the number of (Bernoulli) failures until a success occurs, and p is the probability of a success, $0 < p < 1$. Generate five samples for $p = .6$, using the first five random numbers in column 1 of Table 19.1.

10. *Weibull distribution.* Show how a random sample can be obtained from the Weibull distribution with the following probability density function:

$$f(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^\alpha}, x > 0$$

where $\alpha > 0$ is the shape parameter, and $\beta > 0$ is the scale parameter.

Convolution method. The basic idea of the convolution method is to express the desired sample as the statistical sum of other easy-to-sample random variables. Typical among these distributions are the Erlang and the Poisson whose samples can be obtained from the exponential distribution samples.

Example 19.3-3 (Erlang Distribution)

The m -Erlang random variable is defined as the statistical sum (convolutions) of m independent and identically distributed exponential random variables. Let y represent the m -Erlang random variable; then

$$y = y_1 + y_2 + \dots + y_m$$

where $y_i, i = 1, 2, \dots, m$, are independent and identically distributed exponentials with the following probability density function:

$$f(y_i) = \lambda e^{-\lambda y_i}, y_i > 0, i = 1, 2, \dots, m$$

From Example 19.3-2, a sample from the i th exponential distribution is computed as

$$y_i = -\left(\frac{1}{\lambda}\right) \ln(R_i), i = 1, 2, \dots, m$$

Thus, the m -Erlang sample is computed as

$$\begin{aligned} y &= -\left(\frac{1}{\lambda}\right) [\ln(R_1) + \ln(R_2) + \dots + \ln(R_m)] \\ &= -\left(\frac{1}{\lambda}\right) \ln\left(\prod_{i=1}^m R_i\right) \end{aligned}$$

To illustrate the use of the formula, suppose that $m = 3$ and $\lambda = 4$ events per hour. The first 3 random numbers in column 1 of Table 19.1 yield $R_1 R_2 R_3 = (.0589)(.6733)(.4799) = .0190$, which yields

$$y = -\left(\frac{1}{4}\right) \ln(.019) = .991 \text{ hour}$$

Example 19.3-4 (Poisson Distribution)

Section 18.4.1 shows that if the distribution of the time between the occurrences of successive events is exponential, then the distribution of the number of events per unit time is Poisson, and vice versa. We use this relationship to sample the Poisson distribution.

Assume that mean of the Poisson distribution is λ events per unit time. It follows that the time between events is exponential with mean $\frac{1}{\lambda}$ time units. This means that a Poisson sample, n , will occur during t time units if, and only if,

Period till event n occurs $\leq t <$ Period till event $n + 1$ occurs
This condition translates to

$$\begin{aligned} t_1 + t_2 + \dots + t_n &\leq t < t_1 + t_2 + \dots + t_{n+1}, n > 0 \\ 0 &\leq t < t_1, n = 0 \end{aligned}$$

The random variable $t_i, i = 1, 2, \dots, n + 1$, is a sample from the exponential distribution with mean $\frac{1}{\lambda}$. From the result in Example 19.3-3, we have

$$\begin{aligned} -\left(\frac{1}{\lambda}\right) \ln\left(\prod_{i=1}^n R_i\right) &\leq t < -\left(\frac{1}{\lambda}\right) \ln\left(\prod_{i=1}^{n+1} R_i\right), n > 0 \\ 0 &\leq t < -\left(\frac{1}{\lambda}\right) \ln(R_1), n = 0 \end{aligned}$$

These expressions reduce to

$$\begin{aligned} \prod_{i=1}^n R_i &\geq e^{-\lambda t} > \prod_{i=1}^{n+1} R_i, n > 0 \\ 1 &\geq e^{-\lambda t} > R_1, n = 0 \end{aligned}$$

To illustrate the implementation of the sampling process, suppose that $\lambda = 4$ events per hour. To obtain a sample for a period $t = .5$ hour, we first compute $e^{-\lambda t} = .1353$. The random number $R_1 = .0589$ is less than $e^{-\lambda t} = .1353$. Hence, the corresponding sample is $n = 0$.

Example 19.3-5 (Normal Distribution)

The central limit theorem (see Section 14.4.4) states that the sum (convolution) of n independent and identically distributed random variables becomes asymptotically normal as n becomes sufficiently large. We use this result to generate samples from normal distribution with mean μ and standard deviation σ .

Define

$$x = R_1 + R_2 + \dots + R_n$$

The random variable is asymptotically normal by the central limit theorem. Given that the uniform $(0, 1)$ random number R has a mean of $\frac{1}{2}$ and a variance of $\frac{1}{12}$, it follows that the mean and variance of x are $\frac{n}{2}$ and $\frac{n}{12}$, respectively. Thus, a random sample, y , from a normal distribution $N(\mu, \sigma)$, with mean μ and standard deviation σ , can be computed from x as

$$y = \mu + \sigma \left(\frac{x - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \right)$$

In practice, we take $n = 12$ for convenience, which reduces the formula to

$$y = \mu + \sigma(x - 6)$$

To illustrate the use of this method, suppose that we wish to generate a sample from $N(10, 2)$ (mean $\mu = 10$ and standard deviation $\sigma = 2$). Taking the sum of the first 12 random numbers in columns 1 and 2 of Table 19.1, we get $x = 6.1094$. Thus, $y = 10 + 2(6.1094 - 6) = 10.2188$.

Box-Muller normal sampling formula. The disadvantage of the preceding procedure is that it requires generating 12 random numbers per normal sample, which is computationally inefficient. A more efficient procedure calls for using the transformation

$$x = \cos(2\pi R_2) \sqrt{-2 \ln(R_1)}$$

Box and Muller (1958) prove that x is a standard $N(0, 1)$. Thus, $y = \mu + \sigma x$ will produce a sample from $N(\mu, \sigma)$. The new procedure is more efficient because it requires two 0-1 random numbers only. Actually, this method is even more efficient than stated, because Box and Muller prove that the given formula produces another $N(0, 1)$ sample if $\sin(2\pi R_2)$ replaces $\cos(2\pi R_2)$.

To illustrate the implementation of the Box-Muller procedure to the normal distribution $N(10, 2)$, the first two random numbers in column 1 of Table 19.1 yield the following $N(0, 1)$ samples:

$$x_1 = \cos(2\pi \times .6733) \sqrt{-2 \ln(.0589)} \approx -1.103$$

$$x_2 = \sin(2\pi \times .6733) \sqrt{-2 \ln(.0589)} \approx -2.109$$

Thus, the corresponding $N(10, 2)$ samples are

$$y_1 = 10 + 2(-1.103) = 7.794$$

$$y_2 = 10 + 2(-2.109) = 5.782$$

PROBLEM SET 19.3C¹

- *1. In Example 19.3-3, compute an Erlang sample, given $m = 3$ and $\lambda = 10$ events per hour.
2. In Example 19.3-4, generate three Poisson samples during a half-hour period, given that the mean of the Poisson is 9 events per hour.
3. In Example 19.4-5, generate two samples from $N(7, 2)$ by using both the convolution method and the Box-Muller method.
4. Jobs arrive at Metalco Jobshop according to a Poisson distribution, with a mean of 6 jobs per day. Received jobs are assigned to the five machining centers of the shop on a strict rotational basis. Determine one sample of the interval between the arrival of jobs at the first machine center.
5. The ACT scores for the 1994 senior class at Springdale High is normal, with a mean of 27 points and a standard deviation of 3 points. Suppose that we draw a random sample of six seniors from that class. Use the Box-Muller method to determine the mean and standard deviation of the sample.
- *6. Psychology professor Yataha is conducting a learning experiment in which mice are trained to find their way around a maze. The base of the maze is square. A mouse enters the maze at one of the four corners and must find its way through the maze to exit at the same point where it entered. The design of the maze is such that the mouse must pass by each of the remaining three corner points exactly once before it exits. The multipaths of the maze connect the four corners in a strict clockwise order. Professor Yataha estimates that the time the mouse takes to reach one corner point from another is uniformly distributed between 10 and 20 seconds, depending on the path it takes. Develop a sampling procedure for the time a mouse spends in the maze.
7. In Problem 6, suppose that once a mouse makes an exit from the maze, another mouse instantly enters. Develop a sampling procedure for the number of mice that exit the maze in 5 minutes.

¹For all the problems of this set, use the random numbers in Table 19.1 starting with column 1.

8. *Negative Binomial.* Show how a random sample can be determined from the negative binomial whose distribution is given as

$$f(x) = C_r^{x-1} p^r (1-p)^x, x = 0, 1, 2, \dots$$

where x is the number of failures until the r th success occurs in a sequence of independent Bernoulli trials and p is the probability of success, $0 < p < 1$. (*Hint:* The negative binomial is the convolution of r independent geometric samples. See Problem 9, Set 19.3B.)

19.4 GENERATION OF RANDOM NUMBERS

Uniform (0, 1) random numbers play a key role in sampling from distributions. True 0-1 random numbers can be generated by electronic devices only. However, because simulation models are executed on the computer, the use of electronic devices to generate random numbers is much too slow for that purpose. Additionally, electronic devices are activated by laws of chance, making it impossible to duplicate the same sequence of random numbers at will. This point is important because debugging, verification, and validation of the simulation model often require duplicating the random numbers sequence.

The only feasible way for generating 0-1 random numbers for use in simulation is based on arithmetic operations. Such numbers are not truly random because the entire sequence can be generated in advance. It is thus more appropriate to refer to them as **pseudorandom numbers**.

The most common arithmetic operation for generating (0, 1) random numbers is the **multiplicative congruential method**. Given the parameters u_0 , b , c , and m , a pseudorandom number R_n can be generated from the formulas:

$$u_n = (bu_{n-1} + c) \bmod(m), n = 1, 2, \dots$$

$$R_n = \frac{u_n}{m}, n = 1, 2, \dots$$

The initial value u_0 is usually referred to as the **seed** of the generator.

Variations of the multiplicative congruential method that improve the quality of the generator can be found in Law (2007).

Example 19.4-1

Generate three random numbers based on the multiplicative congruential method using $b = 9$, $c = 5$, and $m = 12$. The seed is $u_0 = 11$.

$$u_1 = (9 \times 11 + 5) \bmod 12 = 8, R_1 = \frac{8}{12} = .6667$$

$$u_2 = (9 \times 8 + 5) \bmod 12 = 5, R_2 = \frac{5}{12} = .4167$$

$$u_3 = (9 \times 5 + 5) \bmod 12 = 2, R_3 = \frac{2}{12} = .1667$$

Excel Moment

Excel template *excelRN.xls* implements the multiplicative congruential method. Figure 19.6 generates the sequence associated with the parameters of Example 19.4-1. Notice that the cycle length is exactly 4, after which the sequence repeats itself. The point to be made here is that the selected values of u_0 , b , c , and m are critical in determining the (statistical) quality of the generator and its cycle length. Thus, "casual" implementation of the congruential formula is not recommended. Instead, one must use a reliable and tested generator. All commercial computer programs are equipped with dependable random number generators.

PROBLEM SET 19.4A

- *1. Use *excelRN.xls* with the following sets of parameters, and compare the results with those of Example 19.4-1:

$$b = 17, c = 111, m = 103, \text{seed} = 7$$

FIGURE 19.6

Excel random numbers output for the data of Example 19.4-1 (file *excelRN.xls*)

	A	B
1	Multiplicative Congruential Method	
2	Input data(B7<=1000)	
3	b =	9
4	c =	5
5	u0 =	11
6	m =	12
7	How many numbers?	10
8	Output results	
9	Press to Generate Sequence	
10	Generated random numbers:	
11	1	0.66667
12	2	0.41667
13	3	0.16667
14	4	0.91667
15	5	0.66667
16	6	0.41667
17	7	0.16667
18	8	0.91667
19	9	0.66667
20	10	0.41667

- Find a random number generator on your computer, and use it to generate 500 zero-one random numbers. Histogram the resulting values (using the Microsoft histogram tool, see Section 12.5) and visually convince yourself that the obtained numbers reasonably follow the (0, 1) uniform distribution. Actually, to test the sequence properly, you would need to apply the following tests: chi-square goodness of fit (see Section 14.6), runs test for independence, and correlation test—see Law (2007) for details.

19.5 MECHANICS OF DISCRETE SIMULATION

This section details how typical statistics are collected in a simulation model. The vehicle of explanation is a single-queue model. Section 19.5.1 uses a numeric example to detail the actions and computations that take place in a single-server queuing simulation model. Because of the tedious computations that typify the execution of a simulation model, Section 19.5.2 shows how the single-server model is modeled and executed using Excel spreadsheet.

19.5.1 Manual Simulation of a Single-Server Model

The interarrival time of customers at HairKare Barbershop is exponential with mean 15 minutes. The shop is operated by only one barber, and it takes between 10 and 15 minutes, uniformly distributed, to do a haircut. Customers are served on a first-in, first-out (FIFO) basis. The objective of the simulation is to compute the following measures of performance:

- The average utilization of the shop.
- The average number of waiting customers.
- The average time a customer waits in queue.

The logic of the simulation model can be described in terms of the actions associated with the arrival and departure events of the model.

Arrival Event

- Generate and store chronologically the occurrence time of the next arrival event (= current simulation time + interarrival time).
- If the facility (barber) is idle
 - Start service and declare the facility busy. Update the facility utilization statistics.
 - Generate and store chronologically the time of the departure event for the customer (= current simulation time + service time).
- If the facility is busy, place the customer in the queue, and update the queue statistics.

Departure Event

- If the queue is empty, declare the facility idle. Update the facility utilization statistics.
- If the queue is not empty
 - Select a customer from the queue, and place it in the facility. Update the facility utilization and queue statistics.

- b. Generate and store chronologically the occurrence time of the departure event for the customer (= current simulation time + service time).

From the data of the problem, the interarrival time is exponential with mean 15 minutes, and the service time is uniform between 10 and 15 minutes. Letting p and q represent random samples of interarrival and service times, then, as explained in Section 19.3.2, we get

$$p = -15 \ln(R) \text{ minutes, } 0 \leq R \leq 1$$

$$q = 10 + 5R \text{ minutes, } 0 \leq R \leq 1$$

For the purpose of this example, we use R from Table 19.1, starting with column 1. We also use the symbol T to represent the simulation clock time. We further assume that the first customer arrives at $T = 0$ and that the facility starts empty.

Because the simulation computations are typically voluminous, the simulation is limited to the first 5 arrivals only. The example is designed to cover all possible situations that could arise in the course of the simulation. Later in Section 19.5.2, we introduce the template *excelSingleServer.xls* that allows experimenting with the model without the need to carry out the computations manually.

Arrival of customer 1 at $T = 0$. Generate the arrival of customer 2 at

$$T = 0 + p_1 = 0 + [-15 \ln(.0589)] = 42.48 \text{ minutes}$$

Because the facility is idle at $T = 0$, customer 1 starts service immediately. The departure time is thus computed as

$$T = 0 + q_1 = 0 + (10 + 5 \times .6733) = 13.37 \text{ minutes}$$

The *chronological* list of future events thus becomes

Time, T	Event
13.37	Departure of customer 1
42.48	Arrival of customer 2

Departure of customer 1 at $T = 13.37$. Because the queue is empty, the facility is declared idle. At the same time, we record that the facility has been busy between $T = 0$ and $T = 13.37$ minutes. The updated list of future events becomes

Time, T	Event
42.48	Arrival of customer 2

Arrival of customer 2 at $T = 42.48$. Customer 3 will arrive at

$$T = 42.48 + [-15 \ln(.4799)] = 53.49 \text{ minutes}$$

Because the facility is idle, customer 2 starts service, and the facility is declared busy. The departure time is

$$T = 42.48 + (10 + 5 \times .9486) = 57.22 \text{ minutes}$$

The list of future events is updated as

Time, T	Event
53.49	Arrival of customer 3
57.22	Departure of customer 2

Arrival of customer 3 at $T = 53.49$. Customer 4 will arrive at

$$T = 53.49 + [-15 \ln(.6139)] = 60.81 \text{ minutes}$$

Because the facility is currently busy (until $T = 57.22$), customer 3 is placed in queue at $T = 53.49$. The updated list of future events is

Time, T	Event
57.22	Departure of customer 2
60.81	Arrival of customer 4

Departure of customer 2 at $T = 57.22$. Customer 3 is taken out of the queue to start service. The waiting time is

$$W_3 = 57.22 - 53.49 = 3.73 \text{ minutes}$$

The departure time is

$$T = 57.22 + (10 + 5 \times .5933) = 70.19 \text{ minutes}$$

The updated list of future events is

Time, T	Event
60.81	Arrival of customer 4
70.19	Departure of customer 3

Arrival of customer 4 at $T = 60.81$. Customer 5 will arrive at

$$T = 60.81 + [-15 \ln(.9341)] = 61.83 \text{ minutes}$$

Because the facility is busy until $T = 70.19$, customer 4 is placed in the queue. The updated list of future events is

Time, T	Event
61.83	Arrival of customer 5
70.19	Departure of customer 3

Arrival of customer 5 at $T = 61.83$. The simulation is limited to 5 arrivals, hence customer 6 arrival is not generated. The facility is still busy, hence the customer is placed in queue at $T = 61.83$. The updated list of events is

Time, T	Event
70.19	Departure of customer 3

Departure of customer 3 at $T = 70.19$. Customer 4 is taken out of the queue to start service. The waiting time is

$$W_4 = 70.19 - 60.81 = 9.38 \text{ minutes}$$

The departure time is

$$T = 70.19 + [10 + 5 \times .1782] = 81.08 \text{ minutes}$$

The updated list of future events is

Time, T	Event
81.08	Departure of customer 4

Departure of customer 4 at $T = 81.08$. Customer 5 is taken out of the queue to start service. The waiting time is

$$W_5 = 81.08 - 61.83 = 19.25 \text{ minutes}$$

The departure time is

$$T = 81.08 + (10 + 5 \times .3473) = 92.82 \text{ minutes}$$

The updated list of future events is

Time, T	Event
92.82	Departure of customer 5

Departure of customer 5 at $T = 92.82$. There are no more customers in the system (queue and facility) and the simulation ends.

Figure 19.7 summarizes the changes in the length of the queue and the utilization of the facility as a function of the simulation time.

The queue length and the facility utilization are known as **time-based** variables because their variation is a function of time. As result, their average values are computed as

$$\left(\begin{array}{l} \text{Average value of a} \\ \text{time-based variable} \end{array} \right) = \frac{\text{Area under curve}}{\text{Simulated period}}$$

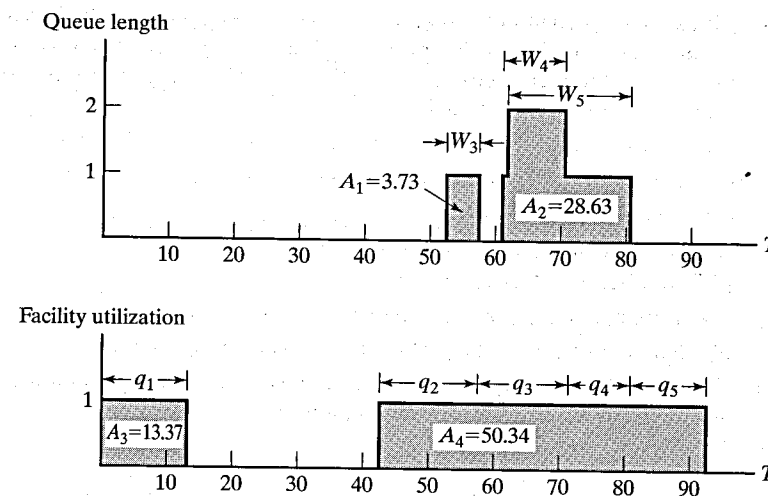


FIGURE 19.7

Changes in queue length and facility utilization as a function of simulation time, T

Implementing this formula for the data in Figure 19.7, we get

$$\left(\begin{array}{l} \text{Average queue} \\ \text{length} \end{array} \right) = \frac{A_1 + A_2}{92.82} = \frac{32.36}{92.82} = .349 \text{ customer}$$

$$\left(\begin{array}{l} \text{Average facility} \\ \text{utilization} \end{array} \right) = \frac{A_3 + A_4}{92.82} = \frac{63.71}{92.82} = .686 \text{ barber}$$

The average waiting time in the queue is an **observation-based** variable whose value is computed as

$$\left(\begin{array}{l} \text{Average value of an} \\ \text{observation-based variable} \end{array} \right) = \frac{\text{Sum of observations}}{\text{Number of observations}}$$

Examination of Figure 19.7 reveals that the area under the queue-length curve actually equals the sum of the waiting time for the three customers who joined the queue; namely,

$$W_1 + W_2 + W_3 + W_4 + W_5 = 0 + 0 + 3.73 + 9.38 + 19.25 = 32.36 \text{ minutes}$$

The average waiting time in the queue for all customers is thus computed as

$$\bar{W}_q = \frac{32.36}{5} = 6.47 \text{ minutes}$$

PROBLEM SET 19.5A

1. Suppose that the barbershop in Section 19.5.1 is operated by two barbers, and customers are served on a FCFS basis. Suppose further that the time to get a haircut is uniformly distributed between 15 and 30 minutes. The interarrival time of customers is exponential,

with a mean of 10 minutes. Simulate the system manually for 75 time units. From the results of the simulation, determine the average time a customer waits in queue, the average number of customers waiting, and the average utilization of the barbers. Use the random numbers in Table 19.1.

2. Classify the following variables as either *observation based* or *time based*:

- ***(a)** Time-to-failure of an electronic component.
- ***(b)** Inventory level of an item.
- (c)** Order quantity of an inventory item.
- (d)** Number of defective items in a lot.
- (e)** Time needed to grade test papers.
- (f)** Number of cars in the parking lot of a car-rental agency.

- *3.** The following table represents the variation in the number of waiting customers in a queue as a function of the simulation time.

Simulation time, T (hr)	No. of waiting customers
$0 \leq T \leq 3$	0
$3 < T \leq 4$	1
$4 < T \leq 6$	2
$6 < T \leq 7$	1
$7 < T \leq 10$	0
$10 < T \leq 12$	2
$12 < T \leq 18$	3
$18 < T \leq 20$	2
$20 < T \leq 25$	1

Compute the following measures of performance:

- (a) The average length of the queue.
- (b) The average waiting time in the queue for those who must wait.

4. Suppose that the barbershop described at the start of Section 19.5.1 is operated by three barbers. Assume further that the utilization of the servers (barbers) is summarized as given in the following table:

Simulation time, T (hr)	No. of busy servers
$0 < T \leq 10$	0
$10 < T \leq 20$	1
$20 < T \leq 30$	2
$30 < T \leq 40$	0
$40 < T \leq 60$	1
$60 < T \leq 70$	2
$70 < T \leq 80$	3
$80 < T \leq 90$	1
$90 < T \leq 100$	0

Determine the following measures of performance:

- The average utilization of the facility.
- The average busy time of the facility.
- The average idle time of the facility.

19.5.2 Spreadsheet-Based Simulation of the Single-Server Model

This section develops a spreadsheet-based model for the single server model. The objective of the development is to reinforce the ideas introduced in Section 19.5.1. Of course, a single-server model is a simple situation that can be modeled readily in a spreadsheet environment. Other situations require more involved modeling effort, a task that is facilitated by available simulation packages (see Section 19.7).

The presentation in Section 19.5.1 shows that the simulation model of the single-server facility requires two basic elements:

1. A chronological list of the model's events.
2. A graph that keeps track of the changes in facility utilization and queue length.

These two elements remain essential in the development of the spreadsheet-based (indeed, any computer-based) simulation model. The difference is that the implementation is realized in a manner that is compatible with the use of the computer. As in Section 19.5.1, customers are served in order of arrival (FIFO).

Figure 19.8 provides the output of *excelSingleServer.xls*. The input data allow representing the interarrival and service time in one of four ways: constant, exponential, uniform, and triangular. The triangular distribution is useful in that it can be used as a rough initial estimate of any distribution, simply by providing three estimates a , b , and c that represent the smallest, the most likely, and the largest values of the interarrival or

FIGURE 19.8

Excel output of a single-server simulation model (file *excelSingleServer.xls*)

	A	B	C	D	E	F	G	H	K	L	M	N	O	P	Q										
1	Simulation of a Single-Server Queueing Model																								
2	Nbr of arrivals = 20 << Maximum 500																								
3	Enter x in column A to select interarrival pdf:									Nbr InterArrT	ServiceT	ArrT	DepartT	Wq	Ws										
4	Constant =									1	3.73	12.83	0.00	12.83	0.00	12.83									
5	x	Exponential: $\lambda =$		0.067					2	5.37	14.71	3.73	27.55	9.10	23.82										
6	Uniform: a =		b =						3	3.86	12.21	9.09	39.75	18.45	30.66										
7	Triangular: a =		b =		c =				4	14.10	11.18	12.95	50.94	26.80	37.98										
8	Enter x in column A to select service time pdf:									5	7.35	14.92	27.05	65.85	23.89	38.00									
9	Constant =									6	35.70	14.22	34.41	80.07	31.45	45.67									
10	Exponential: $\mu =$									7	0.60	14.50	70.11	94.58	9.97	24.47									
11	x	Uniform: a =		10		b =		15		8	4.25	13.35	70.71	107.93	23.87	37.22									
12	Triangular: a =		b =		c =				9	4.85	12.45	74.96	120.38	32.97	45.41										
13	Output Summary									10	7.43	11.57	79.81	131.94	40.56	52.13									
14	Av. facility utilization = 0.98									11	8.99	14.65	87.24	146.59	44.70	59.34									
15	Percent idleness (%) = 1.95									12	49.78	12.95	96.23	159.43	50.36	63.20									
16	Press F9 to trigger a new simulation run.									13	0.42	14.12	146.01	173.55	13.43	27.54									
17										Av. queue length, Lq = 1.57									14	8.77	13.69	146.43	187.24	27.13	40.82
18										Av. nbr in system, Ls = 2.56									15	11.19	10.50	155.20	197.75	32.05	42.55
19										Av. queue time, Wq = 21.24									16	42.82	13.78	166.38	211.53	31.36	45.14
20										Av. system time, Ws = 34.47									17	19.87	12.29	209.20	223.82	2.33	14.62
21										Sum(ServiceTime) = 264.65									18	9.25	12.95	229.07	242.03	0.00	12.95
22	Sum(Wq) = 424.80									19	13.98	12.99	238.33	255.02	3.70	16.69									
23	Sum(Ws) = 689.44									20	58.46	14.88	252.31	269.90	2.71	17.59									

service time. The only other information needed to drive the simulation is the length of the simulation run, which in this model is specified by the number of arrivals that can be generated in the model.

The spreadsheet calculations reserve one row for each arrival. The interarrival and service times for each arrival are generated from the input data. The first arrival is assumed to occur at $T = 0$. Because the facility starts idle, the customer starts service immediately. The spreadsheet provides sufficient information to demonstrate the internal computations given in Section 19.5.1.

Another spreadsheet was developed for simulating multiserver models (*excelMultiServer.xls*). The design of the template is based on the same ideas used in the single-server case. However, the determination of the departure time is not as straightforward and requires the use of VBA macros.

PROBLEM SET 19.5B

1. Using the input data in Section 19.5.1, run the Excel simulator for 10 arrivals and graph the changes in facility utilization and queue length as a function of the simulation time. Verify that the areas under the curves equal the sum of the service times and the sum of the waiting times, respectively.
2. Simulate the $M/M/1$ model for 500 arrivals, given the arrival rate $\lambda = 4$ customers per hour and the service rate $\mu = 6$ departures per hour. Run 5 replications (by refreshing the spreadsheet—pressing F9) and determine a 95% confidence interval for all the measures of performance of the model. Compare the results with the steady-state theoretical values of the $M/M/1$ model.
3. Television units arrive on a conveyor belt every 15 minutes for inspection at a single-operator station. Detailed data for the inspection station are not available. However, the operator estimates that it takes 10 minutes “on the average” to inspect a unit. Under worst conditions, the inspection time does not exceed 13 minutes, and for certain units, inspection time may be as low as 9 minutes.
 - (a) Use the Excel simulator to simulate the inspection of 200 TV units.
 - (b) Based on 5 replications, estimate the average number of units awaiting inspection and the average utilization of the inspection station.

19.6 METHODS FOR GATHERING STATISTICAL OBSERVATIONS

Simulation is a statistical experiment, and its output must be interpreted using proper statistical inference tools (e.g., confidence intervals and hypothesis testing). To accomplish this task, a simulation experiment must satisfy three conditions:

1. Observations are drawn from stationary (identical) distributions.
2. Observations are sampled from a normal population.
3. Observations are independent.

In a strict sense, the simulation experiment does not satisfy any of these conditions. Nevertheless, we can ensure that these conditions remain statistically acceptable by restricting the manner in which the observations are gathered.

First, we consider the issue of stationary distributions. Simulation output is a function of the length of the simulated period. The initial period produces erratic behavior and is usually referred to as the **transient** or **warm-up period**. When the output stabilizes, the system operates under **steady state**. Unfortunately, there is no definitive way to predict the start point of steady state in advance. In general, a longer simulation run has better chance of reaching steady state—meaning that the problem is addressed by using a sufficiently large sample size.

Next, we consider the requirement that simulation observations are drawn from a normal population. This requirement is realized by using the *central limit theorem* (see Section 14.4.4), which confirms that the distribution of the average of a sample is asymptotically normal regardless of the parent population. The central limit theorem is thus the main tool we use for satisfying the normal distribution assumption.

The third condition deals with the independence of the observations. In simulation, an observation can be based on a single independent run, or by subdividing a single run into subintervals each representing an observation. Each method has its advantages and disadvantages. The first method alleviates the question of independence but has the disadvantage of including the transient period in each observation. In the second method, the effect of the transient period is not as pronounced, but it inherently worsens the issue of independence. As will be explained subsequently in this section, a possible remedy calls for increasing the length of the simulation run.

The most common methods for collecting observations in simulation are

1. Subinterval method.
2. Replication method.
3. Regenerative (or cycles) method.

The first two methods can be readily automated in all widely used simulation languages (see Section 19.7). On the other hand, the third method, though it addresses directly the issue of independence by seeking identical starting conditions for the different observations, may be difficult to implement in practice.

Sections 19.6.1 and 19.6.2 present the first two methods. Details of the third method can be found in Law (2007).

19.6.1 Subinterval Method

Figure 19.9 illustrates the idea of the subinterval method. Suppose that the length of the simulation run is T time units. The subinterval method first truncates an initial transient period, and then subdivides the remainder of the simulation run into n equal subintervals (or batches). The average of a desired measure of performance (e.g., queue length or waiting time in queue) within each subinterval is then used to represent a single observation. Truncation of the initial transient period means that no statistical data are collected during that period.

The advantage of the subinterval method is that the effect of the transient (non-stationary) conditions is mitigated, particularly for the observations that are collected toward the end of the simulation run. The disadvantage is that successive batches with common boundary conditions are not necessarily independent. The problem can be alleviated by increasing the time base for each observation.

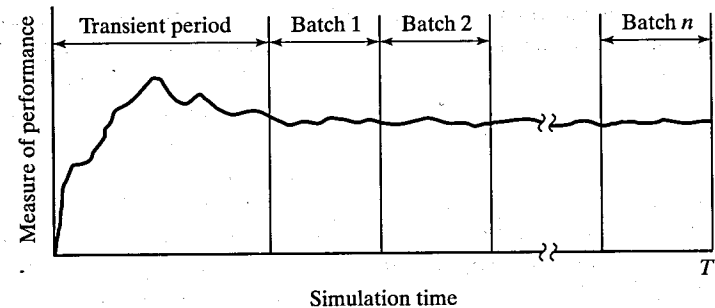


FIGURE 19.9
Collecting simulation data using the subinterval method

Example 19.6-1

Figure 19.10 shows the change in queue length in a single-queue model as a function of the simulation time. The simulation run length is $T = 35$ hours, and the length of the transient period is estimated to equal 5 hours. The time base for an observation is 6 hours, which produces $n = 5$ observation.

Let \bar{Q}_i represent the average queue length in batch i . Because the queue length is a time-based variable, we have

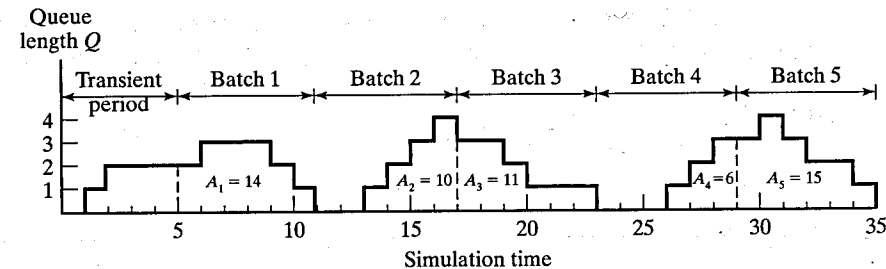
$$\bar{Q}_i = \frac{A_i}{t}, i = 1, 2, \dots, 5$$

where A_i is the area under the queue length curve associated with batch (observation) i , and $t (= 6)$ is the time base per batch.

The data in Figure 19.10 produce the following observations:

Observation i	1	2	3	4	5
A_i	14	10	11	6	15
\overline{Q}_i	2.33	1.67	1.83	1.00	2.50
Sample mean = 1.87		Sample standard deviation = .59			

FIGURE 19.10
Change in queue length with simulation time in Example 19.6-1



The sample mean and variance can be used to compute a confidence interval, if desired. The computation of the sample variance in Example 19.6-1 is based on the following familiar formula:

$$s = \sqrt{\frac{\sum_{i=1}^n x_i^2 - n\bar{x}^2}{n - 1}}$$

This formula is only an approximation of the true standard deviation because it ignores the effect of autocorrelation between the successive batches. The exact formula can be found in Law (2007).

19.6.2 Replication Method

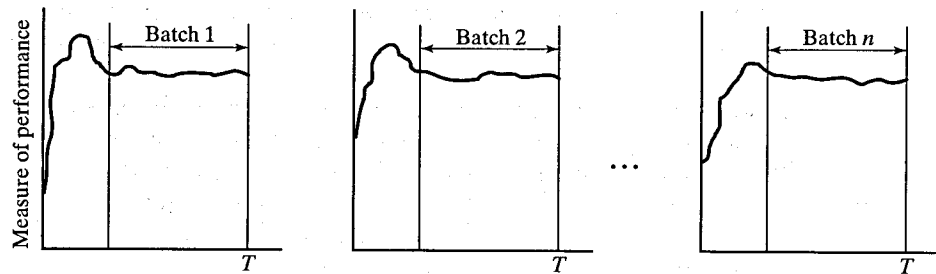
In the replication method, each observation is represented by an independent simulation run in which the transient period is truncated, as illustrated in Figure 19.11. The computation of the observation averages for each batch is the same as in the subinterval method. The only difference is that the standard variance formula is applicable because the batches are not independent.

The advantage of the replication method is that each simulation run is driven by a distinct 0-1 random number stream, which yields statistically independent observations. The disadvantage is that each observation may be biased by the initial effect of the transient conditions. Such a problem may be alleviated by making the run length sufficiently large.

PROBLEM SET 19.6A

1. In Example 19.6-1, use the subinterval method to compute the average waiting time in the queue for those who must wait.
- *2. In a simulation model, the subinterval method is used to compute batch averages. The transient period is estimated to be 100, and each batch has a time base of 100 time units as well. Using the following data, which provide the waiting times for customers as a function of the simulation time, estimate the 95% confidence interval for the mean waiting time.

FIGURE 19.11
Collecting simulation data using the replication method



Time interval	Waiting times
0-100	10, 20, 13, 14, 8, 15, 6, 8
100-200	12, 30, 10, 14, 16
200-300	15, 17, 20, 22
300-400	10, 20, 30, 15, 25, 31
400-500	15, 17, 20, 14, 13
500-600	25, 30, 15

19.7 SIMULATION LANGUAGES

Execution of simulation models entails two distinct types of computations: (1) file manipulations that deal with the chronological storage and processing of model events and (2) arithmetic and bookkeeping computations associated with generation of random samples and collection of model statistics. The first type of computation involves extensive logic in the development of list processing, and the second type entails tedious and time-consuming calculations. The nature of these computations makes the computer an essential tool for executing simulation models, and, in turn, prompts the development of special computer simulation languages for performing these computations conveniently and efficiently.

Available discrete simulation languages fall into two broad categories:

1. Event scheduling
2. Process oriented

In event scheduling languages, the user details the actions associated with the occurrence of each event, in much the same way they are given in Example 19.5-1. The main role of the language in this case is (1) automation of sampling from distributions, (2) storage and retrieval of events in chronological order, and (3) collection of model statistics.

Process-oriented languages use blocks or nodes that can be linked together to form a network that describes the movements of **transactions** or **entities** (i.e., customers) in the system. For example, the three most prominent blocks/nodes in any process-simulation language are a *source* from which transactions are created, a *queue* where they can wait if necessary, and a *facility* where service is performed. Each of these blocks/nodes is defined with all the information needed to drive the simulation automatically. For example, once the interarrival time for the source is specified, a process-oriented language automatically “knows” when arrival events will occur. In effect, each block/node of the model has standing instructions that define *how* and *when* transactions are moved in the simulation network.

Process-oriented languages are internally driven by the same actions used in event-scheduling languages. The difference is that these actions are automated to relieve the user of the tedious computational and logical details. In a way, we can regard process-oriented languages as being based on the input-output concept of the “black box” approach. This essentially means that process-oriented languages trade modeling flexibility for simplicity and ease of use.

Event-scheduling languages (such as SIMSCRIPT, SLAM, and SIMAN) are outdated and are rarely used in practice. Recently, a new language called DEEDS (Elizandro and Taha, 2008) is based on the novice approach of using Excel spreadsheet to drive event scheduling. DEEDS allows the modeling flexibility of event-driven simulation languages while achieving the intuitive nature of a process-oriented language.

The predominant process-oriented commercial package is Arena. It uses extensive user interface to simplify the process of creating a simulation model. It also provides animation capabilities where changes in the system can be observed visually. However, to an experienced simulation professional, these interfaces may appear to reduce the development of a simulation model to a “slow-motion” pace. It is not surprising that some users continue to prefer writing simulation models in higher-level programming languages.

PROBLEM SET 19.7A²

1. Patrons arrive randomly at a three-clerk post office. The interarrival time is exponential with mean 5 minutes. The time a clerk spends with a patron is exponential with a mean of 10 minutes. All arriving patrons form one queue and wait for the first available free clerk. Run a simulation model of the system for 480 minutes to determine the following:
 - (a) The average number of patrons waiting in the queue.
 - (b) The average utilization of the clerks.
 - (c) Compare the simulation results with those of the *M/M/c* queuing model (Chapter 18) and with the spreadsheet *MultiServerSimulator.xls*.
2. Television units arrive for inspection on a conveyor belt at the constant rate of 5 units per hour. The inspection time takes between 10 and 15 minutes, uniformly distributed. Past experience shows that 20% of inspected units must be adjusted and then sent back for reinspection. The adjustment time is also uniformly distributed between 6 and 8 minutes. Run a simulation model for 480 minutes to compute the following:
 - (a) The average time a unit takes until it passes inspection.
 - (b) The average number of times a unit must be reinspected before it exits the system.
3. A mouse is trapped in a maze and desperately “wants out.” After trying between 1 and 3 minutes, uniformly distributed, there is a 30% chance that it will find the right path. Otherwise, it will wander around aimlessly for between 2 and 3 minutes, uniformly distributed, and eventually end up where it started, only to try once again. The mouse can “try freedom” as many times as it pleases, but there is a limit to everything. With so much energy expended in trying and retrying, the mouse is certain to expire if it does not make it within a period that is normally distributed, with a mean of 10 minutes and a standard deviation of 2 minutes. Write a simulation model to estimate the probability that the mouse will be free. For the purpose of estimating the probability, assume that 100 mice will be processed by the model.
4. In the final stage of automobile manufacturing, a car moving on a transporter is situated between two parallel workstations to allow work to be done on both the left and right

²Work these problems using a simulation language of your choice or a higher-order programming language.

sides of the car simultaneously. The operation times for the left and right sides are uniform between 15 and 20 minutes and 18 and 22 minutes, respectively. The transporter arrives at the stations area every 20 minutes. Simulate the process for 480 minutes to determine the utilization of the left and right stations.

5. Cars arrive at a one-bay car wash facility where the interarrival time is exponential, with a mean of 10 minutes. Arriving cars line up in a single lane that can accommodate at most five waiting cars. If the lane is full, newly arriving cars will go elsewhere. It takes between 10 and 15 minutes, uniformly distributed, to wash a car. Simulate the system for 960 minutes, and estimate the time a car spends in the facility.

BIBLIOGRAPHY

- ✓ Banks, J., J. Carson, B. Nelson, and D. Nicol, *Discrete-Event System Simulation*, 4th ed., Prentice Hall, Upper Saddle River, NJ, 2005.
- Box, G., and M. Muller, "A Note on the Generation of Random Normal Deviates," *Annals of Mathematical Statistics*, Vol. 29, pp. 610–611, 1958.
- Elizandro, D., and H. Taha, *Simulation of Industrial Systems: Discrete Event Simulation Using Excel/VBA*, Taylor and Francis, New York, 2008.
- ✓ Law, A., *Simulation Modeling & Analysis*, 4th ed., McGraw-Hill, New York, 2007.
- Rubenstein, R., B. Melamed, and A. Shapiro, *Modern Simulation and Modeling*, Wiley, New York, 1998.
- Taha, H., *Simulation Modeling and SIMNET*, Prentice Hall, Upper Saddle River, NJ, 1988.

CHAPTER 20

Classical Optimization Theory

20.1 UNCONSTRAINED PROBLEMS

An extreme point of a function $f(\mathbf{X})$ defines either a maximum or a minimum of the function. Mathematically, a point $\mathbf{X}_0 = (x_1^0, \dots, x_j^0, \dots, x_n^0)$ is a maximum if

$$f(\mathbf{X}_0 + \mathbf{h}) \leq f(\mathbf{X}_0)$$

for all $\mathbf{h} = (h_1, \dots, h_j, \dots, h_n)$, where $|h_j|$ is sufficiently small for all j . In a similar manner, \mathbf{X}_0 is a minimum if

$$f(\mathbf{X}_0 + \mathbf{h}) \geq f(\mathbf{X}_0)$$

Figure 20.1 illustrates the maxima and minima of a single-variable function $f(x)$ defined in the range $a \leq x \leq b$. The points x_1, x_2, x_3, x_4 , and x_6 are all extrema of $f(x)$, with x_1, x_3 , and x_6 as maxima and x_2 and x_4 as minima. The value $f(x_6) = \max\{f(x_1), f(x_3), f(x_6)\}$ is a **global** or **absolute** maximum, and $f(x_1)$ and $f(x_3)$ are **local** or **relative** maxima. Similarly, $f(x_4)$ is a local minimum and $f(x_2)$ is a global minimum.

Although x_1 (in Figure 20.1) is a (local) maximum point, it differs from remaining local maxima in that the value of f corresponding to at least one point in the neighborhood of x_1 equals $f(x_1)$. In this respect, x_1 is a **weak maximum**, whereas x_3 and x_6 are **strong maxima**. In general, for \mathbf{h} as defined earlier, \mathbf{X}_0 is a weak maximum if $f(\mathbf{X}_0 + \mathbf{h}) \leq f(\mathbf{X}_0)$ and a strong maximum if $f(\mathbf{X}_0 + \mathbf{h}) < f(\mathbf{X}_0)$.

In Figure 20.1, the first derivative (slope) of f equals zero at all extrema. This property is also satisfied at **inflection** and **saddle** points, such as x_5 . If a point with zero slope (gradient) is not an extremum (maximum or minimum), then it must be an inflection or a saddle point.