UNIVERSIDAD TECNOLÓGICA
FACULTAD REGIONAL ROSARIO

SIMULACIÓN

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

4º AÑO

UTN

2018

Letter's
CENTRO DE COPIADO
MENDOZA 1637 ROSARIO
lettersdyn@hotmail.com
Tel. fijo: 222 1308
Cel.: 155 210731
153093558

Profesor: Ing. Darío Weitz

Yo los tomo y los encierro y los suplicio con el estudio, sabiendo bien que lo que es fácil es estéril por esta misma razón. Y mido la importancia del trabajo en la torsión y el sudor. Y por esto he reunido a los maestros de mis escuelas y les he dicho: "No os equivoquéis. Os he confiado los hijos de los hombres no para pesar más adelante la suma de sus conocimientos, sino para regocijarme de la calidad de su ascensión. Y no me interesa aquel de vuestros discípulos que haya conocido, llevado en litera, mil cimas de montañas y así observado mil paisajes, porque en primer lugar no conocerá uno sólo verdaderamente, y luego, porque mil paisajes no constituyen más que una partícula de polvo en la inmensidad del mundo. Me interesará sólo el que haya ejercido sus músculos en la ascensión de una montaña, aunque sea la única, y así estar capacitado para comprender todos los paisajes por venir y, mejor que el otro, vuestro falso sabio, los mil paisajes que le han enseñado.....

.... Pero yo les repito aún: cuando digo montaña, significo montaña para ti, que te has desgarrado en sus zarzas, saltando sus precipicios, sudado contra sus piedras, cogido sus flores y respirado finalmente a pleno aire en su cumbre....

CIUDADELA

Antoine de Saint-Exupéry

CHAPTER 1

BASIC SIMULATION MODELING

Recommended sections for a first reading: 1.1 through 1.4, 1.7, 1.9

1.1 THE NATURE OF SIMULATION

This is a book about techniques for using computers to imitate, or *simulate*, the operations of various kinds of real-world facilities or processes. The facility or process of interest is usually called a *system*, and in order to study it scientifically we often have to make a set of assumptions about how it works. These assumptions, which usually take the form of mathematical or logical relationships, constitute a *model* that is used to try to gain some understanding of how the corresponding system behaves.

If the relationships that compose the model are simple enough, it may be possible to use mathematical methods (such as algebra, calculus, or probability theory) to obtain *exact* information on questions of interest; this is called an *analytic* solution. However, most real-world systems are too complex to allow realistic models to be evaluated analytically, and these models must be studied by means of simulation. In a *simulation* we use a computer to evaluate a model *numerically*, and data are gathered in order to *estimate* the desired true characteristics of the model.

As an example of the use of simulation, consider a manufacturing firm that is contemplating building a large extension onto one of its plants but is not sure if the potential gain in productivity would justify the construction cost. It certainly would not be cost-effective to build the extension and then remove it later if it does not work out. However, a careful simulation study could shed some light on the question by simulating the operation of the plant as it currently exists and as it *would* be *if* the plant were expanded.

Application areas for simulation are numerous and diverse. Below is a list of some particular kinds of problems for which simulation has been found to be a useful and powerful tool:

- Designing and analyzing manufacturing systems
- Evaluating hardware and software requirements for a computer system
- Evaluating a new military weapons system or tactic
- Determining ordering policies for an inventory system
- Designing communications systems and message protocols for them
- Designing and operating transportation facilities such as freeways, airports, subways, or ports
- Evaluating designs for service organizations such as hospitals, post offices, or fast-food restaurants
- Analyzing financial or economic systems

As a technique, simulation is one of the most widely used in operations research and management science. In a survey of graduates of the Department of Operations Research at Case Western Reserve University (one of the first departments of this type), Rasmussen and George (1978) found that among M.S. graduates, simulation ranked fifth among some fifteen subject areas in terms of its value after graduation (behind what they called "statistical methods," "forecasting," "systems analysis," and "information systems," all of which may arguably be outside the realm of operations research and management science). Among Ph.D. graduates, simulation tied (with linear programming) for second (behind "statistical methods"). Thomas and DaCosta (1979), in a survey of a different type, asked some 137 large firms to indicate which of fourteen techniques they used, and simulation came in second, with 84 percent of the firms responding that they used it (what they termed "statistical analysis" came in first in this survey, with 93 percent). The members of the Operations Research Division of the American Institute of Industrial Engineers were surveyed by Shannon, Long, and Buckles (1980), who reported that simulation ranked second in "familiarity" (just behind linear programming), but first in terms of utility and interest, among some twelve methodologies. Forgionne (1983) and Harpell, Lane, and Mansour (1989) also reported that simulation ranked second in utilization (again behind "statistical analysis" only) among eight tools in a survey of large corporations. All of these

surveys are by now several years old, and we can assume that simulation's value and usage have since increased, due to improvements in computing power and in simulation software, as discussed below.

There have been, however, several impediments to even wider acceptance and usefulness of simulation. First, models used to study large-scale systems tend to be very complex, and writing computer programs to execute them can be an arduous task indeed. This task has been eased in recent years by the development of excellent software products that automatically provide many of the features needed to code a simulation model. A second problem with simulation of complex systems is that a large amount of computer time is often required. However, this difficulty is becoming less severe as the cost of computing continues to fall. Finally, there appears to be an unfortunate impression that simulation is just an exercise in computer programming, albeit a complicated one. Consequently, many simulation "studies" have been composed of heuristic model building, coding, and a single run of the program to obtain "the answer." We fear that this attitude, which neglects the important issue of how a properly coded model should be used to make inferences about the system of interest, has doubtless led to erroneous conclusions being drawn from many simulation studies. These questions of simulation *methodology*, which are largely independent of the software and hardware used, form an integral part of the latter chapters of this book.

In the remainder of this chapter (as well as in Chap. 2) we discuss systems and models in considerably more detail and then show how to write computer programs to simulate systems of varying degrees of complexity.

1.2 SYSTEMS, MODELS, AND SIMULATION

A *system* is defined to be a collection of entities, e.g., people or machines, that act and interact together toward the accomplishment of some logical end. [This definition was proposed by Schmidt and Taylor (1970).] In practice, what is meant by "the system" depends on the objectives of a particular study. The collection of entities that compose a system for one study might be only a subset of the overall system for another. For example, if one wants to study a bank to determine the number of tellers needed to provide adequate service for customers who want just to cash a check or make a savings deposit, the system can be defined to be that portion of the bank consisting of the tellers and the customers waiting in line or being served. If, on the other hand, the loan officer and the safety deposit boxes are to be included, the definition of the system must be expanded in an obvious way. [See also Fishman (1978, p. 3).] We define the *state* of a system to be that collection of variables necessary to describe a system at a particular time, relative to the objectives of a study. In a study of a bank, examples of possible state variables are the number of busy tellers, the number of customers in the bank, and the time of arrival of each customer in the bank.

2

We categorize systems to be of two types, discrete and continuous. A *discrete* system is one for which the state variables change instantaneously at separated points in time. A bank is an example of a discrete system, since state variables—e.g., the number of customers in the bank—change only when a customer arrives or when a customer finishes being served and departs. A *continuous* system is one for which the state variables change continuously with respect to time. An airplane moving through the air is an example of a continuous system, since state variables such as position and velocity can change continuously with respect to time. Few systems in practice are wholly discrete or wholly continuous, but since one type of change predominates for most systems, it will usually be possible to classify a system as being either discrete or continuous.

At some point in the lives of most systems, there is a need to study them to try to gain some insight into the relationships among various components, or to predict performance under some new conditions being considered. Figure 1.1 maps out different ways in which a system might be studied.

- *Experiment with the Actual System vs. Experiment with a Model of the System*: If it is possible (and cost-effective) to alter the system physically and then let it operate under the new conditions, it is probably desirable to do so, for in this case there is no question about whether what we study is

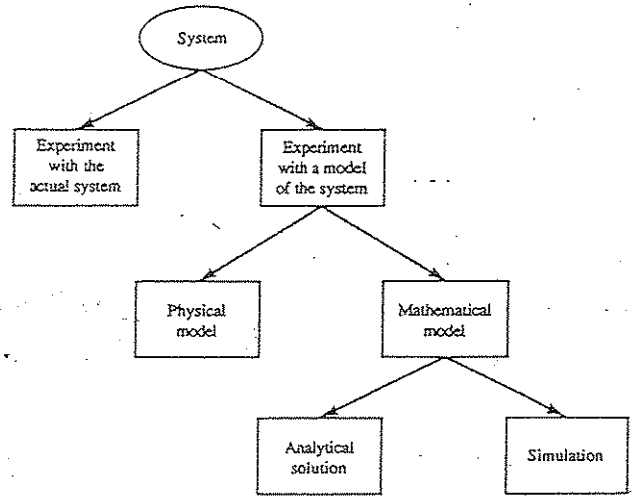


FIGURE 1.1
Ways to study a system.

relevant. However, it is rarely feasible to do this, because such an experiment would often be too costly or too disruptive to the system. For example, a bank may be contemplating reducing the number of tellers to decrease costs, but actually trying this could lead to long customer delays and alienation. More graphically, the "system" might not even exist, but we nevertheless want to study it in its various proposed alternative configurations to see how it should be built in the first place; examples of this situation might be modern flexible manufacturing facilities, or strategic nuclear weapons systems. For these reasons, it is usually necessary to build a *model* as a representation of the system and study it as a surrogate for the actual system. When using a model, there is always the question of whether it accurately reflects the system for the purposes of the decisions to be made; this question of model *validity* is taken up in detail in Chap. 5.

- *Physical Model vs. Mathematical Model*: To most people, the word "model" evokes images of clay cars in wind tunnels, cockpits disconnected from their airplanes to be used in pilot training, or miniature supertankers scurrying about in a swimming pool. These are examples of *physical* models (also called *iconic* models), and are not typical of the kinds of models that are usually of interest in operations research and systems analysis. Occasionally, however, it has been found useful to build physical models to study engineering or management systems; examples include tabletop scale models of material-handling systems, and in at least one case a full-scale physical model of a fast-food restaurant inside a warehouse, complete with full-scale, real (and presumably hungry) humans [see Swart and Donno (1981)]. But the vast majority of models built for such purposes are *mathematical*, representing a system in terms of logical and quantitative relationships that are then manipulated and changed to see how the model reacts, and thus how the system *would* react—*if* the mathematical model is a valid one. Perhaps the simplest example of a mathematical model is the familiar relation $d = rt$, where r is the rate of travel, t is the time spent traveling, and d is the distance traveled. This might provide a valid model in one instance (e.g., a space probe to another planet after it has attained its flight velocity) but a very poor model for other purposes (e.g., rush-hour commuting on congested urban freeways).

- *Analytical Solution vs. Simulation*: Once we have built a mathematical model, it must then be examined to see how it can be used to answer the questions of interest about the system it is supposed to represent. If the model is simple enough, it may be possible to work with its relationships and quantities to get an exact, *analytical* solution. In the $d = rt$ example, if we know the distance to be traveled and the velocity, then we can work with the model to get $t = d/r$ as the time that will be required. This is a very simple, closed-form solution obtainable with just paper and pencil, but some analytical solutions can become extraordinarily complex, requiring vast computing

resources; inverting a large nonsparse matrix is a well-known example of a situation in which there is an analytical formula known in principle, but obtaining it numerically in a given instance is far from trivial. If an analytical solution to a mathematical model is available and is computationally efficient, it is usually desirable to study the model in this way rather than via a simulation. However, many systems are highly complex, so that valid mathematical models of them are themselves complex, precluding any possibility of an analytical solution. In this case, the model must be studied by means of *simulation*, i.e., numerically exercising the model for the inputs in question to see how they affect the output measures of performance.

While there may be an element of truth to pejorative old saws such as "method of last resort" sometimes used to describe simulation, the fact is that we are very quickly led to simulation in many situations, due to the sheer complexity of the systems of interest and of the models necessary to represent them in a valid way.

Given, then, that we have a mathematical model to be studied by means of simulation (henceforth referred to as a *simulation model*), we must then look for particular tools to do this. It is useful for this purpose to classify simulation models along three different dimensions:

- *Static vs. Dynamic Simulation Models*: A *static* simulation model is a representation of a system at a particular time, or one that may be used to represent a system in which time simply plays no role; examples of static simulations are Monte Carlo models, discussed in Sec. 1.8.3. On the other hand, a *dynamic* simulation model represents a system as it evolves over time, such as a conveyor system in a factory.
- *Deterministic vs. Stochastic Simulation Models*: If a simulation model does not contain any probabilistic (i.e., random) components, it is called *deterministic*; a complicated (and analytically intractable) system of differential equations describing a chemical reaction might be such a model. In deterministic models, the output is "determined" once the set of input quantities and relationships in the model have been specified, even though it might take a lot of computer time to evaluate what it is. Many systems, however, must be modeled as having at least some random input components, and these give rise to *stochastic* simulation models. (For an example of the danger of ignoring randomness in modeling a system, see Sec. 4.7.) Most queueing and inventory systems are modeled stochastically. Stochastic simulation models produce output that is itself random, and must therefore be treated as only an estimate of the true characteristics of the model; this is one of the main disadvantages of simulation (see Sec. 1.9) and is dealt with in Chaps. 9 through 12 of this book.
- *Continuous vs. Discrete Simulation Models*: Loosely speaking, we define *discrete* and *continuous* simulation models analogously to the way discrete

and continuous systems were defined above. More precise definitions of discrete (event) simulation and continuous simulation are given in Secs. 1.3 and 1.8, respectively. It should be mentioned that a discrete model is not always used to model a discrete system and vice versa. The decision whether to use a discrete or a continuous model for a particular system depends on the specific objectives of the study. For example, a model of traffic flow on a freeway would be discrete if the characteristics and movement of individual cars are important. Alternatively, if the cars can be treated "in the aggregate," the flow of traffic can be described by differential equations in a continuous model. More discussion on this issue can be found in Sec. 5.2, and in particular in Example 5.1.

The simulation models we consider in the remainder of this book, except for those in Sec. 1.8, will be discrete, dynamic, and stochastic and will henceforth be called *discrete-event simulation models*. (Since deterministic models are a special case of stochastic models, the restriction to stochastic models involves no loss of generality.)

1.3 DISCRETE-EVENT SIMULATION

Discrete-event simulation concerns the modeling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time. (In more mathematical terms, we might say that the system can change at only a *countable* number of points in time.) These points in time are the ones at which an event occurs, where an *event* is defined as an instantaneous occurrence that may change the state of the system. Although discrete-event simulation could conceptually be done by hand calculations, the amount of data that must be stored and manipulated for most real-world systems dictates that discrete-event simulations be done on a digital computer. (In Sec. 1.4.2 we carry out a small hand simulation, merely to illustrate the logic involved.)

Example 1.1. Consider a service facility with a single server—e.g., a one-operator barbershop or an information desk at an airport—for which we would like to estimate the (expected) average delay in queue (line) of arriving customers, where the delay in queue of a customer is the length of the time interval from the instant of his arrival at the facility to the instant he begins being served. For the objective of estimating the average delay of a customer, the state variables for a discrete-event simulation model of the facility would be the status of the server, i.e., either idle or busy, the number of customers waiting in queue to be served (if any), and the time of arrival of each person waiting in queue. The status of the server is needed to determine, upon a customer's arrival, whether the customer can be served immediately or must join the end of the queue. When the server completes serving a customer, the number of customers in the queue is used to determine whether the server will become idle or begin serving the first customer in the queue. The time of arrival of a customer is needed to compute his delay in

4

queue, which is the time he begins being served (which will be known) minus his time of arrival. There are two types of events for this system: the arrival of a customer and the completion of service for a customer, which results in the customer's departure. An arrival is an event since it causes the (state variable) server status to change from idle to busy or the (state variable) number of customers in the queue to increase by 1. Correspondingly, a departure is an event because it causes the server status to change from busy to idle or the number of customers in the queue to decrease by 1. We show in detail how to build a discrete-event simulation model of this single-server queueing system in Sec. 1.4.

In the above example both types of events actually changed the state of the system, but in some discrete-event simulation models events are used for purposes that do not actually effect such a change. For example, an event might be used to schedule the end of a simulation run at a particular time (see Sec. 1.4.8) or to schedule a decision about a system's operation at a particular time (see Sec. 1.5) and might not actually result in a change in the state of the system. This is why we originally said that an event *may* change the state of a system.

1.3.1 Time-Advance Mechanisms

Because of the dynamic nature of discrete-event simulation models, we must keep track of the current value of simulated time as the simulation proceeds, and we also need a mechanism to advance simulated time from one value to another. We call the variable in a simulation model that gives the current value of simulated time the *simulation clock*. The unit of time for the simulation clock is never stated explicitly when a model is written in a general-purpose language such as FORTRAN, Pascal, or C, and it is assumed to be in the same units as the input parameters. Also, there is generally no relationship between simulated time and the time needed to run a simulation on the computer.

Historically, two principal approaches have been suggested for advancing the simulation clock: *next-event time advance* and *fixed-increment time advance*. Since the first approach is used by all major simulation languages and by most people coding their model in a general-purpose language, and since the second is a special case of the first, we shall use the next-event time-advance approach for all discrete-event simulation models discussed in this book. A brief discussion of fixed-increment time advance is given in App. 1A (at the end of this chapter).

With the next-event time-advance approach, the simulation clock is initialized to zero and the times of occurrence of future events are determined. The simulation clock is then advanced to the time of occurrence of the *most imminent* (first) of these future events, at which point the state of the system is updated to account for the fact that an event has occurred, and our knowledge of the times of occurrence of future events is also updated. Then the simulation clock is advanced to the time of the (new) most imminent event, the state of

the system is updated, and future event times are determined, etc. This process of advancing the simulation clock from one event time to another is continued until eventually some prespecified stopping condition is satisfied. Since all state changes occur only at event times for a discrete-event simulation model, periods of inactivity are skipped over by jumping the clock from event time to event time. (Fixed-increment time advance does not skip over these inactive periods, which can eat up a lot of computer time; see App. 1A.) It should be noted that the successive jumps of the simulation clock are generally variable (or unequal) in size.

Example 1.2 We now illustrate in detail the next-event time-advance approach for the single-server queueing system of Example 1.1. We need the following notation:

t_i = time of arrival of the ith customer ($t_0 = 0$) Tiempo
$A_i = t_i - t_{i-1}$ = interarrival time between $(i-1)$st and ith arrivals of customers
S_i = time that server actually spends serving ith customer (exclusive of customer's delay in queue)
D_i = delay in queue of ith customer
$c_i = t_i + D_i + S_i$ = time that ith customer completes service and departs
e_i = time of occurrence of ith event of any type (ith value the simulation clock takes on, excluding the value $e_0 = 0$)

Each of these defined quantities will generally be a random variable. Assume that the probability distributions of the interarrival times $A_1, A_2, \ldots$ and the service times $S_1, S_2, \ldots$ are known and have cumulative distribution functions (see Sec. 4.2) denoted by F_A and F_S, respectively. (In general, F_A and F_S would be determined by collecting data from the system of interest and then fitting distributions to these data using the techniques of Chap. 6.) At time $e_0 = 0$ the status of the server is idle, and the time t_1 of the first arrival is determined by generating A_1 from F_A (techniques for generating random observations from a specified distribution are discussed in Chap. 8) and adding it to 0. The simulation clock is then advanced from e_0 to the time of the next (first) event, $e_1 = t_1$. (See Fig. 1.2, where the curved arrows represent advancing the simulation clock.) Since the customer arriving at time t_1 finds the server idle, she immediately enters service and has a delay in queue of $D_1 = 0$ and the status of the server is changed from idle to busy. The time, c_1, when the arriving customer will complete service is computed by generating S_1 from F_S and adding it to t_1. Finally, the time of the second arrival, t_2, is computed as $t_2 = t_1 + A_2$, where A_2 is generated from F_A. If $t_2 < c_1$, as depicted in Fig. 1.2, the simulation clock is advanced from e_1 to the time of the next event, $e_2 = t_2$. (If c_1 were less than t_2, the clock would be advanced from e_1 to c_1.) Since the customer arriving at time t_2 finds the server already busy, the number of customers in the queue is increased from 0 to 1 and the time of arrival of this customer is recorded; however, his service time S_2 is not generated at this time. Also, the time of the third arrival, t_3, is computed as $t_3 = t_2 + A_3$. If $c_1 < t_3$, as depicted in the figure, the simulation clock is advanced from e_2 to the time of the next event, $e_3 = c_1$, where the customer completing service departs, the customer in the queue (i.e., the one who arrived at time t_2)

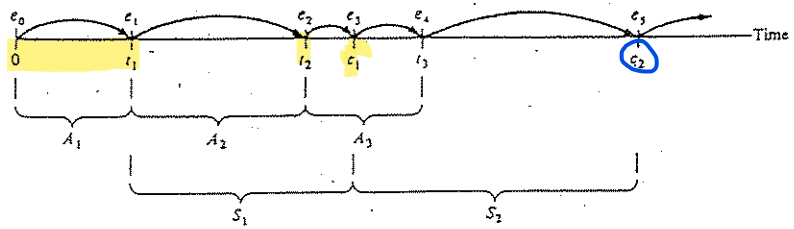


FIGURE 1.2
The next-event time-advance approach illustrated for the single-server queueing system.

begins service and his delay in queue and service-completion time are computed as $D_2 = c_1 - t_2$ and $c_2 = c_1 + S_2$ (S_2 is now generated from F_S), and the number of customers in the queue is decreased from 1 to 0. If $t_3 < c_2$, the simulation clock is advanced from e_3 to the time of the next event, $e_4 = t_3$, etc. The simulation might eventually be terminated when, say, the number of customers whose delays have been observed reaches some specified value.

1.3.2 Components and Organization of a Discrete-Event Simulation Model

Although simulation has been applied to a great diversity of real-world systems, discrete-event simulation models all share a number of common components and there is a logical organization for these components that promotes the coding, debugging, and future changing of a simulation model's computer program. In particular, the following components will be found in most discrete-event simulation models using the next-event time-advance approach:

System state: The collection of state variables necessary to describe the system at a particular time.

Simulation clock: A variable giving the current value of simulated time.

Event list: A list containing the next time when each type of event will occur.

Statistical counters: Variables used for storing statistical information about system performance.

Initialization routine: A subprogram to initialize the simulation model at time zero.

Timing routine: A subprogram that determines the next event from the event list and then advances the simulation clock to the time when that event is to occur.

Event routine: A subprogram that updates the system state when a particular type of event occurs (there is one event routine for each event type).

Library routines: A set of subprograms used to generate random observations from probability distributions that were determined as part of the simulation model.

Report generator: A subprogram that computes estimates (from the statistical counters) of the desired measures of performance and produces a report when the simulation ends.

Main program: A subprogram that invokes the timing routine to determine the next event and then transfers control to the corresponding event routine to update the system state appropriately. The main program may also check for termination and invoke the report generator when the simulation is over.

The logical relationships (flow of control) among these components is shown in Fig. 1.3. The simulation begins at time 0 with the main program invoking the initialization routine, where the simulation clock is set to zero, the system state and the statistical counters are initialized, and the event list is initialized. After control has been returned to the main program, it invokes the timing routine to determine which type of event is most imminent. If an event of type i is the next to occur, the simulation clock is advanced to the time that event type i will occur and control is returned to the main program. Then the main program invokes event routine i, where typically three types of activities occur: (1) the system state is updated to account for the fact that an event of type i has occurred; (2) information about system performance is gathered by updating the statistical counters; and (3) the times of occurrence of future events are generated and this information is added to the event list. Often it is necessary to generate random observations from probability distributions in order to determine these future event times; we will refer to such a generated observation as a *random variate*. After all processing has been completed, either in event routine i or in the main program, a check is typically made to determine (relative to some stopping condition) if the simulation should now be terminated. If it is time to terminate the simulation, the report generator is invoked from the main program to compute estimates (from the statistical counters) of the desired measures of performance and to produce a report. If it is not time for termination, control is passed back to the main program and the main program–timing routine–main program–event routine–termination check cycle is repeated until the stopping condition is eventually satisfied.

Before concluding this section, a few additional words about the system state may be in order. As mentioned in Sec. 1.2, a system is a well-defined collection of *entities*. Entities are characterized by data values called *attributes*, and these attributes are part of the system state for a discrete-event simulation model. Furthermore, entities with some common property are often grouped together in *lists* (or *files* or *sets*). For each entity there is a *record* in the list consisting of the entity's attributes, and the order in which the records are placed in the list depends on some specified rule. (See Chap. 2 for a discussion of efficient approaches for storing lists of records.) For the single-server queueing facility of Examples 1.1 and 1.2, the entities are the server and the

6

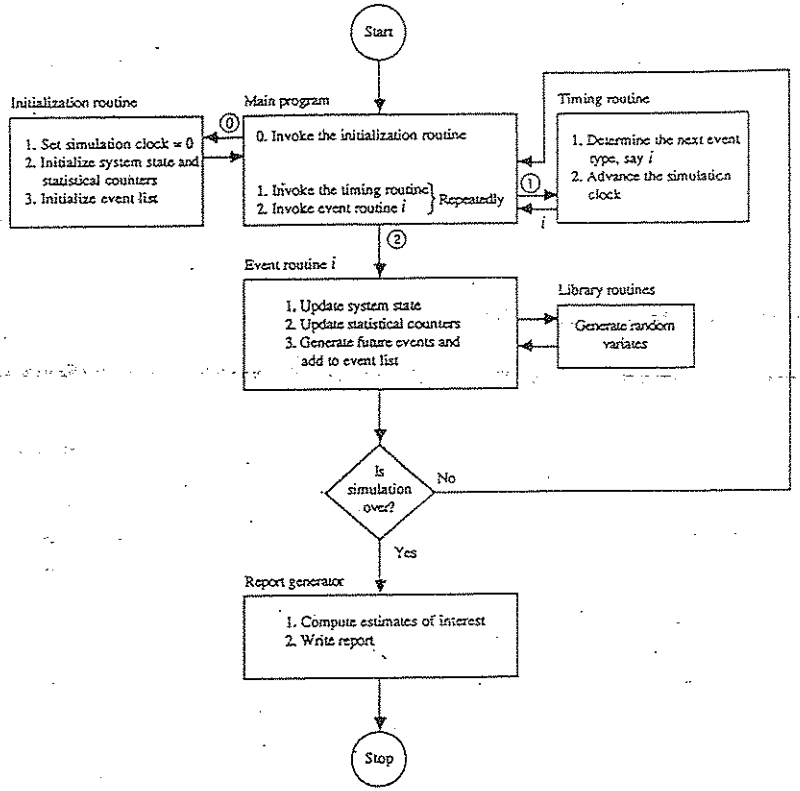


FIGURE 1.3
Flow of control for the next-event time-advance approach.

customers in the facility. The server has the attribute "server status" (busy or idle), and the customers waiting in queue have the attribute "time of arrival." (The number of customers in the queue might also be considered an attribute of the server.) Furthermore, as we shall see in Sec. 1.4, these customers in queue will be grouped together in a list.

The organization and action of a discrete-event simulation program using the next-event time-advance mechanism as depicted above is fairly typical when coding such simulations in a general-purpose programming language such as FORTRAN, Pascal, or C; it is called the *event-scheduling approach* to simulation modeling, since the times of future events are explicitly coded into the model and are scheduled to occur in the simulated future. It should be

mentioned here that there is an alternative approach to simulation modeling, called the *process approach*, that instead views the simulation in terms of the individual entities involved, and the code written describes the "experience" of a "typical" entity as it "flows" through the system; coding simulations modeled from the process point of view usually requires the use of special-purpose simulation software, as discussed in Chap. 3. Even when taking the process approach, however, the simulation is actually executed behind the scenes in the event-scheduling logic as described above.

1.4 SIMULATION OF A SINGLE-SERVER QUEUEING SYSTEM

This section shows in detail how to simulate a single-server queueing system such as a one-operator barbershop. Although this system seems very simple compared with those usually of real interest, how it is simulated is actually quite representative of the operation of simulations of great complexity.

In Sec. 1.4.1 we describe the system of interest and state our objectives more precisely. We explain intuitively how to simulate this system in Sec. 1.4.2 by showing a "snapshot" of the simulated system just after each event occurs. Section 1.4.3 describes the language-independent organization and logic of the FORTRAN, Pascal, and C codes given in Secs. 1.4.4, 1.4.5, and 1.4.6. The simulation's results are discussed in Sec. 1.4.7, and Sec. 1.4.8 alters the stopping rule to another common way to end simulations. Finally, Sec. 1.4.9 briefly describes a technique for identifying and simplifying the event and variable structure of a simulation.

1.4.1 Problem Statement

Consider a single-server queueing system (see Fig. 1.4) for which the interarrival times $A_1, A_2, \ldots$ are independent, identically distributed (IID) random variables. ("Identically distributed" means that the interarrival times have the same probability distribution.) A customer who arrives and finds the server idle enters service immediately, and the service times $S_1, S_2, \ldots$ of the successive customers are IID random variables that are independent of the interarrival times. A customer who arrives and finds the server busy joins the end of a single queue. Upon completing service for a customer, the server chooses a customer from the queue (if any) in a first-in, first-out (FIFO) manner. (For a discussion of other queue disciplines and queueing systems in general, see App. 1B.)

The simulation will begin in the "empty-and-idle" state; i.e., no customers are present and the server is idle. At time 0, we will begin waiting for the arrival of the first customer, which will occur after the first interarrival time, A_1, rather than at time 0 (which would be a possibly valid, but different, modeling assumption). We wish to simulate this system until a fixed number

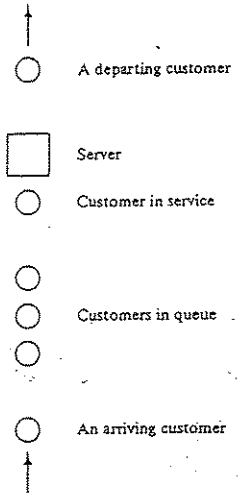


FIGURE 1.4
A single-server queueing system.

(n) of customers have completed their delays in queue; i.e., the simulation will stop when the nth customer enters service. Note that the *time* the simulation ends is thus a random variable, depending on the observed values for the interarrival and service-time random variables.

To measure the performance of this system, we will look at estimates of three quantities. First, we will estimate the expected average delay in queue of the n customers completing their delays during the simulation; we denote this quantity by $d(n)$. The word "expected" in the definition of $d(n)$ means this: On a given run of the simulation (or, for that matter, on a given run of the actual system the simulation model represents), the actual average delay observed of the n customers depends on the interarrival and service-time random variable observations that happen to have been obtained. On another run of the simulation (or on a different day for the real system) there would probably be arrivals at different times, and the service times required would also be different; this would give rise to a different value for the average of the n delays. Thus, the average delay on a given run of the simulation is properly regarded as a random variable itself. What we want to estimate, $d(n)$, is the *expected value* of this random variable. One interpretation of this is that $d(n)$ is the average of a large (actually, infinite) number of n-customer average delays. From a single run of the simulation resulting in customer delays $D_1, D_2, \ldots, D_n$, an obvious estimator of $d(n)$ is

$$\hat{d}(n) = \frac{\sum_{i=1}^{n} D_i}{n}$$

Demora corrida i

cant de corridas

which is just the average of the n D_i's that were observed in the simulation [so that $\hat{d}(n)$ could also be denoted by $\bar{D}(n)$]. (Throughout this book, a hat (^) above a symbol denotes an estimator.) It is important to note that by "delay" we do not exclude the possibility that a customer could have a delay of zero in the case of an arrival finding the system empty and idle (with this model, we know for sure that $D_1 = 0$); delays with a value of zero *are* counted in the average, since if many delays were zero this would represent a system providing very good service, and our output measure should reflect this. One reason for taking the average of the D_i's, as opposed to just looking at them individually, is that they will not have the same distribution (e.g., $D_1 = 0$, but D_2 could be positive), and the average gives us a single composite measure of all the customers' delays; in this sense, this is not the usual "average" taken in basic statistics, as the individual terms are not random observations from the same distribution. Note also that by itself, $\hat{d}(n)$ is an estimator based on a "sample" (here, a set of *complete* simulation runs) of size *1*, since we are making only a single simulation run. From elementary statistics, we know that a sample of size 1 is not worth much; we return to this issue in Chaps. 9 through 12.

While an estimate of $d(n)$ gives information about system performance from the customers' point of view, the management of such a system may want different information; indeed, since most real simulations are quite complex and may be costly to run, we usually collect many output measures of performance, describing different aspects of system behavior. One such measure for our simple model here is the expected average number of customers in the queue (but not being served), denoted by $q(n)$, where the n is necessary in the notation to indicate that this average is taken over the time period needed to observe the n delays defining our stopping rule. This is a different kind of "average" than the average delay in queue, because it is taken over (continuous) time, rather than over customers (being discrete). Thus, we need to define what is meant by this *time*-average number of customers in queue. To do this, let $Q(t)$ denote the number of customers in queue at time t, for any real number $t \geq 0$, and let $T(n)$ be the time required to observe our n delays in queue. Then for any time t between 0 and $T(n)$, $Q(t)$ is a nonnegative integer. Further, if we let p_i be the expected *proportion* (which will be between 0 and 1) of the time that $Q(t)$ is equal to i, then a reasonable definition of $q(n)$ would be

$$q(n) = \sum_{i=0}^{\infty} i p_i$$

Proporcion de que haya i clientes en la cola

Thus, $q(n)$ is a weighted average of the possible values i for the queue length $Q(t)$, with the weights being the expected proportion of time the queue spends at each of its possible lengths. To estimate $q(n)$ from a simulation, we simply replace the p_i's with estimates of them, and get

$$\hat{q}(n) = \sum_{i=0}^{\infty} i \hat{p}_i \qquad (1.1)$$

Tamanio promedio de cola

where $\hat{p}_i$ is the *observed* (rather than expected) proportion of the time *during the simulation* that there were i customers in the queue. Computationally, however, it is easier to rewrite $\hat{q}(n)$ using some geometric considerations. If we let T_i be the *total* time during the simulation that the queue is of length i, then $T(n) = T_0 + T_1 + T_2 + \cdots$ and $\hat{p}_i = T_i/T(n)$, so that we can rewrite Eq. (1.1) above as

$$\hat{q}(n) = \frac{\sum_{i=0}^{\infty} iT_i}{T(n)} \qquad (1.2)$$

Q(t)

Figure 1.5 illustrates a possible time path, or *realization*, of $Q(t)$ for this system in the case of $n = 6$; ignore the shading for now. Arrivals occur at times 0.4, 1.6, 2.1, 3.8, 4.0, 5.6, 5.8, and 7.2. Departures (service completions) occur at times 2.4, 3.1, 3.3, 4.9, and 8.6, and the simulation ends at time $T(6) = 8.6$. Remember in looking at Fig. 1.5 that $Q(t)$ does not count the customer in service (if any), so between times 0.4 and 1.6 there is one customer in the system being served, even though the queue is empty [$Q(t) = 0$]; the same is true between times 3.1 and 3.3, between times 3.8 and 4.0, and between times 4.9 and 5.6. Between times 3.3 and 3.8, however, the system is empty of customers and the server is idle, as is obviously the case between times 0 and 0.4. To compute $\hat{q}(n)$, we must first compute the T_i's, which can be read off Fig. 1.5 as the (sometimes separated) intervals over which $Q(t)$ is equal to 0, 1,

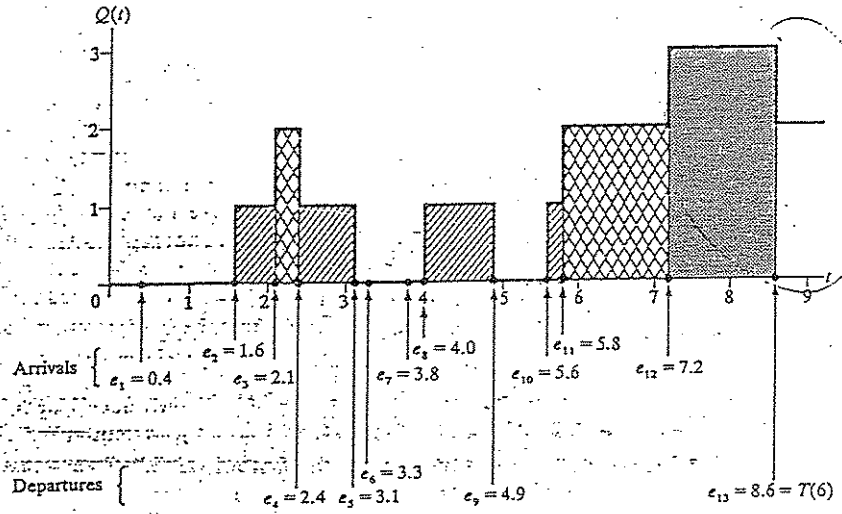


FIGURE 1.5
$Q(t)$, arrival times, and departure times for a realization of a single-server queueing system.

Tiempo en que se tuvo i cantidad de personas en cola Tsub(i)

2, and so on:

$$T_0 = (1.6 - 0.0) + (4.0 - 3.1) + (5.6 - 4.9) = 3.2$$
$$T_1 = (2.1 - 1.6) + (3.1 - 2.4) + (4.9 - 4.0) + (5.8 - 5.6) = 2.3$$
$$T_2 = (2.4 - 2.1) + (7.2 - 5.8) = 1.7$$
$$T_3 = (8.6 - 7.2) = 1.4$$

• Sumar Tn

($T_i = 0$ for $i \geq 4$, since the queue never grew to those lengths in this realization.) The numerator in Eq. (1.2) is thus

$$\sum_{i=0}^{\infty} iT_i = (0 \times 3.2) + (1 \times 2.3) + (2 \times 1.7) + (3 \times 1.4) = 9.9 \qquad (1.3)$$

and so our estimate of the time-average number in queue from this particular simulation run is $\hat{q}(6) = 9.9/8.6 = 1.15$. Now, note that each of the nonzero terms on the right-hand side of Eq. (1.3) corresponds to one of the shaded areas in Fig. 1.5: 1×2.3 is the diagonally shaded area (in four pieces), 2×1.7 is the cross-hatched area (in two pieces), and 3×1.4 is the screened area (in a single piece). In other words, the summation in the numerator of Eq. (1.2) is just the *area under the $Q(t)$ curve between the beginning and the end of the simulation*. Remembering that "area under a curve" is an integral, we can thus write

$$\sum_{i=0}^{\infty} iT_i = \int_0^{T(n)} Q(t)\,dt$$

and the estimator of $q(n)$ can then be expressed as

$$\hat{q}(n) = \frac{\int_0^{T(n)} Q(t)\,dt}{T(n)} \qquad (1.4)$$

While Eqs. (1.4) and (1.2) are equivalent expressions for $\hat{q}(n)$, Eq. (1.4) is preferable since the integral in this equation can be accumulated as simple areas of rectangles as the simulation progresses through time. It is less convenient to carry out the computations to get the summation in Eq. (1.2) explicitly. Moreover, the appearance of Eq. (1.4) suggests a continuous average of $Q(t)$, since in a rough sense, an integral can be regarded as a continuous summation.

The third and final output measure of performance for this system is a measure of how busy the server is. The expected *utilization* of the server is the expected proportion of time during the simulation [from time 0 to time $T(n)$] that the server is busy (i.e., not idle), and is thus a number between 0 and 1; denote it by $u(n)$. From a single simulation, then, our estimate of $u(n)$ is $\hat{u}(n)$ = the *observed* proportion of time during the simulation that the server is busy. Now $\hat{u}(n)$ could be computed directly from the simulation by noting the times at which the server changes status (idle to busy or vice versa) and then

doing the appropriate subtractions and division. However, it is easier to look at this quantity as a continuous-time average, similar to the average queue length, by defining the "busy function"

$$B(t) = \begin{cases} 1 \text{ if the server is busy at time } t \\ 0 \text{ if the server is idle at time } t \end{cases}$$

and so $\hat{u}(n)$ could be expressed as the proportion of time that $B(t)$ is equal to 1. Figure 1.6 plots $B(t)$ for the same simulation realization as used in Fig. 1.5 for $Q(t)$. In this case, we get

$$\hat{u}(n) = \frac{(3.3 - 0.4) + (8.6 - 3.8)}{8.6} = \frac{7.7}{8.6} = 0.90 \tag{1.5}$$

indicating that the server was busy about 90 percent of the time during this simulation. Again, however, the numerator in Eq. (1.5) can be viewed as the area under the $B(t)$ function over the course of the simulation, since the height of $B(t)$ is always either 0 or 1. Thus,

$$\hat{u}(n) = \frac{\int_0^{T(n)} B(t)\, dt}{T(n)} \tag{1.6}$$

and we see again that $\hat{u}(n)$ is the continuous average of the $B(t)$ function, corresponding to our notion of utilization. As was the case for $\hat{q}(n)$, the reason for writing $\hat{u}(n)$ in the integral form of Eq. (1.6) is that computationally, as the simulation progresses, the integral of $B(t)$ can easily be accumulated by adding up areas of rectangles. For many simulations involving "servers" of some sort, utilization statistics are quite informative in identifying bottlenecks (utilizations

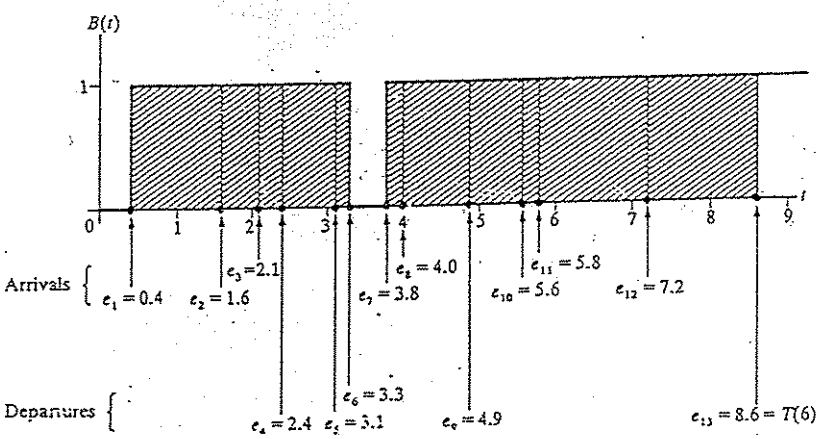


FIGURE 1.6
$B(t)$, arrival times, and departure times for a realization of a single-server queueing system (same realization as in Fig. 1.5).

near 100 percent, coupled with heavy congestion measures for the queue leading in) or excess capacity (low utilizations); this is particularly true if the "servers" are expensive items such as robots in a manufacturing system or large mainframe computers in a data-processing operation.

To recap, the three measures of performance are: the average delay in queue $\hat{d}(n)$, the time-average number of customers in queue $\hat{q}(n)$, and the proportion of time the server is busy $\hat{u}(n)$. The average delay in queue is an example of a *discrete-time statistic*, since it is defined relative to the collection of random variables $\{D_i\}$ that have a discrete "time" index, $i = 1, 2, \ldots$. The time-average number in queue and the proportion of time the server is busy are examples of *continuous-time statistics*, since they are defined on the collection of random variables $\{Q(t)\}$ and $\{B(t)\}$, respectively, each of which is indexed on the continuous time parameter $t \in [0, \infty)$. (The symbol $\in$ means "contained in." Thus, in this case, t can be any nonnegative real number.) Both discrete-time and continuous-time statistics are common in simulation, and they furthermore can be other than averages. For example, we might be interested in the *maximum* of all the delays in queue observed (a discrete-time statistic), or the *proportion* of time during the simulation that the queue contained at least five customers (a continuous-time statistic).

The events for this system are the arrival of a customer and the departure of a customer (after a service completion); the state variables necessary to estimate $d(n)$, $q(n)$, and $u(n)$ are the status of the server (0 for idle and 1 for busy), the number of customers in the queue, the time of arrival of each customer currently in the queue (represented as a list), and the time of the last (i.e., most recent) event. The time of the last event, defined to be e_{i-1} if $e_{i-1} \le t < e_i$ (where t is the current time in the simulation), is needed to compute the width of the rectangles for the area accumulations in the estimates of $q(n)$ and $u(n)$.

1.4.2 Intuitive Explanation

We begin our explanation of how to simulate a single-server queueing system by showing how its simulation model would be represented inside the computer at time $e_0 = 0$ and the times $e_1, e_2, \ldots, e_{13}$ at which the 13 successive events occur that are needed to observe the desired number, $n = 6$, of delays in queue. For expository convenience, we assume that the interarrival and service times of customers are

$$A_1 = 0.4,\ A_2 = 1.2,\ A_3 = 0.5,\ A_4 = 1.7,\ A_5 = 0.2,\ A_6 = 1.6,\ A_7 = 0.2,\ A_8 = 1.4,\ A_9 = 1.9, \ldots$$

$$S_1 = 2.0,\ S_2 = 0.7,\ S_3 = 0.2,\ S_4 = 1.1,\ S_5 = 3.7,\ S_6 = 0.6, \ldots$$

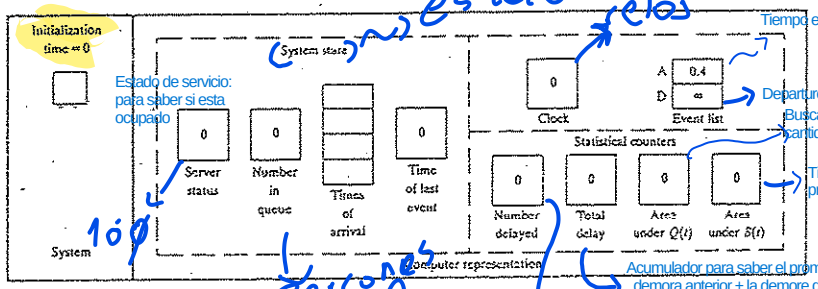
Thus, between time 0 and the time of the first arrival there is 0.4 time unit, between the arrivals of the first and second customers there are 1.2 time units, etc., and the service time required for the first customer is 2.0 time units, etc. Note that it is not necessary to declare what the time units are (minutes, hours, etc.), but only to be sure that all time quantities are expressed in the *same*

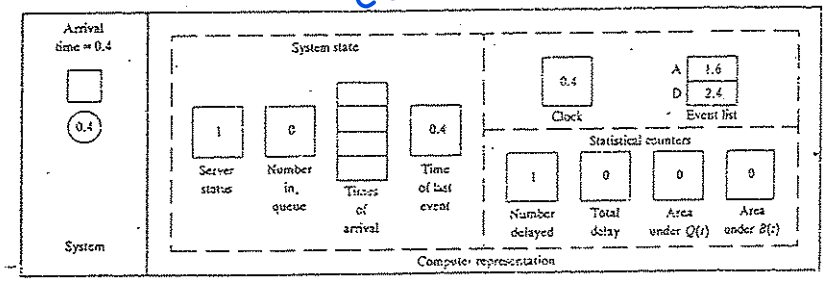
10

units. In an actual simulation (see Secs. 1.4.4 through 1.4.6), the A_i's and the S_i's would be generated from their corresponding probability distributions, as needed, during the course of the simulation. The numerical values for the A_i's and the S_i's given above have been artificially chosen so as to generate the same simulation realization as depicted in Figs. 1.5 and 1.6 illustrating the $Q(t)$ and $B(t)$ processes.

Figure 1.7 gives a snapshot of the system itself and of a computer representation of the system at each of the times $e_0 = 0$, $e_1 = 0.4, \ldots, e_{13} = 8.6$. In the "system" pictures, the square represents the server, and circles represent customers; the numbers inside the customer circles are the times of their arrivals. In the "computer representation" pictures, the values of the variables shown are *after* all processing has been completed at that event. Our discussion will focus on how the computer representation changes at the event times.

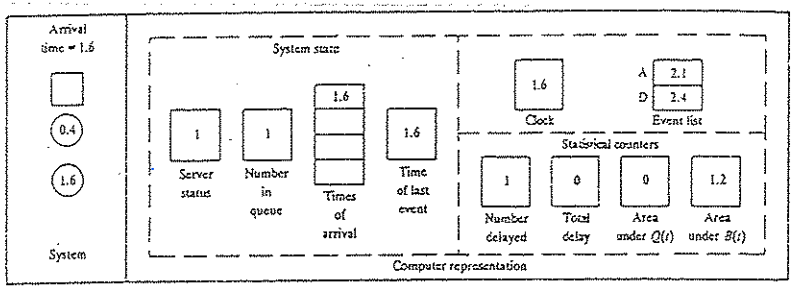
$t = 0$: *Initialization.* The simulation begins with the main program invoking the initialization routine. Our modeling assumption was that





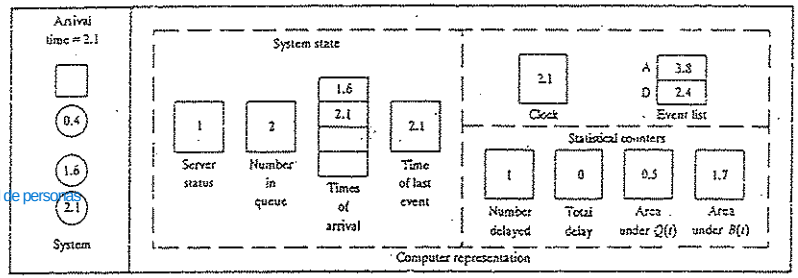
(b)

FIGURE 1.7
Snapshots of the system and of its computer representation at time 0 and at each of the thirteen succeeding event times.



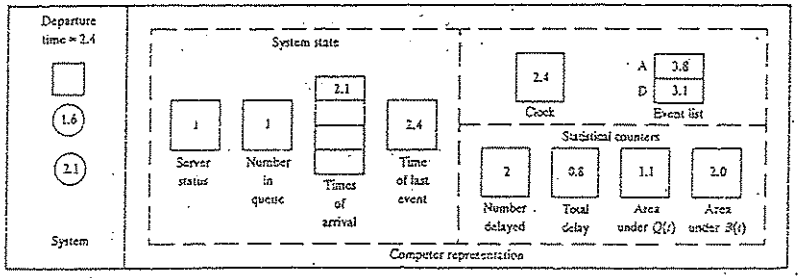
(c)

$Q(t) + \text{Constante cola} \times (t_e - t_{e-1})$



(d)

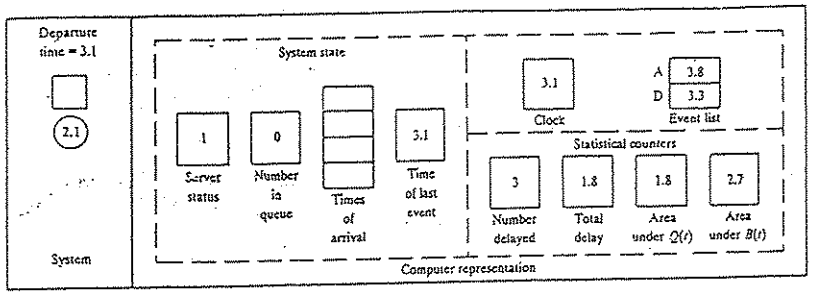
Pasaron 0,3 y x debe dar en cola $\Rightarrow Q(t) = Q(t)_{-1} + 0,6$



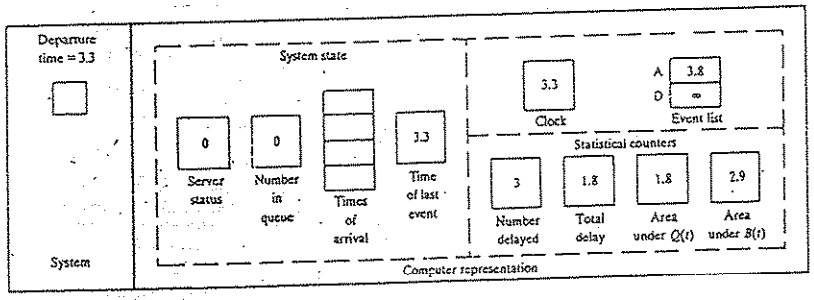
(e)

FIGURE 1.7
(*Continued.*)

11

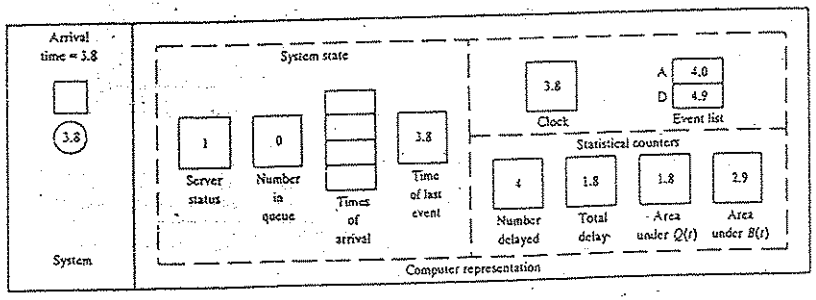


(f)



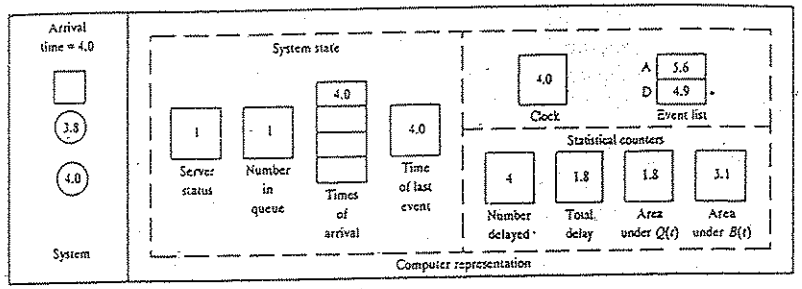
(g)

$d(n) = \frac{B(t)}{\text{Clock}}$ = tiempo promedio que estuvo ocupado el server

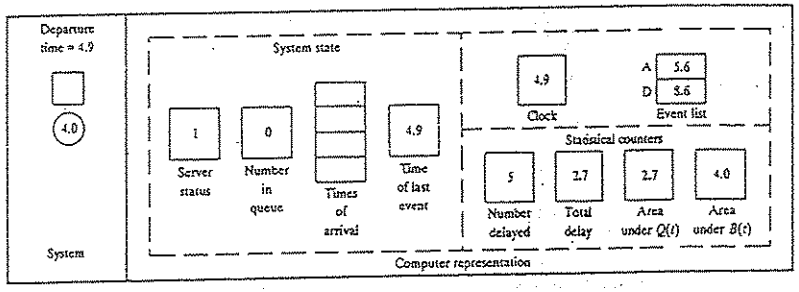


(h)

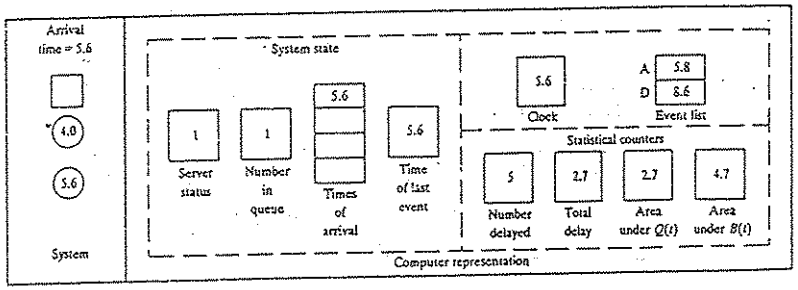
FIGURE 1.7
(*Continued.*)



(i)



(j)



(k)

FIGURE 1.7
(*Continued.*)

Frecuencias de arribo y servicio

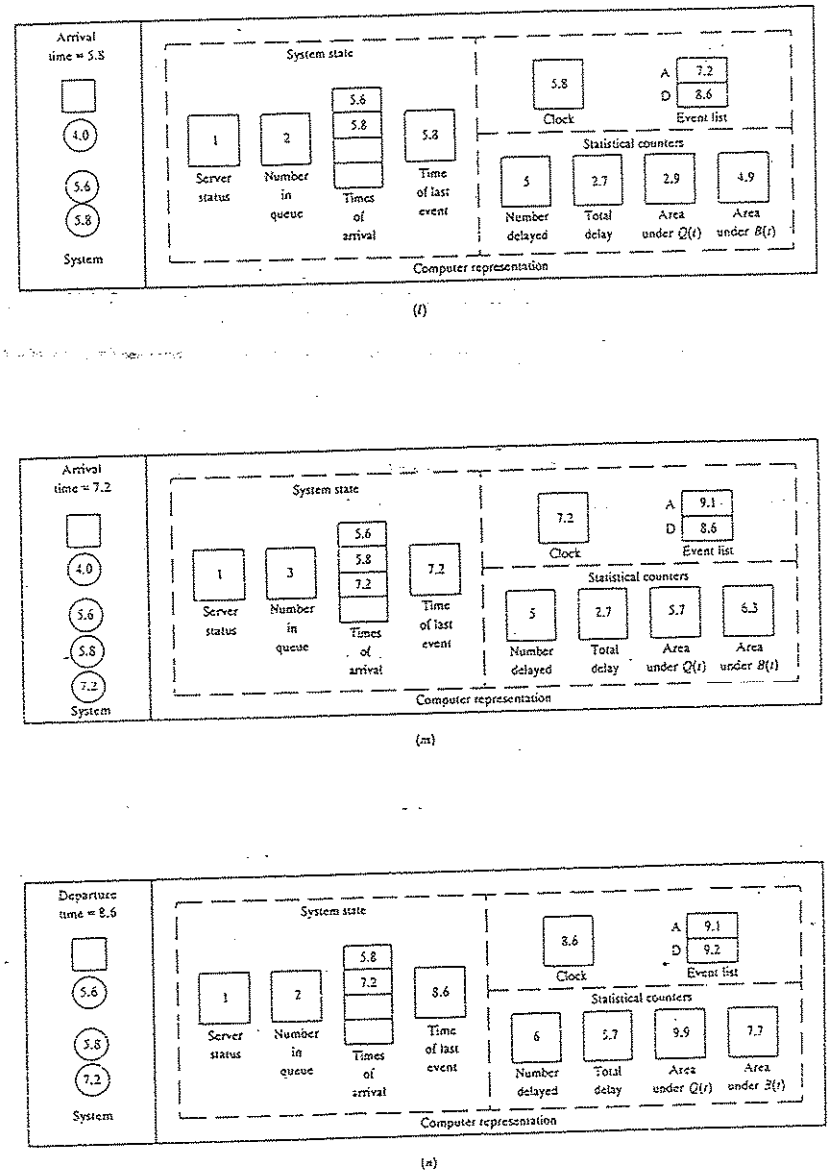


FIGURE 1.7
(*Continued.*)

initially the system is empty of customers and the server is idle, as depicted in the "system" picture of Fig. 1.7*a*. The model state variables are initialized to represent this: server status is zero [we use 0 to represent an idle server and 1 to represent a busy server, similar to the definition of the $B(t)$ function], and the number of customers in the queue is zero. There is a one-dimensional array to store the times of arrival of customers *currently in the queue*; this array is initially empty, and as the simulation progresses its length will grow and shrink. The time of the last (most recent) event is initialized to zero, so that at the time of the first event (when it is used), it will have its correct value. The simulation clock is set to zero, and the *event list*, giving the times of the next occurrence of each of the event types, is initialized as follows. The time of the first arrival is $0 + A_1 = 0.4$, and is denoted by "A" next to the event list. Since there is no customer in service, it does not even make sense to talk about the time of the next departure ("D" by the event list), and we know that the first event will be the initial customer arrival at time 0.4. However, the simulation progresses in general by looking at the event list and picking the smallest value from it to determine what the next event will be, so by scheduling the next departure to occur at time ∞ (or a very large number in a computer program), we effectively eliminate the departure event from consideration and force the next event to be an arrival. (This is sometimes called *poisoning* the departure event.) Finally, the four statistical counters are initialized to zero. When all initialization is done, control is returned to the main program, which then calls the timing routine to determine the next event. Since $0.4 < \infty$, the next event will be an arrival at time 0.4, and the timing routine advances the clock to this time, then passes control back to the main program with the information that the next event is to be an arrival.

$t = 0.4$: *Arrival of customer 1.* At time 0.4, the main program passes control to the arrival routine to process the arrival of the first customer. Figure 1.7*b* shows the system and its computer representation *after* all changes have been made to process this arrival. Since this customer arrived to find the server idle (status equal to 0), he begins service immediately and has a delay in queue of $D_1 = 0$ (which *does* count as a delay). The server status is set to 1 to represent that the server is now busy, but the queue itself is still empty. The clock has been advanced to the current time, 0.4, and the event list is updated to reflect this customer's arrival: The next arrival will be $A_2 = 1.2$ time units from now, at time $0.4 + 1.2 = 1.6$, and the next departure (the service completion of the customer now arriving) will be $S_1 = 2.0$ time units from now, at time $0.4 + 2.0 = 2.4$. The number delayed is incremented to 1 (when this reaches $n = 6$, the simulation will end), and $D_1 = 0$ is added into the total delay (still at zero). The

43

area under $Q(t)$ is updated by adding in the product of the *previous* value (i.e., the level it had between the last event and now) of $Q(t)$ (0 in this case) times the width of the interval of time from the last event to now, $t -$ (time of last event) $= 0.4 - 0$ in this case. Note that the time of the last event used here is its *old* value (0), before it is updated to its new value (0.4) in this event routine. Similarly, the area under $B(t)$ is updated by adding in the product of its previous value (0) times the width of the interval of time since the last event. [Look back at Figs. 1.5 and 1.6 to trace the accumulation of the areas under $Q(t)$ and $B(t)$.] Finally, the time of the last event is brought up to the current time, 0.4, and control is passed back to the main program. It invokes the timing routine, which scans the event list for the smallest value, and determines that the next event will be another arrival at time 1.6; it updates the clock to this value and passes control back to the main program with the information that the next event is an arrival.

$t = 1.6$: *Arrival of customer 2.* At this time we again enter the arrival routine, and Fig. 1.7*c* shows the system and its computer representation after all changes have been made to process this event. Since this customer arrives to find the server busy (status equal to 1 upon her arrival), she must queue up in the first location in the queue, her time of arrival is stored in the first location in the array, and the number-in-queue variable rises to 1. The time of the next arrival in the event list is updated to $A_3 = 0.5$ time units from now, $1.6 + 0.5 = 2.1$; the time of the next departure is not changed, since its value of 2.4 is the departure time of customer 1, who is still in service at this time. Since we are not observing the end of anyone's delay in queue, the number-delayed and total-delay variables are unchanged. The area under $Q(t)$ is increased by 0 [the previous value of $Q(t)$] times the time since the last event, $1.6 - 0.4 = 1.2$. The area under $B(t)$ is increased by 1 [the previous value of $B(t)$] times this same interval of time, 1.2. After updating the time of the last event to now, control is passed back to the main program and then to the timing routine, which determines that the next event will be an arrival at time 2.1.

$t = 2.1$: *Arrival of customer 3.* Once again the arrival routine is invoked, as depicted in Fig. 1.7*d*. The server stays busy, and the queue grows by one customer, whose time of arrival is stored in the queue array's second location. The next arrival is updated to $t + A_4 = 2.1 + 1.7 = 3.8$, and the next departure is still the same, as we are still waiting for the service completion of customer 1. The delay counters are unchanged, since this is not the end of anyone's delay in queue, and the two area accumulators are updated by adding in 1 [the previous values of both $Q(t)$ and $B(t)$] times the time since the last event, $2.1 - 1.6 = 0.5$. After bringing the time of the last event up to the present, we go back to the main program and invoke the timing routine, which looks at the event list to determine that the next event will be a departure at time 2.4, and updates the clock to that time.

$t = 2.4$: *Departure of customer 1.* Now the main program invokes the departure routine, and Fig. 1.7*e* shows the system and its representation after this occurs. The server will maintain its busy status, since customer 2 moves out of the first place in queue and into service. The queue shrinks by one, and the time-of-arrival array is moved up one place, to represent that customer 3 is now first in line. Customer 2, now entering service, will require $S_2 = 0.7$ times units, so the time of the next departure (that of customer 2) in the event list is updated to S_2 time units from now, or at time $2.4 + 0.7 = 3.1$; the time of the next arrival (that of customer 4) is unchanged, since this was scheduled earlier at the time of customer 3's arrival, and we are still waiting at this time for customer 4 to arrive. The delay statistics are updated, since at this time customer 2 is entering service and is completing her delay in queue. Here we make use of the time-of-arrival array, and compute the second delay as the current time minus the second customer's time of arrival, or $D_2 = 2.4 - 1.6 = 0.8$. (Note that the value of 1.6 was stored in the first location in the time-of-arrival array *before* it was changed, so this delay computation would have to be done before advancing the times of arrival in the array.) The area statistics are updated by adding in $2 \times (2.4 - 2.1)$ for $Q(t)$ [note that the previous value of $Q(t)$ was used], and $1 \times (2.4 - 2.1)$ for $B(t)$. The time of the last event is updated, we return to the main program, and the timing routine determines that the next event is a departure at time 3.1.

$t = 3.1$: *Departure of customer 2.* The changes at this departure are similar to those at the departure of customer 1 at time 2.4 just discussed. Note that we observe another delay in queue, and that after this event is processed the queue is again empty, but the server is still busy.

$t = 3.3$: *Departure of customer 3.* Again, the changes are similar to those in the above two departure events, with one important exception: Since the queue is now empty, the server becomes idle and we must set the next departure time in the event list to ∞, since the system now looks the same as it did at time 0 and we want to force the next event to be the arrival of customer 4.

$t = 3.8$: *Arrival of customer 4.* Since this customer arrives to find the server idle, he has a delay of zero (i.e., $D_4 = 0$) and goes right into service. Thus, the changes here are very similar to those at the arrival of the first customer at time $t = 0.4$.

The remaining six event times are depicted in Figs. 1.7*i* through 1.7*n*, and readers should work through these to be sure they understand why the variables and arrays are as they appear; it may be helpful to follow along in the

14

plots of $Q(t)$ and $B(t)$ in Figs. 1.5 and 1.6. With the departure of customer 5 at time $t = 8.6$, customer 6 leaves the queue and enters service, at which time the number delayed reaches 6 (the specified value of n) and the simulation ends. At this point, the main program would invoke the report generator to compute the final output measures [$\hat{d}(6) = 5.7/6 = 0.95$, $\hat{q}(6) = 9.9/8.6 = 1.15$, and $\hat{u}(6) = 7.7/8.6 = 0.90$] and write them out.

A few specific comments about the above example illustrating the logic of a simulation should be made:

- Perhaps the key element in the dynamics of a simulation is the interaction between the simulation clock and the event list. The event list is maintained, and the clock jumps to the next event, as determined by scanning the event list at the end of each event's processing for the smallest (i.e., next) event time. This is how the simulation progresses through time.
- While processing an event, no "simulated" time passes. However, even though time is standing still for the model, care must be taken to process updates of the state variables and statistical counters in the appropriate order. For example, it would be incorrect to update the number in queue before updating the area-under-$Q(t)$ counter, since the height of the rectangle to be used is the *previous* value of $Q(t)$ [before the effect of the current event on $Q(t)$ has been implemented]. Similarly, it would be incorrect to update the time of the last event before updating the area accumulators. Yet another type of error would result if the queue list were changed at a departure before the delay of the first customer in queue were computed, since his time of arrival to the system would be lost.
- It is sometimes easy to overlook contingencies that seem out of the ordinary but that nevertheless must be accommodated. For example, it would be easy to forget that a departing customer could leave behind an empty queue, necessitating that the server be idled and the departure event again be eliminated from consideration. Also, termination conditions are often more involved than they might seem at first sight; in the above example, the simulation stopped in what seems to be the "usual" way, after a departure of one customer, allowing another to enter service and contribute the last delay needed, but the simulation *could* actually have ended instead with an arrival event—how?
- In some simulations it can happen that two (or more) entries in the event list are tied for smallest, and a decision rule must be incorporated to break such *time ties* (this happens with the inventory simulation considered later in Sec. 1.5). The tie-breaking rule can affect the results of the simulation, so must be chosen in accordance with how the system is to be modeled. In many simulations, however, we can ignore the possibility of ties, since the use of continuous random variables may make their occurrence an event with probability zero. In the above model, for example, if the interarrival-time or service-time distribution is continuous, then a time tie in the event list is a probability-zero event.

The above exercise is intended to illustrate the changes and data structures involved in carrying out a discrete-event simulation from the event-scheduling point of view, and contains most of the important ideas needed for more complex simulations of this type. The interarrival and service times used could have been drawn from a random-number table of some sort, constructed to reflect the desired probability distributions; this would result in what might be called a *hand simulation*, which in principle could be carried out to any length. The tedium of doing this should now be clear, so we will next turn to the use of computers (which are not easily bored) to carry out the arithmetic and bookkeeping involved in longer or more complex simulations.

1.4.3 Program Organization and Logic

In this section we set up the necessary ingredients for the programs to simulate the single-server queueing system in FORTRAN (Sec. 1.4.4), Pascal (Sec. 1.4.5), and C (Sec. 1.4.6). The organization and logic described in this section apply for all three languages, so the reader need only go through one of Secs. 1.4.4, 1.4.5, or 1.4.6, according to language preference.

There are several reasons for choosing a general-purpose language such as FORTRAN, Pascal, or C, rather than a more powerful high-level simulation language, for introducing computer simulation at this point:

- By learning to simulate in a general-purpose language, in which one must pay attention to every detail, there will be a greater understanding of how simulations actually operate, and thus less chance of conceptual errors if a switch is later made to a high-level simulation language.
- Despite the fact that there are now several very good and powerful simulation languages available (see Chap. 3), it is often necessary to write at least parts of complex simulations in a general-purpose language if the specific, detailed logic of complex systems is to be represented faithfully.
- General-purpose languages are widely available, and entire simulations are sometimes still written in this way.

It is not our purpose in this book to teach any particular simulation language in detail, although we survey several in Chap. 3. With the understanding promoted by our more general approach and by going through our simulations in this and the next chapter, the reader should find it easier to learn a specialized simulation language. Appendix 1C contains details on the particular computers and compilers used for the examples in this and the next chapter.

The single-server queueing model that we will simulate in the following three sections differs in two respects from the model used in the previous section:

- The simulation will end when $n = 1000$ delays in queue have been completed, rather than $n = 6$, in order to collect more data (and maybe to

15

impress the reader with the patience of computers, since we have just slugged it out by hand in the $n = 6$ case in the preceding section). It is important to note that this change in the stopping rule changes the model itself, in that the output measures are defined relative to the stopping rule; hence the "n" in the notation for the quantities $d(n)$, $q(n)$, and $u(n)$ being estimated.

- The interarrival and service times will now be modeled as independent random variables from exponential distributions with mean 1 minute for the interarrival times and mean 0.5 minute for the service times. The exponential distribution with mean β (any positive real number) is continuous, with probability density function

$$f(x) = \frac{1}{\beta} e^{-x/\beta} \qquad \text{for } x \geq 0$$

(See Chaps. 4 and 6 for more information on density functions in general, and on the exponential distribution in particular.) We make this change here since it is much more common to generate input quantities (which drive the simulation) such as interarrival and service times from specified distributions than to assume that they are "known" as we did in the preceding section. The choice of the exponential distribution with the above particular values of β is essentially arbitrary, and is made primarily because it is easy to generate exponential random variates on a computer. (Actually, the assumption of exponential interarrival times is often quite realistic; assuming exponential service times, however, is seldom plausible.) Chapter 6 addresses in detail the important issue of how one chooses distribution forms and parameters for modeling simulation input random variables.

The single-server queue with exponential interarrival and service times is commonly called the *M/M/1 queue*, as discussed in App. 1B.

To simulate this model we need a way to generate random variates from an exponential distribution. The subprograms used by the FORTRAN, Pascal, and C codes all operate in the same way, which we will now develop. First, a *random-number generator* (discussed in detail in Chap. 7) is invoked to generate a variate U that is distributed (continuously) uniformly between 0 and 1; this distribution will henceforth be referred to as U(0, 1) and has probability density function

$$f(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

It is easy to show that the probability that a U(0, 1) random variable falls in any subinterval $[x, x + \Delta x]$ contained in the interval $[0, 1]$ is (uniformly) Δx (see Sec. 6.2.2). The U(0, 1) distribution is fundamental to simulation modeling because, as we shall see in Chap. 8, a random variate from any distribution can be generated by first generating one or more U(0, 1) random variates and then performing some kind of transformation. After obtaining U, we shall take

the natural logarithm of it, multiply the result by β, and finally change the sign to return what we will show to be an exponential random variate with mean β, that is, $-\beta \ln U$.

To see why this algorithm works, recall that the (*cumulative*) *distribution function* of a random variable X is defined, for any real x, to be $F(x) = P(X \leq x)$ (Chap. 4 contains a review of basic probability theory). If X is exponential with mean β, then

$$F(x) = \int_0^x \frac{1}{\beta} e^{-t/\beta}\, dt$$
$$= 1 - e^{-x/\beta}$$

for any real $x \geq 0$, since the probability density function of the exponential distribution at the argument $t \geq 0$ is $(1/\beta)e^{-t/\beta}$. To show that our method is correct, we can try to verify that the value it returns will be less than or equal to x (any nonnegative real number), with probability $F(x)$ given above:

$$\begin{aligned} P(-\beta \ln U \leq x) &= P\left(\ln U \geq -\frac{x}{\beta}\right) \\ &= P(U \geq e^{-x/\beta}) \\ &= P(e^{-x/\beta} \leq U \leq 1) \\ &= 1 - e^{-x/\beta} \end{aligned}$$

The first line in the above is obtained by dividing through by $-\beta$ (recall that $\beta > 0$, so $-\beta < 0$ and the inequality reverses), the second line is obtained by exponentiating both sides (the exponential function is monotone increasing, so the inequality is preserved), the third line is just rewriting, together with knowing that U is in $[0, 1]$ anyway, and the last line follows since U is U(0, 1), and the interval $[e^{-x/\beta}, 1]$ is contained within the interval $[0, 1]$. Since the last line is $F(x)$ for the exponential distribution, we have verified that our algorithm is correct. Chapter 8 discusses how to generate random variates and processes in general.

In our programs, we prefer to use a particular method for random-number generation to obtain the variate U described above, as expressed in the FORTRAN, Pascal, and C codes of Figs. 7.5 through 7.8 in App. 7A of Chap. 7. While most compilers do have some kind of built-in random-number generator, many of these are of extremely poor quality and should not be used; this issue is discussed fully in Chap. 7.

It is convenient (if not the most computationally efficient) to modularize the programs into several subprograms to clarify the logic and interactions, as discussed in general in Sec. 1.3.2. In addition to a main program, the simulation program includes routines for initialization, timing, report generation, and generating exponential random variates, as in Fig. 1.3. It also simplifies matters if we write a separate routine to update the continuous-time

16

statistic , being the accumulated areas under the $Q(t)$ and $B(t)$ curves. The most important action, however, takes place in the routines for the events, which we number as follows:

Event description	Event type
Arrival of a customer to the system	1
Departure of a customer from the system after completing service	2

As the logic of these event routines is independent of the particular language to be used, we shall discuss it here. Figure 1.8 contains a flowchart for

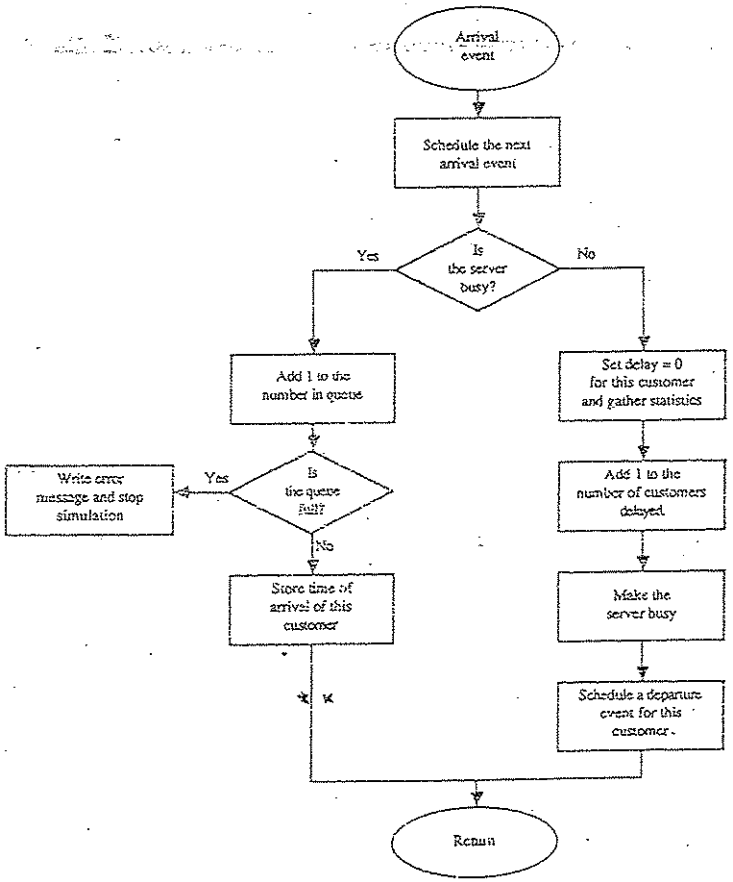


FIGURE 1.8
Flowchart for arrival routine, queueing model.

the arrival event. First, the time of the next arrival in the future is generated and placed in the event list. Then a check is made to determine whether the server is busy. If so, the number of customers in the queue is incremented by one, and we ask whether the storage space allocated to hold the queue is already full (see the code in Sec. 1.4.4, 1.4.5, or 1.4.6 for details). If the queue

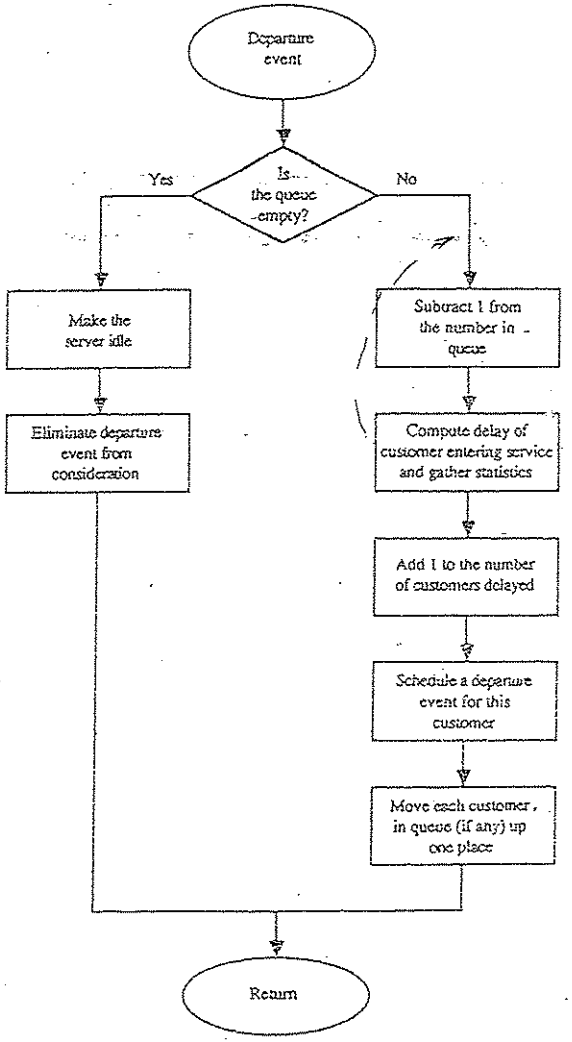
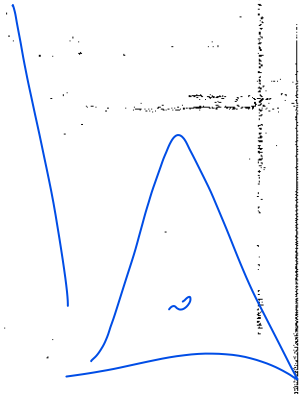


FIGURE 1.9
Flowchart for departure routine, queueing model.



17

is already full, an error message is produced and the simulation is stopped; if there is still room in the queue, the arriving customer's time of arrival is put at the (new) end of the queue. On the other hand, if the arriving customer finds the server idle, then this customer has a delay of zero, which is counted as a delay, and the number of customer delays completed is incremented by one. The server must be made busy, and the time of departure from service of the arriving customer is scheduled into the event list.

The departure event's logic is depicted in the flowchart of Fig. 1.9. Recall that this routine is invoked when a service completion (and subsequent departure) occurs. If the departing customer leaves no other customers behind in queue, the server is idled and the departure event is eliminated from consideration, since the next event must be an arrival. On the other hand, if one or more customers are left behind by the departing customer, the first customer in queue will leave the queue and enter service, so the queue length is reduced by one, and the delay in queue of this customer is computed and registered in the appropriate statistical counter. The number delayed is increased by one, and a departure event for the customer now entering service is scheduled. Finally, the rest of the queue (if any) is advanced one place. Our implementation of the list for the queue will be very simple in this chapter, and is certainly not the most efficient; Chap. 2 discusses better ways of handling lists to model such things as queues.

In the next three sections we give examples of how the above setup can be used to write simulation programs in FORTRAN, Pascal, and C. Again, only one of these sections need be studied, depending on language familiarity or preference; the logic and organization is essentially identical, except for changes dictated by a particular language's features or shortcomings. The results (which were identical across all three languages) are discussed in Sec. 1.4.7. These programs are neither the simplest nor most efficient possible, but were instead designed to illustrate how one might organize programs for more complex simulations.

1.4.7 Simulation Output and Discussion

The output (in a file named mm1.out if the FORTRAN or C program above was used) is shown in Fig. 1.37; since the same method for random-number generation was used for the programs in all three languages, they produced identical results. In this run, the average delay in queue was 0.430 minute, there was an average of 0.418 customer in the queue, and the server was busy

```
Single-server queueing system

Mean interarrival time        1.000 minutes

Mean service time             0.500 minutes

Number of customers            1000

Average delay in queue        0.430 minutes

Average number in queue       0.418

Server utilization            0.460

Time simulation ended      1027.915 minutes
```

FIGURE 1.37
Output report, queueing model.

46 percent of the time. It took 1027.915 simulated minutes to run the simulation to the completion of 1000 delays, which seems reasonable since the expected time between customer arrivals was 1 minute. (It is not a coincidence that the average delay, average number in queue, and utilization are all so close together for this model; see App. 1B.)

Note that these particular numbers in the output were determined, at root, by the numbers the random-number generator happened to come up with this time. If a different random-number generator were used, or if this one were used in another way (with another "seed" or "stream," as discussed in Chap. 7), then different numbers would have been produced in the output. Thus, these numbers are not to be regarded as "The Answers," but rather as estimates (and perhaps poor ones) of the expected quantities we want to know about, $d(n)$, $q(n)$, and $u(n)$; the statistical analysis of simulation output data is discussed in Chaps. 9 through 12. Also, the results are functions of the input parameters, in this case the mean interarrival and service times, and the $n = 1000$ stopping rule; they are also affected by the way we initialized the simulation (empty and idle).

In some simulation studies, we might want to estimate *steady-state* characteristics of the model (see Chap. 9), i.e., characteristics of a model after the simulation has been running a very long (in theory, an infinite) amount of time. For the simple $M/M/1$ queue we have been considering, it is possible to compute *analytically* the steady-state average delay in queue, the steady-state time-average number in queue, and the steady-state server utilization, all of these measures of performance being 0.5 [see, for example, Ross (1989, p. 352)]. Thus, if we wanted to determine these steady-state measures, our estimates based on the stopping rule $n = 1000$ delays were not too far off, at least in absolute terms. However, we were somewhat lucky, since $n = 1000$ was chosen arbitrarily! In practice, the choice of a stopping rule that will give good estimates of steady-state measures is quite difficult. To illustrate this point, suppose for the $M/M/1$ queue that the arrival rate of customers were increased from 1 per minute to 1.98 per minute (the mean interarrival time is now 0.505 minute), that the mean service time is unchanged, and that we wish to estimate the steady-state measures from a run of length $n = 1000$ delays, as before. We performed this simulation run and got values for the average delay, average number in queue, and server utilization of 17.404 minutes, 34.831, and 0.997, respectively. Since the true steady-state values of these measures are 49.5 minutes, 98.01, and 0.99 (respectively), it is clear that the stopping rule cannot be chosen arbitrarily. We discuss how to specify the run length for a steady-state simulation in Chap. 9.

The reader may have wondered why we did not estimate the expected average waiting time in the system of a customer, $w(n)$, rather than the expected average delay in queue, $d(n)$, where the waiting time of a customer is defined as the time interval from the instant the customer arrives to the instant the customer completes service and departs. There were two reasons. First, for many queueing systems we believe that the customer's delay in queue while

19

waiting for other customers to be served is the most troublesome part of the customer's wait in the system. Moreover, if the queue represents part of a manufacturing system where the "customers" are actually parts waiting for service at a machine (the "server"), then the delay in queue represents a loss, whereas the time spent in service is "necessary." Our second reason for focusing on the delay in queue is one of statistical efficiency. The usual estimator of $w(n)$ would be

$$\hat{w}(n) = \frac{\sum_{i=1}^{n} W_i}{n} = \frac{\sum_{i=1}^{n} D_i}{n} + \frac{\sum_{i=1}^{n} S_i}{n} = \hat{d}(n) + \bar{S}(n) \tag{1.7}$$

where $W_i = D_i + S_i$ is the waiting time in system of the ith customer and $\bar{S}(n)$ is the average of the n customers' service times. Since the service-time distribution would have to be known to perform a simulation in the first place, the expected or mean service time, $E(S)$, would also be known and an alternative estimator of $w(n)$ is

$$\tilde{w}(n) = \hat{d}(n) + E(S)$$

[Note that $\bar{S}(n)$ is an unbiased estimator of $E(S)$ in Eq. (1.7).] In almost all queueing simulations, $\tilde{w}(n)$ will be a more efficient (less variable) estimator of $w(n)$ than $\hat{w}(n)$ and is thus preferable (both estimators are unbiased). Therefore, if one wants an estimate of $w(n)$, estimate $d(n)$ and add the known expected service time, $E(S)$. In general, the moral is to replace estimators by their expected values whenever possible (see the discussion of indirect estimators in Sec. 11.5).

1.5 SIMULATION OF AN INVENTORY SYSTEM

We shall now see how simulation can be used to compare alternative ordering policies for an inventory system. Many of the elements of our model are representative of those found in actual inventory systems.

1.5.1 Problem Statement

A company that sells a single product would like to decide how many items it should have in inventory for each of the next n months. The times between demands are IID exponential random variables with a mean of 0.1 month. The sizes of the demands, D, are IID random variables (independent of when the demands occur), with

$$D = \begin{cases} 1 & \text{w.p. } \frac{1}{6} \\ 2 & \text{w.p. } \frac{1}{3} \\ 3 & \text{w.p. } \frac{1}{3} \\ 4 & \text{w.p. } \frac{1}{6} \end{cases}$$

where w.p. is read "with probability."

At the beginning of each month, the company reviews the inventory level and decides how many items to order from its supplier. If the company orders Z items, it incurs a cost of $K + iZ$, where $K = \$32$ is the *setup cost* and $i = \$3$ is the *incremental cost* per item ordered. (If $Z = 0$, no cost is incurred.) When an order is placed, the time required for it to arrive (called the *delivery lag* or *lead time*) is a random variable that is distributed uniformly between 0.5 and 1 month.

The company uses a stationary (s, S) policy to decide how much to order, i.e.,

Maximo stock

Cantidad pedida

$$Z = \begin{cases} S - I & \text{if } I < s \\ 0 & \text{if } I \geq s \end{cases}$$

Cantidad de stock

Minimo stock

where I is the inventory level at the beginning of the month.

When a demand occurs, it is satisfied immediately if the inventory level is at least as large as the demand. If the demand exceeds the inventory level, the excess of demand over supply is backlogged and satisfied by future deliveries. (In this case, the new inventory level is equal to the old inventory level minus the demand size, resulting in a negative inventory level.) When an order arrives, it is first used to eliminate as much of the backlog (if any) as possible; the remainder of the order (if any) is added to the inventory.

costos extra

So far we have discussed only one type of cost incurred by the inventory system, the ordering cost. However, most real inventory systems also have two additional types of costs, *holding* and *shortage* costs, which we discuss after introducing some additional notation. Let $I(t)$ be the inventory level at time t [note that $I(t)$ could be positive, negative, or zero], let $I^+(t) = \max\{I(t), 0\}$ be the number of items physically on hand in the inventory at time t [note that $I^+(t) \geq 0$], and let $I^-(t) = \max\{-I(t), 0\}$ be the backlog at time t [$I^-(t) \geq 0$ as well]. A possible realization of $I(t)$, $I^+(t)$, and $I^-(t)$ is shown in Fig. 1.54. The time points at which $I(t)$ decreases are the ones at which demands occur.

For our model, we shall assume that the company incurs a holding cost of $h = \$1$ per item per month held in (positive) inventory. The holding cost includes such costs as warehouse rental, insurance, taxes, and maintenance, as well as the opportunity cost of having capital tied up in inventory rather than invested elsewhere. We have ignored in our formulation the fact that some holding costs are still incurred when $I^+(t) = 0$. However, since our goal is to *compare* ordering policies, ignoring this factor, which after all is independent of the policy used, will not affect our assessment of which policy is best. Now, since $I^+(t)$ is the number of items held in inventory at time t, the time-average (per month) number of items held in inventory for the n-month period is

$$\bar{I}^+ = \frac{\int_0^n I^+(t)\,dt}{n}$$

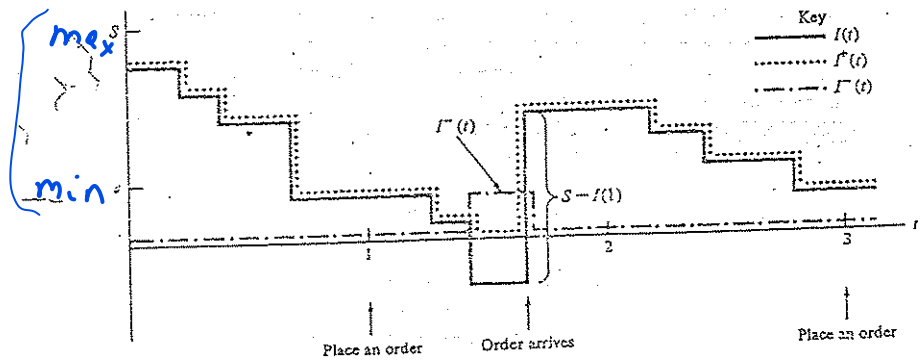


FIGURE 1.54
A realization of $I(t)$, $I^+(t)$, and $I^-(t)$ over time.

Que Pro? Preguntas

$\{ hI^+ + PI^- + iZ + K$

Preparación (setup)

costo incremental (costo unitario)

which is akin to the definition of the time-average number of customers in queue given in Sec. 1.4.1. Thus, the average holding cost per month is $h\bar{I}^+$.

Pers. ida

Similarly, suppose that the company incurs a backlog cost of $\pi = \$5$ per item per month in backlog; this accounts for the cost of extra record keeping when a backlog exists, as well as loss of customers' goodwill. The time-average number of items in backlog is

$$\bar{I}^- = \frac{\int_0^n I^-(t)\,dt}{n}$$

so the average backlog cost per month in $\pi\bar{I}^-$.

Assume that the initial inventory level is $I(0) = 60$ and that no order is outstanding. We simulate the inventory system for $n = 120$ months and use the average total cost per month (which is the sum of the average ordering cost per month, the average holding cost per month, and the average shortage cost per month) to compare the following nine inventory policies:

s	20	20	20	20	40	40	40	60	60
S	40	60	80	100	60	80	100	80	100

We do not address here the issue of how these particular policies were chosen for consideration; statistical techniques for making such a determination are discussed in Chap. 12.

It should be noted that the state variables for a simulation model of this inventory system are the inventory level $I(t)$, the amount of an outstanding order from the company to the supplier, and the time of the last event [which is needed to compute the areas under the $I^+(t)$ and $I^-(t)$ functions].

1.5.2 Program Organization and Logic

Our model of the inventory system uses the following types of events:

Event description	Event type
Arrival of an order to the company from the supplier	1
Demand for the product from a customer	2
End of the simulation after n months	3
Inventory evaluation (and possible ordering) at the beginning of a month	4

We have chosen to make the end of the simulation event type 3 rather than type 4, since at time 120 both "end-simulation" and "inventory-evaluation" events will eventually be scheduled and we would like to execute the former event first at this time. (Since the simulation is over at time 120, there is no sense in evaluating the inventory and possibly ordering, incurring an ordering cost for an order that will never arrive.) The execution of event type 3 before event type 4 is guaranteed because the timing routines (in all three languages) give preference to the lowest-numbered event if two or more events are

22

scheduled to occur at the same time. In general, a simulation model should be designed to process events in an appropriate order when time ties occur. An event graph (see Sec. 1.4.9) appears in Fig. 1.55.

There are three types of random variates needed to simulate this system. The interdemand times are distributed exponentially, so the same algorithm (and code) as developed in Sec. 1.4 can be used here. The demand-size random variate D must be discrete, as described above, and can be generated as follows. First divide the unit interval into the contiguous subintervals $C_1 = [0, \frac{1}{6})$, $C_2 = [\frac{1}{6}, \frac{1}{2})$, $C_3 = [\frac{1}{2}, \frac{5}{6})$, and $C_4 = [\frac{5}{6}, 1]$, and obtain a U(0, 1) random variate U from the random-number generator. If U falls in C_1, return $D = 1$; if U falls in C_2, return $D = 2$; and so on. Since the width of C_1 is $\frac{1}{6} - 0 = \frac{1}{6}$, and since U is uniformly distributed over [0, 1], the probability that U falls in C_1 (and thus that we return $D = 1$) is $\frac{1}{6}$; this agrees with the desired probability that $D = 1$. Similarly, we return $D = 2$ if U falls in C_2, having probability equal to the width of C_2, $\frac{1}{2} - \frac{1}{6} = \frac{1}{3}$, as desired, and so on for the other intervals. The subprograms to generate the demand sizes all use this principle, and take as input the cutoff points defining the above subintervals, which are the *cumulative* probabilities of the distribution of D.

The delivery lags are uniformly distributed, but not over the unit interval [0, 1]. In general, we can generate a random variate distributed uniformly over any interval $[a, b]$ by generating a U(0, 1) random number U, and then returning $a + U(b - a)$. That this method is correct seems intuitively clear, but will be formally justified in Sec. 8.3.1.

Of the four events, only three actually involve state changes (the end-simulation event being the exception). Since their logic is language-independent, we will describe it here.

The order-arrival event is flowcharted in Fig. 1.56, and must make the changes necessary when an order (which was previously placed) arrives from the supplier. The inventory level is increased by the amount of the order, and

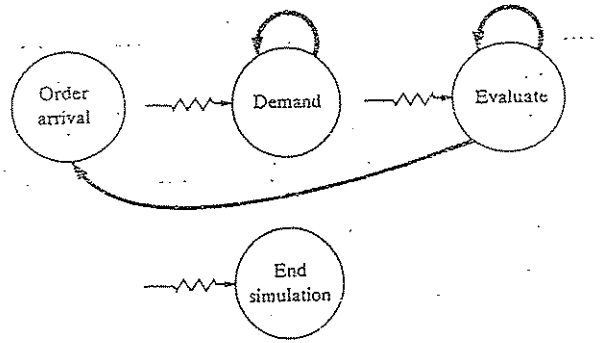


FIGURE 1.55
Event graph, inventory model.

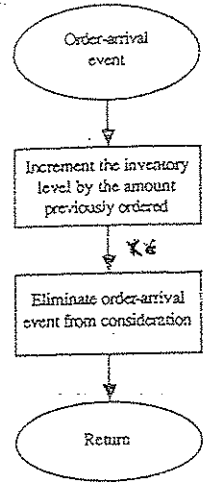


FIGURE 1.56
Flowchart for order-arrival routine, inventory model.

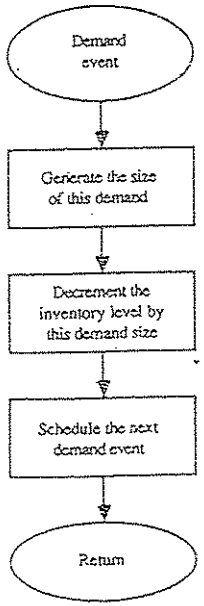


FIGURE 1.57
Flowchart for demand routine, inventory model.

23

the order-arrival event must be eliminated from consideration. (See Prob. 1.12 for consideration of the issue of whether there could be more than one order outstanding at a time for this model with these parameters.)

A flowchart for the demand event is given in Fig. 1.57, and processes the changes necessary to represent a demand's occurrence. First, the demand size is generated, and the inventory is decremented by this amount. Finally, the time of the next demand is scheduled into the event list. Note that this is the place where the inventory level might become negative.

The inventory-evaluation event, which takes place at the beginning of each month, is flowcharted in Fig. 1.58. If the inventory level $I(t)$ at the time of the evaluation is at least s, then no order is placed, and nothing is done except

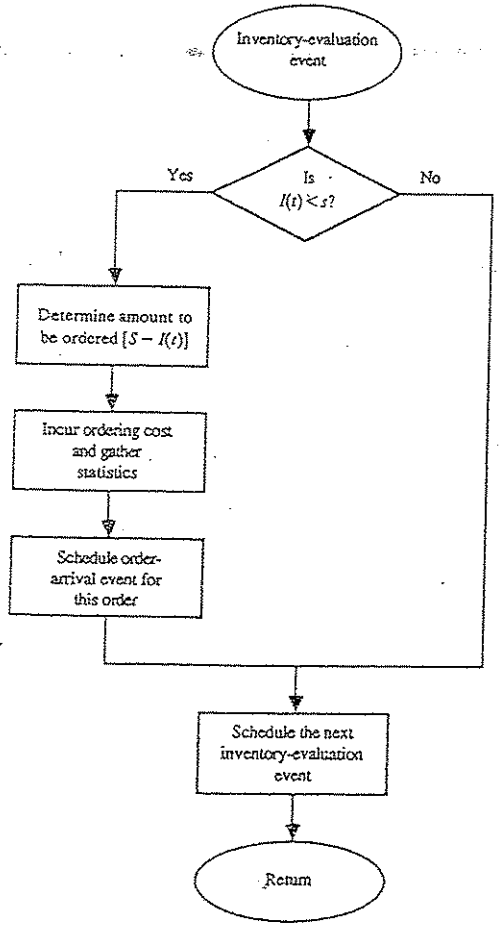


FIGURE 1.58
Flowchart for inventory-evaluation routine, inventory model.

to schedule the next evaluation into the event list. On the other hand, if $I(t) < s$, we want to place an order for $S - I(t)$ items. This is done by storing the amount of the order $[S - I(t)]$ until the order arrives, and scheduling its arrival time. In this case as well, we want to schedule the next inventory-evaluation event.

As in the single-server queueing model, it is convenient to write a separate nonevent routine to update the continuous-time statistical accumulators. For this model, however, doing so is slightly more complicated, so a flowchart for this activity appears in Fig. 1.59. The principal issue is whether we need to update the area under $I^-(t)$ or $I^+(t)$ (or neither). If the inventory level since the last event has been negative, then we have been in backlog, so the area under $I^-(t)$ only should be updated. On the other hand, if the inventory level has been positive, we need only update the area under $I^+(t)$. If the inventory level has been zero (a possibility), then neither update is needed. The code in each language for this routine also brings the variable for the time of the last event up to the present time. This routine will be invoked from the main program just after returning from the timing routine, regardless of the event type or whether the inventory level is actually changing at this point. This provides a simple (if not the most computationally efficient) way of updating integrals for continuous-time statistics.

Sections 1.5.3, 1.5.4, and 1.5.5, respectively, contain programs to simu-

Preguntar

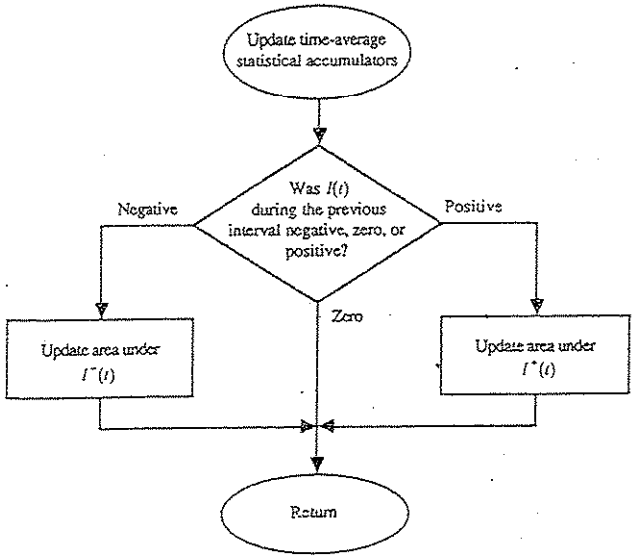


FIGURE 1.59
Flowchart for routine to update the continuous-time statistical accumulators, inventory model.

24

late this model in FORTRAN, Pascal, and C. As in the single-server queueing model, only one of these sections should be read, according to language preference. Neither the timing nor exponential-variate-generation subprograms will be shown, as they are the same as for the single-server queueing model in Sec. 1.4 (except for the FORTRAN version of TIMING, where the declarations file "mm1.dcl" must be changed to "inv.dcl" in the INCLUDE statement). The reader should also note the considerable similarity between the main programs of the queueing and inventory models in a given language.

25

1.5.6 Simulation Output and Discussion

The simulation report (in file inv.out if either the FORTRAN or C version was used) is given in Fig. 1.90. For this model, there were some differences in the results across different languages, compilers, and computers, even though the same random-number-generator algorithm was being used; see App. 1C for details and an explanation of this discrepancy.

The three separate components of the average total cost per month were reported to see how they respond individually to changes in s and S, as a possible check on the model and the code. For example, fixing $s = 20$ and

```
Single-product inventory system

Initial inventory level                          60 items

Number of demand sizes                            4

Distribution function of demand sizes        0.167   0.500   0.833   1.000

Mean interdemand time                        0.10 months

Delivery lag range                           0.50 to      1.00 months

Length of the simulation                      120 months

K =  32.0    i =   3.0    h =   1.0    pi =   5.0

Number of policies                                9
```

Policy	Average total cost	Average ordering cost	Average holding cost	Average shortage cost
(20, 40)	126.61	99.26	9.25	18.10
(20, 60)	122.74	90.52	17.39	14.83
(20, 80)	123.86	87.36	26.24	10.26
(20,100)	125.32	81.37	36.00	7.95
(40, 60)	126.37	98.43	25.99	1.95
(40, 80)	125.46	88.40	35.92	1.14
(40,100)	132.34	84.62	46.42	1.30
(60, 80)	150.02	105.69	44.02	0.31
(60,100)	143.20	89.05	53.91	0.24

FIGURE 1.90
Output report, inventory model.

(30)

E.J. Ehmke

increasing S from 40 to 100 increases the holding cost steadily from \$9.25 per month to \$36.00 per month, while reducing shortage cost at the same time; the effect of this increase in S on the ordering cost is to reduce it, evidently since ordering up to larger values of S implies that these larger orders will be placed less frequently, thereby avoiding the fixed ordering cost more often. Similarly, fixing S at, say, 100, and increasing s from 20 to 60 leads to a decrease in shortage cost (\$7.95, \$1.30, \$0.24) but an increase in holding cost (\$36.00, \$46.42, \$53.91), since increases in s translate into less willingness to let the inventory level fall to low values. While we could probably have predicted the *direction* of movement of these components of cost without doing the simulation, it would not have been possible to have said much about their *magnitude* without the aid of the simulation output.

Since the overall criterion of *total* cost per month is the sum of three components that move in sometimes different directions in reaction to changes in s and S, we cannot predict even the direction of movement of this criterion without the simulation. Thus, we simply look at the values of this criterion, and it would *appear* that the (20, 60) policy is the best, having an average total cost of \$122.74 per month. However, in the present context where the length of the simulation is fixed (the company wants a planning horizon of 10 years), what we *really* want to estimate for each policy is the *expected* average total cost per month for the first 120 months. The numbers in Fig. 1.90 are *estimates* of these expected values, each estimate based on a sample of size *1* (simulation run or replication). Since these estimates may have large variances, the ordering of them may differ considerably from the ordering of the expected values, which is the desired information. In fact, if we reran the nine simulations using different U(0, 1) random variates, the estimates obtained might differ greatly from those in Fig. 1.90. Furthermore, the ordering of the new estimates might also be different.

We conclude from the above discussion that when the simulation run length is fixed by the problem context, it will generally not be sufficient to make a single simulation run of each policy or system of interest. In Chap. 9 we address the issue of just how many runs are required to get a good estimate of a desired expected value. Chapters 10 and 12 consider related problems when we are concerned with several different expected values arising from alternative system designs.

26

1.1 Introducción a la simulación

⇒ PPT Dada en clase 12/04

En años recientes, el advenimiento de nuevos y mejores desarrollos en el área de la computación ha traído consigo innovaciones igualmente importantes en los terrenos de la toma de decisiones y el diseño de procesos y productos. En este sentido, una de las técnicas de mayor impacto es la **simulación**.

Hoy en día, el analista tiene a su disposición una gran cantidad de software de simulación que le permite tomar decisiones en temas muy diversos. Por ejemplo, determinar la mejor localización de una nueva planta, diseñar un nuevo sistema de trabajo o efectuar el análisis productivo de un proceso ya existente pero que requiere mejoras. Sin duda, la facilidad que otorga a la resolución de éstas y muchas otras problemáticas, ha hecho de la simulación una herramienta cuyo uso y desarrollo se han visto significativamente alentados. Cada vez resulta más sencillo encontrar paquetes de software con gran capacidad de análisis, así como mejores animaciones y características para generación de reportes. En general, dichos paquetes —ya sea orientados a procesos, a servicios o de índole general— nos proveen de una enorme diversidad de herramientas estadísticas que permiten un manejo más eficiente de la información relevante bajo análisis, y una mejor presentación e interpretación de la misma.

El concepto de simulación engloba soluciones para muchos propósitos diferentes. Por ejemplo, podríamos decir que el modelo de un avión a escala que se introduce a una cámara por donde se hace pasar un flujo de aire, puede simular los efectos que experimentará un avión real cuando se vea sometido a turbulencia. Por otro lado, algunos paquetes permiten hacer la representación de un proceso de fresado o torneado: una vez que el usuario establezca ciertas condiciones iniciales, podrá ver cómo se llevaría a cabo el proceso real, lo que le permitiría revisarlo sin necesidad de desperdiciar material ni poner en riesgo la maquinaria.

Entre los distintos tipos de procesos de simulación que podemos utilizar, en este libro nos ocuparemos del que se basa en el uso de ecuaciones matemáticas y estadísticas, conocido como **simulación de eventos discretos**. Este proceso consiste en relacionar los diferentes eventos que pueden cambiar el estado de un sistema bajo estudio por medio de distribuciones de probabilidad y condiciones lógicas del problema que se esté analizando. Por ejemplo, un proceso de inspección donde sabemos estadísticamente que 0.2% de los productos tiene algún tipo de defecto puede simularse con facilidad mediante una simple hoja de cálculo, considerando estadísticas de rechazos y productos conformes, y asignando una distribución de probabilidad con 0.2% de oportunidad de defecto para cada intento de inspección.

En el presente capítulo abordaremos las definiciones básicas de los conceptos de la simulación de eventos discretos. En los siguientes se presentarán algunos otros elementos relevantes, como los números pseudo aleatorios y las pruebas estadísticas necesarias para comprobar esta aleatoriedad, la generación de variables aleatorias y la caracterización de algunas distribuciones de probabilidad de uso común en la simulación, lo cual nos permitirá realizar una simulación sencilla con ayuda de una hoja de cálculo. Por último, describiremos la utilización de un software comercial: Promodel, una versión limitada del cual se incluye en este libro.

1.2 Definiciones de simulación

Para poder realizar un buen estudio de simulación es necesario entender los conceptos básicos que componen nuestro modelo.

Comenzaremos por definir el concepto de **simulación de eventos discretos** como el *conjunto de relaciones lógicas, matemáticas y probabilísticas que integran el comportamiento de un sistema bajo estudio cuando se presenta un evento determinado*. El objetivo del modelo de simulación consiste, precisamente, en comprender, analizar y mejorar las condiciones de operación relevantes del sistema.

En la definición anterior encontramos elementos como sistema, modelo y evento, de los cuales se desprenden otros conceptos importantes dentro de una simulación, por lo que a continuación abundaremos en cada uno de ellos.

La definición básica de **sistema** nos dice que se trata de un *conjunto de elementos que se interrelacionan para funcionar como un todo*; desde el punto de vista de la simulación, tales elementos deben tener una frontera clara. Por ejemplo, podemos hablar del sistema de atención de clientes en un banco, del sistema de inventarios de una empresa o del sistema de atención en la sala de emergencia de un hospital. Cada uno de ellos puede dividirse en elementos que son relevantes para la construcción de lo que constituirá su modelo de simulación; entre ellos tenemos entidades, estado del sistema, eventos actuales y futuros, localizaciones, recursos, atributos, variables y el reloj de la simulación.

Una **entidad** es la *representación de los flujos de entrada a un sistema*; éste es el elemento responsable de que el estado del sistema cambie. Ejemplos de entidades pueden ser los clientes que llegan a la caja de un banco, las piezas que llegan a un proceso o el embarque de piezas que llega a un inventario.

El **estado del sistema** es la *condición que guarda el sistema bajo estudio en un momento determinado*; es como una fotografía de lo que está pasando en el sistema en cierto instante. El estado del sistema se compone de variables o características de operación puntuales (digamos el número de piezas que hay en el sistema en ese momento), y de variables o características de operación acumuladas, o promedio (como podría ser el tiempo promedio de permanencia de una entidad en el sistema, en una fila, almacén o equipo).

Un **evento** es un *cambio en el estado actual del sistema*; por ejemplo, la entrada o salida de una entidad, la finalización de un proceso en un equipo, la interrupción o reactivación de una operación (digamos por un descanso del operario), o la descompostura de una máquina. Podemos catalogar estos eventos en dos tipos: **eventos actuales**, que son aquellos que están sucediendo en el sistema en un momento dado, y **eventos futuros**, que son cambios que se presentarán en el sistema después del tiempo de simulación, de acuerdo con una programación específica. Por ejemplo, imagine que cierta pieza entra a una máquina para que ésta realice un proceso. El evento actual sería precisamente que la entidad llamada "pieza" se encuentra en la máquina. El evento futuro podría ser el momento en que la máquina concluirá su trabajo con la pieza y ésta seguirá su camino hacia el siguiente proceso lógico, de acuerdo con la programación: almacenamiento, inspección o entrada a otra máquina.

Las **localizaciones** son todos aquellos *lugares en los que la pieza puede detenerse para ser transformada o esperar a serlo*. Dentro de estas localizaciones tenemos almacenes, bandas transportadoras, máquinas, estaciones de inspección, etcétera.

Los **recursos** son aquellos *dispositivos* —diferentes a las localizaciones— *necesarios para llevar a cabo una operación*. Por ejemplo, un montacargas que transporta una pieza de un lugar a otro: una persona que realiza la inspección en una estación y toma turnos para descansar; una herramienta necesaria para realizar un proceso pero que no forma parte de una localización específica, sino que es trasladada de acuerdo con los requerimientos de aquel.

Un **atributo** es una *característica de una entidad*. Por ejemplo, si la entidad es un motor, los atributos serían su color, peso, tamaño o cilindraje. Los atributos son muy útiles para diferenciar entidades sin necesidad de generar una entidad nueva, y pueden adjudicarse al momento de la creación de la entidad, o asignarse y/o cambiarse durante el proceso.

Como indica su nombre, las **variables** son *condiciones cuyos valores se crean y modifican por medio de ecuaciones matemáticas y relaciones lógicas*. Pueden ser *continuas* (por ejemplo, el costo promedio de operación de un sistema) o *discretas* (por ejemplo, el número de unidades que deberá empacarse en un contenedor). Las variables son muy útiles para realizar conteos de piezas y ciclos de operación, así como para determinar características de operación del sistema.

El **reloj de la simulación** es el *contador de tiempo de la simulación*, y su función consiste en responder preguntas tales como cuánto tiempo se ha utilizado el modelo en la simulación, y cuánto tiempo en total se quiere que dure esta última. En general, el reloj de simulación se relaciona con la tabla de eventos futuros, pues al cumplirse el tiempo programado para la realización de un evento futuro, éste se convierte en un evento actual. Regresando al ejemplo de la pieza en la máquina, cuando el tiempo de proceso se cumpla, la pieza seguirá su camino hasta su siguiente localización; el reloj de la simulación simula precisamente ese tiempo.

Podemos hablar de dos tipos de reloj de simulación: el **reloj de simulación absoluto**, que parte de cero y termina en un tiempo total de simulación definido, y el **reloj de simulación relativo**, que sólo considera el lapso de tiempo que transcurre entre dos eventos. Por ejemplo, podemos decir que el tiempo de proceso de una pieza es relativo, mientras que el tiempo absoluto sería el tiempo global de la simulación: desde que la pieza entró a ser procesada hasta el momento en el que terminó su proceso.

Como se mencionó antes, existen distintos **modelos** de simulación que permiten representar situaciones reales de diferentes tipos. Podemos tener modelos físicos —como el del avión que mencionamos en la sección anterior— o modelos matemáticos, a los cuales pertenecen los modelos de simulación de eventos discretos. Asimismo, los modelos pueden diferenciarse según el tipo de ecuaciones matemáticas que los componen. Por ejemplo, se conoce como **modelos continuos** a aquellos en los que las relaciones entre las variables relevantes de la situación real se definen por medio de ecuaciones diferenciales, dado que éstas permiten conocer el comportamiento de las variables en un lapso de tiempo continuo. Problemas como saber de qué manera se transfiere el calor en un molde o determinar cómo fluye cierto material dentro de una tubería, e incluso discernir el comportamiento del nivel de un tanque de gasolina al paso del tiempo mientras el vehículo está en marcha, pueden simularse en estos términos.

Además de modelos continuos tenemos **modelos discretos**. En ellos el comportamiento que nos interesa analizar puede representarse por medio de ecuaciones evaluadas en un punto determinado. Por ejemplo, si hacemos un muestreo del número de

personas que llegaron a un banco en un lapso de tiempo específico, podemos simular esta variable con ecuaciones ligadas a distribuciones de probabilidad que reflejen dicho comportamiento.

Otro tipo de clasificación es el de los modelos dinámicos o estáticos. Los **modelos dinámicos** son aquellos en los que el estado del sistema que estamos analizando cambia respecto del tiempo. Por ejemplo, el número de personas que hacen fila para entrar a una sala de cine varía con el tiempo. Por otro lado, los **modelos estáticos** representan un resultado bajo un conjunto de situaciones o condiciones determinado; por ejemplo, al lanzar un dado los únicos valores que se puede obtener son 1, 2, 3, 4, 5 o 6, de manera que el resultado de la simulación será uno de tales valores posibles; este tipo de simulación generalmente se conoce como simulación de Monte Carlo.

Por último, podemos hablar de **modelos determinísticos** y **modelos probabilísticos**, conocidos también como **estocásticos.** Los primeros se refieren a relaciones constantes entre los cambios de las variables del modelo. Por ejemplo, si las cajas empleadas en un proceso contienen siempre 5 productos, cada vez que se añada una caja al inventario éste se incrementará en 5 unidades. Si, por el contrario, se da una distribución de probabilidad en el proceso de manera que algunas cajas contienen 3 productos, otras 4 y así por el estilo, el inventario se modificará según el número de piezas de cada caja y, en consecuencia, será necesario un modelo estocástico. En el caso de la simulación de eventos discretos hablaremos de modelos matemáticos, discretos, dinámicos, y que pueden incluir variables determinísticas y probabilísticas.

Ejemplo 1.1

Un taller recibe ciertas piezas, mismas que son acumuladas en un almacén temporal en donde esperan a ser procesadas. Esto ocurre cuando un operario transporta las piezas del almacén a un torno. Desarrolle un modelo que incluya el número de piezas que hay en el almacén esperando a ser atendidas en todo momento, y el número de piezas procesadas en el torno.

En la siguiente figura podemos observar cómo se vería un modelo de simulación para este ejemplo.

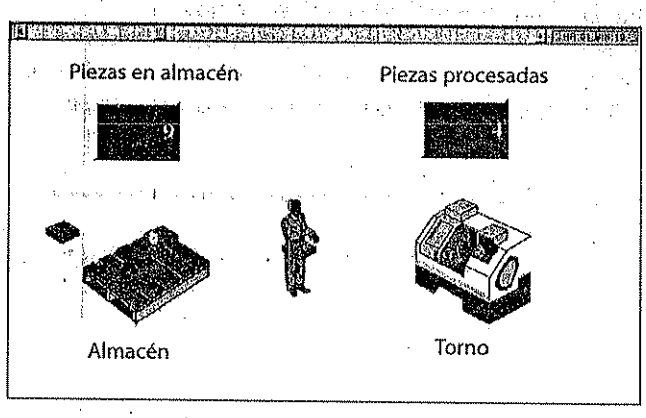


Figura 1.1
Modelo de simulación para el ejemplo 1.1

30

En este ejemplo podemos identificar algunos de los elementos que participan en un modelo de simulación, de acuerdo con las definiciones que hemos comentado:

Sistema: En este caso, el sistema está conformado por el conjunto de elementos interrelacionados para el funcionamiento del proceso: las piezas, el almacén temporal, el operario, el torno.
Entidades: En este modelo sólo tenemos una entidad: las piezas, que representan los flujos de entrada al sistema del problema bajo análisis.
Estado del sistema: Podemos observar que cuando llevamos 1 hora 10 minutos de simulación (vea el extremo superior derecho de la figura) en el almacén se encuentran 9 piezas esperando a ser procesadas; el operario está transportando una pieza más para procesarla en el torno. El torno, por lo tanto, no está trabajando en ese momento, aunque ya ha procesado 4 piezas. Además de estos datos, podemos llevar un control de otras estadísticas relacionadas con el estado del sistema, como el tiempo promedio de permanencia de las piezas en los estantes del almacén temporal o en el sistema global.
Eventos: Entre otros, podríamos considerar como eventos de este sistema el tiempo de descanso del operario o la salida de una pieza tras ser procesada por el torno. Además es posible identificar un evento futuro: la llegada de la siguiente pieza al sistema (tendríamos más eventos de este tipo respecto de las piezas que esperan a que el operario las tome).
Localizaciones: En este caso tenemos el almacén al que deberán llegar las piezas y en el que esperarán a ser procesadas, así como el torno en donde esto ocurrirá.
Recursos: En este modelo, un recurso es el operario que transporta las piezas del almacén al torno.
Atributos: Digamos que (aunque no se menciona en el ejemplo) las piezas pueden ser de tres tamaños diferentes. En este caso, un atributo llamado tamaño podría agregarse a la información de cada pieza que llega al sistema, para posteriormente seleccionar el tipo de operación que deberá realizarse y el tiempo necesario para llevarla a cabo de acuerdo con dicho atributo.
Variables: Tenemos dos variables definidas en este caso: el número de piezas en el almacén y el número de piezas procesadas en el torno.
Reloj de la simulación: Como se puede ver en la esquina superior derecha de la figura 1.1, en este momento la simulación lleva 1 hora 10 minutos. El reloj de la simulación continuará avanzando hasta el momento que se haya establecido para el término de la simulación, o hasta que se cumpla una condición lógica para detenerla, por ejemplo, el número de piezas que se desean simular.

Otro concepto importante que vale la pena definir es el de **réplica** o **corrida** de la simulación. Cuando ejecutamos el modelo en una ocasión, los valores que obtenemos de las variables y parámetros al final del tiempo de simulación generalmente serán distintos de los que se producirán si lo volvemos a correr usando diferentes números pseudo aleatorios. Por lo tanto, es necesario efectuar más de una réplica del modelo que se esté analizando, con la finalidad de obtener estadísticas de intervalo que nos den una mejor ubicación del verdadero valor de la variable bajo los diferentes escenarios que se presentan al modificar los números pseudo aleatorios en cada oportunidad.

En este sentido, la pregunta clave es cuánto tiempo se debe simular un modelo para obtener resultados confiables. En general, podemos decir que todas las variables que se obtienen en términos de promedios presentan dos diferentes etapas: un **estado transitorio** y un **estado estable**. El primero se presenta al principio de la simulación; por ejemplo, en el arranque de una planta, cuando no tiene material en proceso: el último de los procesos estará inactivo hasta que el primer cliente llegue, y si el tiempo de simulación es bajo, su impacto sobre la utilización promedio de este proceso será muy alto, lo cual no ocurriría si el modelo se simulara lo suficiente para lograr una compensación. En el estado transitorio hay mucha variación entre los valores promedio de las variables de decisión del modelo, por lo que formular conclusiones con base en ellos sería muy arriesgado, toda vez que difícilmente nos darían una representación fiel de la realidad.

Por otro lado, en el **estado estable** los valores de las variables de decisión permanecen muy estables, presentando sólo variaciones poco significativas. En este momento las decisiones que se tomen serán mucho más confiables. Sin embargo no todas las variables convergen al estado estable con la misma rapidez: algunas pasan con más lentitud que otras de un estado transitorio a un estado estable. Es responsabilidad del analista verificar que las variables de decisión del modelo se encuentren en estado estable antes de detener el tiempo de la simulación.

Otro factor importante para decidir el tiempo de simulación es el costo de la corrida. Mayor tiempo de simulación requiere más tiempo computacional, lo cual implica, necesariamente, un costo más alto. Por supuesto, la situación empeora si a esto le agregamos que en algunos casos es necesario efectuar más de tres réplicas.

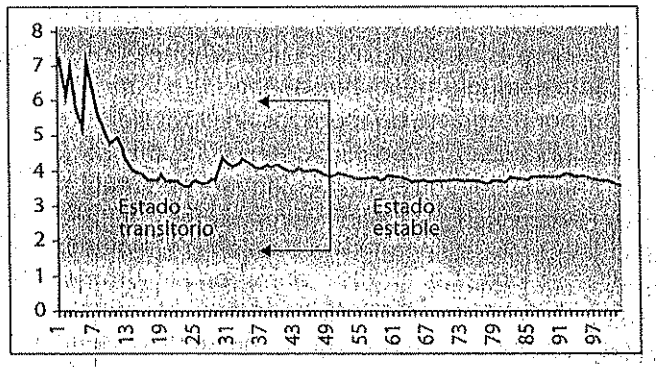


Figura 1.2
Gráfica de estabilización de una variable

1.3 Ventajas y desventajas de la simulación

Como hemos visto hasta ahora, la simulación es una de las diversas herramientas con las que cuenta el analista para tomar decisiones y mejorar sus procesos. Sin embargo, es necesario destacar que, como todas las demás opciones de que disponemos, la simulación de eventos discretos presenta ventajas y desventajas que es preciso tomar en cuenta al determinar si es apta para resolver un problema determinado.

32

Dentro de las ventajas más comunes que ofrece la simulación podemos citar las siguientes:

a) Es muy buena herramienta para conocer el impacto de los cambios en los procesos sin necesidad de llevarlos a cabo en la realidad.
b) Mejora el conocimiento del proceso actual al permitir que el analista vea cómo se comporta el modelo generado bajo diferentes escenarios.
c) Puede utilizarse como medio de capacitación para la toma de decisiones.
d) Es más económico realizar un estudio de simulación que hacer muchos cambios en los procesos reales.
e) Permite probar varios escenarios en busca de las mejores condiciones de trabajo de los procesos que se simulan.
f) En problemas de gran complejidad, la simulación permite generar una buena solución.
g) En la actualidad los paquetes de software para simulación tienden a ser más sencillos, lo que facilita su aplicación.
h) Gracias a las herramientas de animación que forman parte de muchos de esos paquetes es posible ver cómo se comportará un proceso una vez que sea mejorado.

Entre las desventajas que puede llegar a presentar la simulación están:

a) Aunque muchos paquetes de software permiten obtener el mejor escenario a partir de una combinación de variaciones posibles, la simulación *no* es una herramienta de optimización.
b) La simulación puede ser costosa cuando se quiere emplearla en problemas relativamente sencillos de resolver, en lugar de utilizar soluciones analíticas que se han desarrollado de manera específica para ese tipo de casos.
c) Se requiere bastante tiempo —generalmente meses— para realizar un buen estudio de simulación; por desgracia, no todos los analistas tienen la disposición (o la oportunidad) de esperar ese tiempo para obtener una respuesta.
d) Es preciso que el analista domine el uso del paquete de simulación y que tenga sólidos conocimientos de estadística para interpretar los resultados.

1.4 Elementos clave para garantizar el éxito de un modelo de simulación

Independientemente de los beneficios que conlleva la simulación, es imposible garantizar que un modelo tendrá éxito. Existen ciertas condiciones clave que pueden traer problemas si no se les pone atención al momento de usar la simulación para la toma de decisiones. A continuación destacaremos algunas de las causas por las que un modelo de simulación podría no tener los resultados que se desean:

- **Tamaño insuficiente de la corrida.** Como se mencionó antes, para poder llegar a conclusiones estadísticas válidas a partir de los modelos de simulación es necesario que las variables aleatorias de respuesta estén en estado estable. El problema estriba en que, ge-

neralmente, cuando el modelo consta de más de una variable de decisión, es difícil que éstas alcancen un estado estable al mismo tiempo: es posible que una se encuentre estable y la otra no en un momento determinado, por lo que las conclusiones respecto de la segunda variable no serán estadísticamente confiables.

Variable(s) de respuesta mal definida(s). Aun cuando el modelo de simulación sea muy eficiente y represente la realidad en gran medida, si la variable de respuesta seleccionada no es la apropiada será imposible tomar decisiones que tengan impacto en la operación del sistema bajo estudio.

Por ejemplo, digamos que una variable de respuesta es el nivel de inventarios de cierto producto. Al mismo tiempo, la política de la empresa establece que no se debe parar ninguno de los procesos de fabricación. En consecuencia, el problema no será el inventario final, sino el ritmo de producción necesario para que aquel cumpla con los requerimientos de diseño que se desean.

Errores al establecer las relaciones entre las variables aleatorias. Un error común de programación es olvidar las relaciones lógicas que existen entre las variables aleatorias del modelo, o minimizar su impacto. Si una de estas variables no está definida de manera correcta, ciertamente aún es posible tener un modelo que se apegue a la realidad actual; sin embargo, si el sistema no se lleva hasta su máxima capacidad para observar su comportamiento, podría resultar imposible visualizar el verdadero impacto de las deficiencias.

Errores al determinar el tipo de distribución asociado a las variables aleatorias del modelo. Este tipo de problema es muy similar al anterior, sólo que en este caso se utilizan distribuciones que no son las más adecuadas o que responden únicamente a un intento de simplificar los estudios estadísticos. Digamos, por ejemplo, que se nos dan los siguientes parámetros de producción aproximados: mínimo 10, máximo 40 y promedio 30. En esta circunstancia la tentación de simplificar el estudio de la variable asignándole una distribución triangular con parámetros (10, 30, 40) es muy grande; no obstante, hacerlo afectaría de manera importante los resultados de la simulación, pues el modelo podría alejarse de lo que sucede en la realidad.

Falta de un análisis estadístico de los resultados. Un problema común por el que la simulación suele ser objeto de crítica, radica en asumir que se trata de una herramienta de optimización. Esta apreciación es incorrecta, ya que involucra variables aleatorias y características propias de un modelo que incluye probabilidades. Por lo mismo —como se apuntó antes—, es necesario realizar varias corridas a fin de producir diferentes resultados finales para las variables de respuesta y, a partir de esos valores, obtener intervalos de confianza que puedan dar un rango en dónde encontrar los valores definitivos. Este tipo de problemas se presentan también al comparar dos escenarios: podríamos encontrar un mejor resultado para uno de ellos, pero si los intervalos de confianza de las variables de respuesta se traslapan resultaría imposible decir que el resultado de un escenario es mejor que el del otro. De hecho, estadísticamente hablando ambos resultados pueden ser iguales. En ese caso incrementar el tamaño de corrida o el número de réplicas puede ayudar a obtener mejores conclusiones.

Uso incorrecto de la información obtenida. Un problema que se presenta en ocasiones es el uso incorrecto de la información recabada para la realización del estudio, ya sea a través de un cliente o de cualesquiera otras fuentes. Muchas veces esta información se recolecta, analiza y administra de acuerdo con las necesidades propias de la empresa, lo

que implica que no siempre está en el formato y la presentación que se requiere para la simulación. Si la información se utiliza para determinar los parámetros del modelo sin ser depurada y reorganizada, es muy probable que la precisión de los resultados del estudio se vea afectada.

Falta o exceso de detalle en el modelo. Otro punto importante a considerar es el nivel de detalle del modelo. En muchas ocasiones algún proceso se simplifica tanto que tiende a verse como una "caja negra" que nos impide ver qué ocurre en el interior, aunque sí haya entrada y salida de datos que interactúan con otras partes del modelo. Cuando esto sucede, el impacto que podrían tener los subprocesos que se llevan a cabo en la "caja negra" (es decir, del proceso sobresimplificado) no se incluye en la simulación. Por ejemplo, si se analiza un sistema de distribución y se da por sentado que el almacén *siempre* surte sus pedidos, no incluiremos el impacto de los tiempos necesarios para surtir las órdenes, ni la posibilidad de que haya faltantes de producto; excluiremos también los horarios de comida, en los que no se surten pedidos, y las fallas en los montacargas que transportan los pedidos hasta los camiones para su distribución. Por otra parte, si el modelo se hace demasiado detallado, tanto el tiempo dedicado al estudio como el costo de llevarlo a cabo podrían incrementarse sustancialmente. Es labor del encargado de la simulación sugerir y clarificar los niveles de detalle que se requieren en el modelo, resaltando los alcances y limitaciones de cada uno.

1.5 Pasos para realizar un estudio de simulación

→ cumplir en el TP

Debemos considerar que —igual a como ocurre con otras herramientas de investigación— la realización de un estudio de simulación requiere la ejecución de una serie de actividades y análisis que permitan sacarle el mejor provecho. A continuación se mencionan los pasos básicos para realizar un estudio de simulación, aunque en muchas ocasiones será necesario agregar otros o suprimir algunos de los aquí enumerados, de acuerdo con la problemática en cuestión.

1. Definición del sistema bajo estudio. En esta etapa es necesario conocer el sistema a modelar. Para ello se requiere saber qué origina el estudio de simulación y establecer los supuestos del modelo: es conveniente definir con claridad las variables de decisión del modelo, determinar las interacciones entre éstas y establecer con precisión los alcances y limitaciones que aquel podría llegar a tener.

Antes de concluir este paso es recomendable contar con la información suficiente para lograr establecer un modelo conceptual del sistema bajo estudio, incluyendo sus fronteras y todos los elementos que lo componen, además de las interacciones entre éstos, flujos de productos, personas y recursos, así como las variables de mayor interés para el problema.

2. Generación del modelo de simulación base. Una vez que se ha definido el sistema en términos de un modelo conceptual, la siguiente etapa del estudio consiste en la generación de un modelo de simulación base. No es preciso que este modelo sea demasiado detallado, pues se requiere mucha más información estadística sobre el comportamiento de las variables de decisión del sistema. La generación de este modelo es el primer reto para el programador de la simulación, toda vez que debe traducir a un lenguaje de simulación

la información que se obtuvo en la etapa de definición del sistema, incluyendo las interrelaciones de todos los posibles subsistemas que existan en el problema a modelar. En caso de que se requiera una animación, éste también es un buen momento para definir qué gráfico puede representar mejor el sistema que se modela.

Igual que ocurre en otras ramas de la investigación de operaciones, la simulación exige ciencia y arte en la generación de sus modelos. El realizador de un estudio de simulación es, en este sentido, como un artista que debe usar toda su creatividad para realizar un buen modelo que refleje la realidad del problema que se está analizando. Conforme se avanza en el modelo base se pueden ir incluyendo las variables aleatorias del sistema, con sus respectivas distribuciones de probabilidad asociadas.

3. Recolección y análisis de datos. De manera paralela a la generación del modelo base, es posible comenzar la recopilación de la información estadística de las variables aleatorias del modelo. En esta etapa se debe determinar qué información es útil para la determinación de las distribuciones de probabilidad asociadas a cada una de las variables aleatorias innecesarias para la simulación. Aunque en algunos casos se logra contar con datos estadísticos, suele suceder que el formato de almacenamiento o de generación de reportes no es el apropiado para facilitar el estudio. Por ello es muy importante dedicar el tiempo suficiente a esta actividad. De no contar con la información necesaria o en caso de desconfiar de la que se tiene disponible, será necesario realizar un estudio estadístico del comportamiento de la variable que se desea identificar, para posteriormente incluirla en el modelo. El análisis de los datos necesarios para asociar una distribución de probabilidad a una variable aleatoria, así como las pruebas que se debe aplicar a los mismos, se analizarán más adelante. Al finalizar la recolección y análisis de datos para todas las variables del modelo, se tendrán las condiciones necesarias para generar una versión preliminar del problema que se está simulando.

4. Generación del modelo preliminar. En esta etapa se integra la información obtenida a partir del análisis de los datos, los supuestos del modelo y todos los datos que se requieran para tener un modelo lo más cercano posible a la realidad del problema bajo estudio. En algunos casos —sobre todo cuando se trata del diseño de un nuevo proceso o esquema de trabajo— no se cuenta con información estadística, por lo que debe estimarse un rango de variación o determinar (con ayuda del cliente) valores constantes que permitan realizar el modelado. Si éste es el caso, el encargado de la simulación puede, con base en su experiencia, realizar algunas sugerencias de distribuciones de probabilidad que comúnmente se asocien al tipo de proceso que se desea incluir en el modelo. Al finalizar esta etapa el modelo está listo para su primera prueba: su verificación o, en otras palabras, la comparación con la realidad.

5. Verificación del modelo. Una vez que se han identificado las distribuciones de probabilidad de las variables del modelo y se han implantado los supuestos acordados, es necesario realizar un proceso de verificación de datos para comprobar la propiedad de la programación del modelo, y comprobar que todos los parámetros usados en la simulación funcionen correctamente. Ciertos problemas, en especial aquellos que requieren muchas operaciones de programación o que involucran distribuciones de probabilidad difíciles de programar, pueden ocasionar que el comportamiento del sistema sea muy diferente del que se esperaba. Por otro lado, no se debe descartar la posibilidad de que ocurran errores humanos al alimentar el modelo con la información. Incluso podría darse el

Possible question Difference between 5. and 6

36

caso de que los supuestos iniciales hayan cambiado una o varias veces durante el desarrollo del modelo. Por lo tanto, debemos asegurarnos de que el modelo que se va a ejecutar esté basado en los más actuales.

Una vez que se ha completado la verificación, el modelo está listo para su comparación con la realidad del problema que se está modelando. A esta etapa se le conoce también como validación del modelo.

6. Validación del modelo. El proceso de validación del modelo consiste en realizar una serie de pruebas al mismo, utilizando información de entrada real para observar su comportamiento y analizar sus resultados.

Si el problema bajo simulación involucra un proceso que se desea mejorar, el modelo debe someterse a prueba con las condiciones actuales de operación, lo que nos dará como resultado un comportamiento similar al que se presenta realmente en nuestro proceso. Por otro lado, si se está diseñando un nuevo proceso la validación resulta más complicada. Una manera de validar el modelo en este caso, consiste en introducir algunos escenarios sugeridos por el cliente y validar que el comportamiento sea congruente con las expectativas que se tienen de acuerdo con la experiencia. Cualquiera que sea la situación, es importante que el analista conozca bien el modelo, de manera que pueda justificar aquellos comportamientos que sean contrarios a las experiencias de los especialistas en el proceso que participan de su validación.

7. Generación del modelo final. Una vez que el modelo se ha validado, el analista está listo para realizar la simulación y estudiar el comportamiento del proceso. En caso de que se desee comparar escenarios diferentes para un mismo problema, éste será el *modelo raíz*; en tal situación, el siguiente paso es la definición de los escenarios a analizar.

8. Determinación de los escenarios para el análisis. Tras validar el modelo es necesario acordar con el cliente los escenarios que se quiere analizar. Una manera muy sencilla de determinarlos consiste en utilizar un escenario pesimista, uno optimista y uno intermedio para la variable de respuesta más importante. Sin embargo, es preciso tomar en cuenta que no todas las variables se comportan igual ante los cambios en los distintos escenarios, por lo que tal vez sea necesario que más de una variable de respuesta se analice bajo las perspectivas pesimista, optimista e intermedia. El riesgo de esta situación radica en que el analista podría caer en un diseño de experimentos capaz de generar una gran cantidad de réplicas, lo que redundaría en un incremento considerable de costo, análisis y tiempo de simulación. Es por ello que muchos paquetes de simulación cuentan con herramientas para realizar este proceso, eliminando la animación y acortando los tiempos de simulación. Estas herramientas permiten realizar varias réplicas del mismo escenario para obtener resultados con estadísticas importantes respecto de la toma de decisiones (por ejemplo, los intervalos de confianza).

Por su parte, el analista también puede contribuir a la selección de escenarios, sugiriendo aquellos que considere más importantes; al hacerlo dará pie a que se reduzca el número de combinaciones posibles.

9. Análisis de sensibilidad. Una vez que se obtienen los resultados de los escenarios es importante realizar pruebas estadísticas que permitan comparar los escenarios con los mejores resultados finales. Si dos de ellos tienen resultados similares será necesario comparar sus intervalos de confianza respecto de la variable de respuesta final. Si no hay intersección de intervalos podremos decir con certeza estadística que los resultados no son

iguales; sin embargo, si los intervalos se traslapan será imposible determinar, estadísticamente hablando, que una solución es mejor que otra. Si se desea obtener un escenario "ganador" en estos casos, será necesario realizar más réplicas de cada modelo y/o incrementar el tiempo de simulación de cada corrida. Con ello se busca acortar los intervalos de confianza de las soluciones finales y, por consiguiente, incrementar la probabilidad de diferenciar las soluciones.

10. Documentación del modelo, sugerencias y conclusiones. Una vez realizado el análisis de los resultados, es necesario efectuar toda la documentación del modelo.

Esta documentación es muy importante, pues permitirá el uso del modelo generado en caso de que se requieran ajustes futuros. En ella se deben incluir los supuestos del modelo, las distribuciones asociadas a sus variables, todos sus alcances y limitaciones y, en general, la totalidad de las consideraciones de programación. También es importante incluir sugerencias tanto del uso del modelo como sobre los resultados obtenidos, con el propósito de realizar un reporte más completo. Por último, deberán presentarse asimismo las conclusiones del proyecto de simulación, a partir de las cuales es posible obtener los reportes ejecutivos para la presentación final.

En la figura 1.3 se presenta una gráfica de Gantt en donde se muestra, a manera de ejemplo, la planificación de los pasos para realizar una simulación que hemos comentado en esta sección.

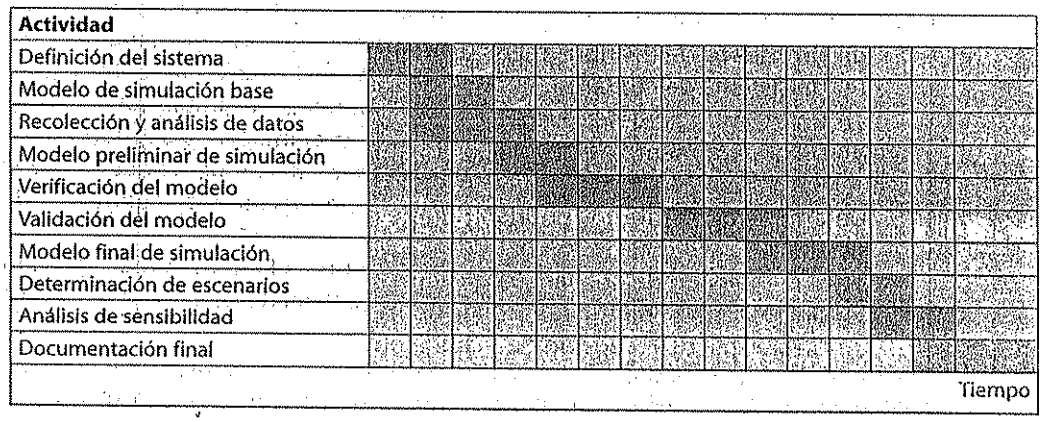


Figura 1.3 Gráfica de Gantt de un proyecto de simulación

1.6 Problemas

1. Determine los elementos de cada uno de los siguientes sistemas, de acuerdo con lo que se comentó en la sección 1.2.
   a) La sala de emergencia de un hospital.
   b) Un banco mercantil.

CAPÍTULO

13

MODELOS DE COLAS

National Public TV está llevando a cabo un teletón con el fin de recabar fondos. El apoyo corporativo casi se ha agotado, y la red necesita abrir una fuente más amplia de ingresos de los espectadores privados. La red tiene compromisos de voluntarios para encargarse de las líneas telefónicas durante el esfuerzo de cinco días, que es el tiempo que dura el evento, pero la administración aún debe decidir qué sistema telefónico es la mejor alternativa. ¿Cuántas líneas de teléfono debe alquilar la red? La administración no desea gastar el dinero que necesita desesperadamente en la programación de un número innecesario de líneas telefónicas. Sin embargo, la red no puede darse el lujo de que las personas que llaman para ofrecer dinero encuentren que la línea está ocupada o se les ponga a esperar tanto tiempo que cuelguen.

Este capítulo le proporciona las herramientas que usted necesita para analizar la situación de National Public TV, que pertenece a una categoría de problemas conocidos como modelos de colas.

Muchas industrias de productos y de servicios tienen un sistema de colas, en el que los "productos" (o clientes) llegan a una "estación" esperan en una "fila" (o cola), obtienen algún tipo de "servicio" y luego salen del sistema. Considere los siguientes ejemplos:

Sistema de colas Sistema en el que los productos (o los clientes) llegan a una estación, esperan en una fila (o cola), obtienen algún tipo de servicio y luego salen del sistema.

- Los clientes llegan a un banco, esperan en una fila para obtener un servicio de uno de los cajeros, y después salen del banco.
- Las partes de un proceso de producción llegan a una estación de trabajo particular desde diferentes estaciones, esperan en un compartimiento para ser procesadas por una máquina, y luego son enviadas a otra estación de trabajo.
- Después de hacer sus compras, los clientes eligen una fila en las cajas, esperan a que el cajero les cobre y luego salen de la tienda.
- Las llamadas telefónicas llegan a un centro de reservaciones de una aerolínea, esperan al agente de ventas disponible, son atendidas por ese agente y dejan el sistema cuando el cliente cuelga.

Los problemas administrativos relacionados con tales sistemas de colas se clasifican en dos grupos básicos:

1. *Problemas de análisis.* Usted podría estar interesado en saber si un sistema *dado* está funcionando satisfactoriamente. Necesita responder una o más de las siguientes preguntas:

   a. ¿Cuál es el tiempo promedio que un cliente tiene que esperar en la fila antes de ser atendido?
   b. ¿Qué fracción del tiempo ocupan los servidores en atender a un cliente o en procesar un producto?
   c. ¿Cuáles son el número promedio y el máximo de clientes que esperan en la fila?

   Basándose en estas preguntas, los gerentes tomarán decisiones como emplear o no a más gente; agregar una estación de trabajo adicional para mejorar el nivel de servicio; o si es necesario o no aumentar el tamaño del área de espera.

2. *Problemas de diseño.* Usted desea diseñar las características de un sistema que logre un objetivo general. Esto puede implicar el planteamiento de preguntas como las siguientes:

   a. ¿Cuántas personas o estaciones deben emplearse para proporcionar un servicio aceptable?
   b. ¿Deberán los clientes esperar en una sola fila (como se hace en muchos bancos) o en diferentes filas (como en el caso de los supermercados)?
   c. ¿Deberá haber una estación de trabajo separada que maneje las cuestiones "especiales" (como el caso del acceso a primera clase en el mostrador de una aerolínea?
   d. ¿Qué tanto espacio se necesita para que los clientes o los productos puedan esperar? Por ejemplo, en un sistema de reservaciones por teléfono, ¿qué tan grande debe ser la capacidad de retención? Esto es, ¿cuántas llamadas telefónicas se deben mantener en espera antes de que la siguiente obtenga la señal de ocupado?

Estas decisiones de diseño se toman mediante la evaluación de los méritos de las diferentes alternativas, respondiendo a las preguntas de análisis del grupo 1 y luego seleccionando la alternativa que cumpla con los objetivos administrativos.

39

En el presente capítulo se proporcionan las técnicas para analizar un sistema de colas dado. Sin embargo, las técnicas matemáticas específicas dependen de la *clase* de sistema al cual pertenece su problema de colas. Estas clases, basadas en las características de los diferentes componentes del sistema, se presentan en la sección 13.1. En la sección 13.2, se describen varias medidas utilizadas para evaluar el desempeño de tales sistemas.

Población de clientes
Conjunto de todos los clientes posibles de un sistema de colas.

Proceso de llegada
La forma en que los clientes de la población llegan a solicitar un servicio.

Proceso de colas
La forma en que los clientes esperan a que se les dé un servicio.

Disciplina de colas
La forma en que los clientes son elegidos para proporcionarles un servicio.

Proceso de servicio
Forma y rapidez con que son atendidos los clientes.

Proceso de salida
Forma en que los productos o los clientes abandonan un sistema de colas.

Sistema de colas de un paso
Sistema en el cual los productos o los clientes abandonan el sistema después de ser atendidos en un solo centro o estación de trabajo.

Red de colas
Sistema en el que un producto puede proceder de una estación de trabajo y pasar a otra antes de abandonar el sistema.

13.1 CARACTERÍSTICAS DE UN SISTEMA DE COLAS

Para analizar un sistema de colas, es mejor primero identificar las características importantes que aparecen en la siguiente sección de características clave, y que se ilustran en la figura 13.1.

CARACTERÍSTICAS CLAVE

Las siguientes características se aplican a los sistemas de colas:

- ✓ **Una población de clientes**, que es el conjunto de todos los clientes posibles.
- ✓ **Un proceso de llegada**, que es la forma en que llegan los clientes de esa población.
- ✓ **Un proceso de colas**, que está conformado por (a) la manera en que los clientes esperan para ser atendidos y (b) la disciplina de colas, que es la forma en que son elegidos para proporcionarles el servicio.
- ✓ **Un proceso de servicio**, que es la forma y la rapidez con la que es atendido el cliente.
- ✓ **Procesos de salida**, que son de los siguientes dos tipos:
  a. Los elementos abandonan completamente el sistema después de ser atendidos, lo que tiene como resultado un sistema de colas de un paso. Por ejemplo, como se muestra en la figura 13.2(a), los clientes de un banco esperan en una sola fila, son atendidos por uno de los tres cajeros y, después de que son atendidos, abandonan el sistema.
  b. Los productos, ya que son procesados en una estación de trabajo, son trasladados a alguna otra para someterlos a otro tipo de proceso, lo que tiene como resultado una red de colas. Por ejemplo, los productos que se muestran en la figura 13.2(b) primero son procesados en la estación de trabajo A y después enviados a la estación B o C. Los productos terminados en ambas estaciones, B y C, luego son procesados en la estación D, antes de abandonar el sistema.

Se necesitan diferentes análisis matemáticos para cada uno de estos dos tipos de procesos de salida. En el presente capítulo solamente se considerarán sistema de un paso.

El análisis de un sistema de colas de un paso depende de las características precisas de los primeros cuatro componentes, que se analizarán con detalle a continuación.

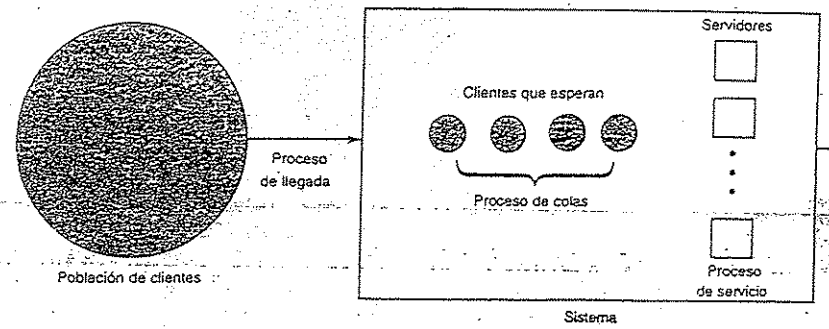


Figura 13.1 Componentes de un sistema de colas.

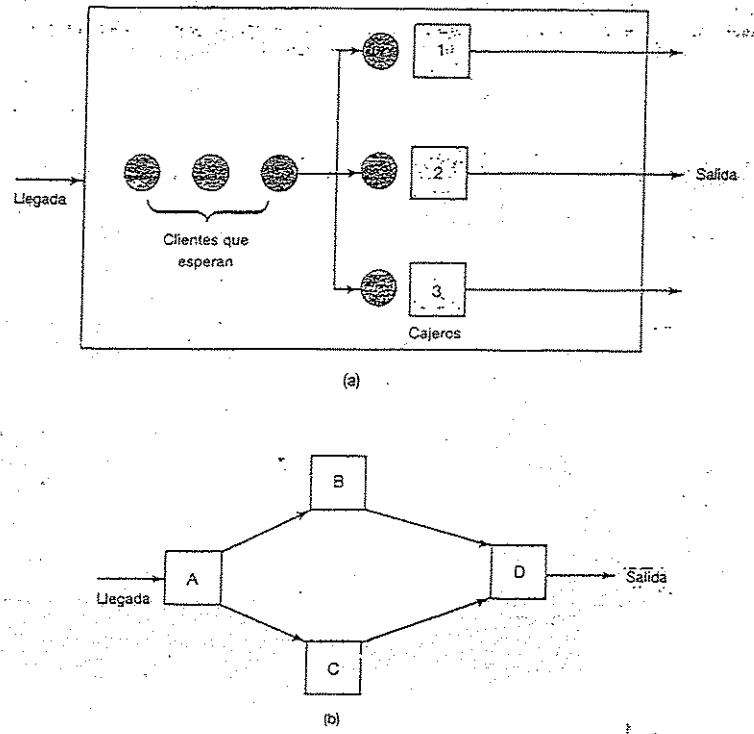


Figura 13.2 Proceso de salida de un sistema de colas.

40

13.1.1 La población de clientes

Al tomar en cuenta la base de clientes, la principal preocupación es el *tamaño* de la población. Para problemas como los de un banco o de un supermercado, en donde el número de clientes potenciales es bastante grande (cientos o miles), el tamaño de la población se considera, para fines prácticos, como si fuera *infinita*.

Al contrario, considere una fábrica que tiene cuatro máquinas, que a menudo se descomponen y requieren servicio de reparación en un taller especializado. En este caso, las máquinas están en lugar de los clientes y el taller es el centro de servicio. El tamaño de la población de clientes, en este caso, es de solamente cuatro. El análisis de poblaciones *finitas* (es decir, de tamaño limitado) es más complicado que el análisis en donde la base de población se considera infinita.

13.1.2 El proceso de llegada

Tiempo entre llegadas
Intervalo de tiempo que existe entre dos llegas sucesivas de clientes a un sistema de colas.

El proceso de llegada es la forma en que los clientes llegan a solicitar un servicio. La característica más importante del proceso de llegada es el tiempo entre llegadas, que es la cantidad de tiempo entre dos llegadas sucesivas. Este lapso es importante porque mientras menor sea el intervalo de tiempo, con más frecuencia llegan los clientes, lo cual aumenta la demanda de servidores disponibles.

CARACTERÍSTICAS CLAVE

Existen dos clases básicas de tiempos entre llegadas:

- ✓ *Determinístico*, en el cual clientes sucesivos llegan en un mismo intervalo de tiempo, fijo y conocido. Un ejemplo clásico es el caso de una línea de ensamblaje, en donde los artículos llegan a una estación en intervalos invariables de tiempo (conocidos como *ciclos de tiempo*).
- ✓ *Probabilístico*, en el cual el tiempo entre llegadas sucesivas es incierto y variable. Los tiempos entre llegadas probabilísticos se describen mediante una distribución de probabilidad.

En el caso probabilístico, la determinación de la distribución real, a menudo, resulta difícil. Sin embargo, una distribución, la *distribución exponencial*, ha probado ser confiable en muchos problemas prácticos. La función de densidad para una distribución exponencial depende de un parámetro, digamos λ (la letra griega lambda), y está dada por:

$$f(t) = (1/\lambda)e^{-t/T}$$

en donde λ (lambda) es el número promedio de llegadas por unidad de tiempo.

Con una cantidad, T, de tiempo, usted puede hacer uso de la función de densidad para calcular la probabilidad de que el siguiente cliente llegue dentro de las siguientes T unidades a partir de la llegada anterior, de la manera siguiente:

$$P\text{ (tiempo entre llegadas} \le T) = 1 - e^{-\lambda T}$$

Por ejemplo, si los clientes llegan al banco con una rapidez promedio de $\lambda = 20$ por hora y si un cliente acaba de llegar, entonces la probabilidad de que el siguiente llegue dentro de los siguientes diez minutos (es decir $Y = 1/6$ de hora) es:

$$\begin{aligned} P\text{ (tiempo entre llegadas} \le 1/6 \text{ hora)} &= 1 - e^{-20*(1/6)} \\ &= 1 - e^{-3.3333} \\ &= 1 - 0.036 \\ &= 0.964 \end{aligned}$$

Otro planteamiento igualmente válido para describir el proceso de llegadas consiste en utilizar la distribución de probabilidad del *número de llegadas*. Por ejemplo, usted podría estar interesado en la probabilidad de que dos clientes lleguen dentro de los diez minutos siguientes. Cuando la distribución de tiempos entre llegadas es una función exponencial con parámetro λ, la distribución de probabilidad para el número de llegadas se conoce como **distribución de Poisson** y está dada por:

$$P\text{ (tiempo entre llegadas } T = k) = \frac{e^{-\lambda * T}(\lambda * T)^k}{k!}$$

en la que $k! = k(k-1)\ldots(2)(1)$.

Por ejemplo, cuando $\lambda = 20$ clientes por hora y $T = 1/6$ de hora, la probabilidad de que lleguen $k = 2$ clientes en los siguientes diez minutos es:

$$\begin{aligned} P\text{ (tiempo de llegadas en 10 minutos} = 2) &= \frac{e^{-(20)(1/6)}(20/6)^2}{2!} \\ &= \frac{0.036 * 11.111}{2} \\ &= 0.20 \end{aligned}$$

En este caso, el proceso de llegadas se conoce como **proceso de Poisson**, pero en general, un proceso de llegadas puede obedecer a cualquier otra distribución.

Distribución de Poisson
Distribución que describe la probabilidad de que se presenten un número dado de llegadas en un intervalo dado de tiempo, cuando el tiempo entre llegadas sigue una distribución exponencial.

Proceso de Poisson
Proceso aleatorio en que el tiempo entre llegadas sucesivas sigue una distribución exponencial.

13.1.3 El proceso de colas

Parte del proceso de colas tiene que ver con la forma en que los clientes esperan para ser atendidos. Los clientes pueden esperar en una sola fila, como en un banco; observe la figura 13.3(a), éste es un **sistema de colas de una sola línea**. Al contrario, los clientes pueden elegir una de varias filas en la que deben esperar a ser atendidos, como en las cajas cobradoras de un supermercado; observe la figura 13.3(b), éste es un **sistema de colas de líneas múltiples**.

Otra característica del proceso de colas es el número de espacios de espera en cada fila, es decir, el número de clientes que pueden esperar (o que esperarán) para ser atendidos en cada línea. En algunos casos, como en un banco, ese número es bastante grande y no significa ningún problema práctico, pues para cuestiones de análisis la cantidad de espacio de espera se considera *infinita*. En contraste, un sistema telefónico puede mantener solamente un número *finito* (es decir limitado) de llamadas, después del cual las llamadas subsecuentes no tienen acceso al sistema. Las condiciones de espacio de espera infinito y finito requieren análisis matemáticos diferentes.

Sistema de colas de una sola línea
Sistema de colas en el cual los clientes esperan en una sola línea para tener acceso al siguiente prestador de servicio disponible.

Sistema de colas de líneas múltiples
Sistema de colas en el cual los clientes que llegan pueden elegir una de varias líneas en la cual esperar a ser atendidos.

41

CARACTERÍSTICAS CLAVE

Otra característica del proceso de colas es la *disciplina de colas*, es decir, la forma en que los clientes que esperan son seleccionados para ser atendidos. A continuación presentamos algunas de las formas más comunes.

✓ **Primero en entrar, primero en salir (PEPS).** Los clientes son atendidos en el orden en que van llegando a la fila. Los clientes de un banco y de un supermercado, por ejemplo, son atendidos de esta manera.

✓ **Último en entrar, primero en salir (VEPS).** El cliente que ha llegado más recientemente es el primero en ser atendido. Un ejemplo de esta disciplina se da en un proceso de producción en el que los productos llegan a una estación de trabajo y son apilados uno encima del otro. El trabajador elige, para su procesamiento, el producto que está en la cima de la pila, que fue el último que llegó para ser procesado o para brindarle un servicio.

✓ **Selección de prioridad.** A cada cliente que llega se le da una prioridad y se le elige según ésta para brindarle el servicio. Un ejemplo de esta disciplina son los pacientes que llegan a la sala de urgencias de un hospital. Mientras más severo sea el caso, mayor será la prioridad del "cliente".

Primero en entrar, primero en salir (PEPS)
Disciplina de colas en la que los clientes son atendidos en el orden en que van llegando.

Último en entrar, primero en salir (VEPS)
Disciplina de colas en la que el cliente que ha llegado más recientemente es el primero en ser atendido.

Selección de prioridad
Proceso de llegadas en el que a cada cliente se le da una prioridad y de acuerdo a ésta es seleccionado para el servicio.

En el presente capítulo, sólo se analizará la selección PEPS, que es la disciplina de colas más comúnmente utilizada.

13.1.4 El proceso de servicio

El proceso de servicio define cómo son atendidos los clientes. En algunos casos, puede existir más de una estación en el sistema en la cual se proporcione el servicio requerido. Los bancos y los supermercados, de nuevo, son buenos ejemplos de lo anterior. Cada ventanilla y cada caja registradora son estaciones que proporcionan el mismo servicio. A tales estructuras se les conoce como sistemas de colas de canal múltiple. En dichos sistemas, los servidores pueden ser *idénticos*, en el sentido de que proporcionan la misma clase de servicio con igual rapidez, o pueden ser *no idénticos*. Por ejemplo, si todos los cajeros de un banco tienen la misma experiencia, pueden considerarse como idénticos. En este capítulo, se tomarán en cuenta solamente servidores idénticos.

Al contrario de un sistema de canal múltiple, considere un proceso de producción con una estación de trabajo que proporciona el servicio requerido. Todos los productos deben pasar por esa estación de trabajo; en este caso se trata de un sistema de colas de canal sencillo. Es importante hacer notar que incluso en un sistema de canal sencillo pueden existir muchos servidores que, *juntos*, llevan a cabo la tarea necesaria. Por ejemplo, un negocio de lavado a mano de automóviles, que es una sola estación, puede tener dos empleados que trabajan en un auto de manera simultánea.

Otra característica del proceso de servicio es el número de clientes atendidos al mismo tiempo en una estación. En los bancos y en los supermercados (sistemas de canal múltiple), y en el negocio de lavado de automóviles (sistema de canal sencillo), solamente un cliente es atendido a la vez. Por el contrario, los pasajeros que esperan en una parada de autobús son atendidos en grupo, según la capacidad del autobús que llegue. En el presente capítulo solamente se verá el servicio de uno a la vez.

Sistema de colas de canal múltiple
Sistema en el cual los clientes que llegan pueden pasar a una de varias estaciones de trabajo posibles.

Sistema de colas de canal sencillo
Sistema en el cual los clientes que llegan pasan por una estación de trabajo.

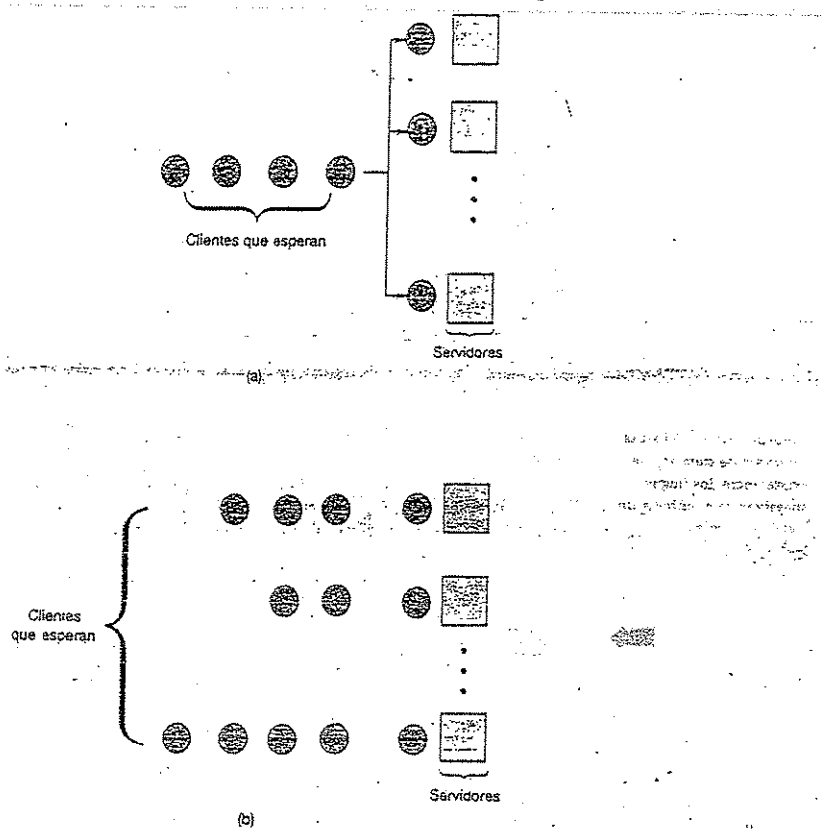


Figura 13.3 Sistemas de colas de (a) una sola fila y (b) múltiples filas.

Otra característica más de un proceso de servicio es si se permite o no la prioridad, esto es, ¿puede un servidor detener el proceso con el cliente que está atendiendo para dar lugar a un cliente que acaba de llegar? Por ejemplo, en una sala de urgencias, la prioridad se presenta cuando un médico, que está atendiendo un caso que no es crítico, es llamado a atender un caso más crítico. En este capítulo, los modelos a analizar no permiten la prioridad.

Cualquiera que sea el proceso de servicio, es necesario tener una idea de cuánto tiempo se requiere para llevar a cabo el servicio. Esta cantidad es importante debido a que cuanto más dure el servicio, más tendrán que esperar los clientes que llegan. Como en el caso del proceso de llegada, este tiempo puede ser *determinístico* o *probabilístico*. Con un tiempo de servicio determinístico, cada cliente requiere precisamente la misma cantidad conocida de tiempo para ser atendido. Con un tiempo de servicio probabilístico, cada cliente requiere una cantidad distinta e incierta de tiempo de servicio.

Prioridad
Proceso de servicio en el cual un servidor puede interrumpir el servicio que está proporcionando para dar lugar a un nuevo cliente.

42

Los tiempos de servicio probabilísticos se describen matemáticamente mediante una distribución de probabilidad. En la práctica resulta difícil determinar cuál es la distribución real. Sin embargo, una distribución que ha resultado confiable en muchas aplicaciones, como cuando se trata el caso de bancos y supermercados, es la *distribución exponencial*. En este caso, su función de densidad depende de un parámetro, digamos μ (la letra griega my), y está dada por:

$$s(t) = (1/\mu)e^{-\mu t}$$

en la que:

μ = número promedio de clientes atendidos por unidad de tiempo, de modo que

1/μ = tiempo promedio invertido en atender a un cliente.

En general, el tiempo de servicio puede seguir cualquier distribución, pero, antes de que pueda analizar el sistema, usted necesita identificar dicha distribución.

13.1.5 Clasificaciones de los modelos de colas

Como se mencionó al inicio del presente capítulo, para aplicar las técnicas matemáticas apropiadas, usted debe identificar las características de su sistema de colas, basado en la población de clientes y en los procesos de llegada, de colas y de servicio. El método de clasificación presentado aquí pertenece a un sistema de colas en el que el tamaño de la población de clientes es infinita, los clientes que llegan esperan en una sola fila y el espacio de espera en cada línea es efectivamente infinito.

CARACTERÍSTICAS CLAVE

En este método, los símbolos describen las características del sistema.

✓ El *proceso de llegada*. Este símbolo describe la distribución de tiempo entre llegadas, que es uno de los siguientes:
  a. D para denotar que el tiempo entre llegadas es determinístico.
  b. M para denotar que los tiempos entre llegadas son probabilísticos y siguen una distribución exponencial.
  c. G para denotar que los tiempos entre llegadas son probabilísticos y siguen una distribución general diferente a la exponencial.

✓ El *proceso de servicio*. Este símbolo describe la distribución de tiempos de servicio, que es uno de los siguientes:
  a. D para describir un tiempo de servicio determinístico.
  b. M para denotar que los tiempos de servicio son probabilísticos y siguen una distribución exponencial.
  c. G para denotar que los tiempos de servicio son probabilísticos y siguen una distribución general diferente a la exponencial.

✓ El *proceso de colas*. Este número, c, representa cuántas estaciones o canales paralelos existen en el sistema. (Recuerde que se supone los servidores idénticos en su rapidez de servicio.)

Considere un sistema etiquetado como M/M/3. La primera M indica que el tiempo entre llegadas es probabilístico y sigue una distribución exponencial. La segunda M denota que el tiempo de servicio es probabilístico y sigue, también, una distribución exponencial. El 3 significa que el sistema tiene tres estaciones paralelas, cada una dando un servicio con la misma rapidez.

CARACTERÍSTICAS CLAVE

Cuando el espacio de espera y/o el tamaño de la población de clientes es finito, los dos siguientes símbolos adicionales se incluyen para indicar estas limitaciones:

✓ Un número K que representa el número máximo de clientes que pueden estar en el sistema en cualquier momento (es decir, en servicio o en espera en la fila). Este número es igual al número de estaciones paralelas más el número total de clientes que pueden esperar para ser atendidos.
✓ Un número L que representa el número total de clientes de la población.

Cuando se omite cualquiera de los símbolos, se supone que el valor correspondiente es infinito. Por ejemplo, M/M/3/10 indica que el sistema tiene espacio para un número infinito de clientes, el número K no se ha puesto, y que solamente 10 posibles clientes existen.

En la presente sección, usted ha aprendido que las características básicas de un sistema de colas incluyen el número de clientes disponibles y los procesos de llegada, de colas y de servicio. Estas características se utilizan para clasificar un sistema de modo que se puedan aplicar los análisis matemáticos adecuados para evaluar el desempeño del sistema, sobre la base de las medidas presentadas en la sección 13.2.

13.2 MEDIDAS DE RENDIMIENTO PARA EVALUAR UN SISTEMA DE COLAS

El objetivo último de la teoría de colas consiste en responder cuestiones administrativas pertenecientes al diseño y a la operación de un sistema de colas. El gerente de un banco puede querer decidir si programa tres o cuatro cajeros durante la hora del almuerzo. En una estructura de producción, el administrador puede desear evaluar el impacto de la compra de una nueva máquina que pueda procesar los productos con mayor rapidez.

Cualquier sistema de colas pasa por dos fases básicas. Por ejemplo, considere la cantidad de tiempo que los clientes tienen que esperar en un banco durante el curso de un día, como se muestra en la figura 13.4. Cuando el banco abre en la mañana, no hay nadie en el sistema, de modo que el primer cliente es atendido de manera inmediata. Conforme van llegando más clientes, lentamente se va formando la cola y la cantidad de tiempo que tienen que esperar empieza a aumentar. A medida que avanza el día, el sistema llega a una condición en la que el efecto de la falta inicial de clientes ha sido eliminado y el tiempo de espera de cada cliente ha alcanzado un nivel bastante estable. Como se indica en la figura 13.4, la fase inicial, que conserva los efectos de las condiciones iniciales, se conoce como **fase transitoria**. Después de que los efectos de las condiciones iniciales son eliminados, el sistema entra en una estado estable. A pesar de que las preguntas pertenecientes a ambas fases son importantes, esta sección trata *solamente* sobre el comportamiento del estado estable.

Fase transitoria
El periodo inicial de un sistema de colas en que se conservan los efectos de las condiciones iniciales.

Estado estable
Condición del sistema después de que se han eliminado las condiciones iniciales.

43

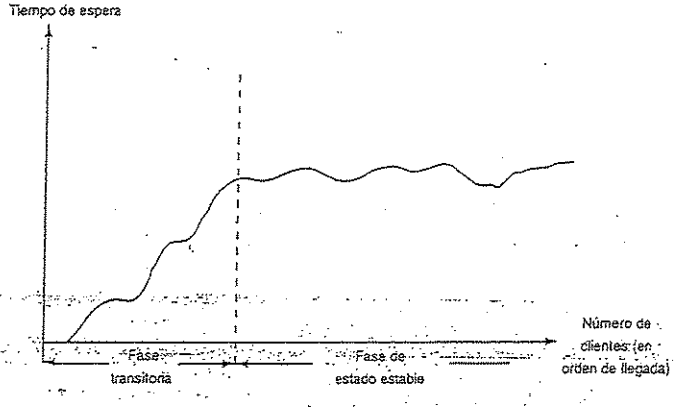


Figura 13.4 Las fases de estado transitorio y estado estable.

Medida de rendimiento Valor numérico que se utiliza para evaluar los méritos de un sistema de colas en estado estable.

Tiempo promedio de espera (W_q) Tiempo promedio que un cliente que llega tiene que esperar en la cola antes de ser atendido.

Tiempo promedio en el sistema (W) Tiempo promedio que un cliente invierte desde su llegada hasta su salida de un sistema de colas.

Longitud media de la cola (L_q) Número promedio de clientes que se encuentran esperando en la fila para ser atendidos.

Número medio en el sistema (L) Número promedio de clientes que se encuentran en el sistema a cualquier tiempo dado.

Probabilidad de bloqueo (p_b) Probabilidad de que un cliente que llega tenga que esperar a ser atendido.

Utilización (U) Fracción de tiempo, en promedio, que un servidor está ocupado.

13.2.1 Algunas medidas de rendimiento comunes

Existen muchas medidas de rendimiento diferentes que se utilizan para evaluar un sistema de colas en estado estable, algunas de las cuales se describen en la presente sección. Para diseñar y poner en operación un sistema de colas, por lo general, los administradores se preocupan por el nivel de servicio que recibe un cliente, así como el uso apropiado de las instalaciones de servicio de la empresa. Algunas de las medidas que se utilizan para evaluar el rendimiento surgen de hacerse las siguientes preguntas:

1. Preguntas relacionadas con el tiempo, centradas en el cliente, como:
   a. ¿Cuál es el tiempo promedio que un cliente recién llegado tiene que esperar en la fila antes de ser atendido? La medida de rendimiento asociada es el tiempo promedio de espera, representado con W_q.
   b. ¿Cuál es el tiempo promedio que un cliente invierte en el sistema entero, incluyendo el tiempo de espera y de servicio? La medida de rendimiento asociada es el tiempo promedio en el sistema, denotado con W.
2. Preguntas cuantitativas pertenecientes al número de clientes, como:
   a. En promedio, ¿cuántos clientes están esperando en la cola para ser atendidos? La medida de rendimiento asociada es la longitud media de la cola, representada con L_q.
   b. ¿Cuál es el número promedio de clientes en el sistema? La medida de rendimiento asociada es el número medio en el sistema, representado con L.
3. Preguntas probabilísticas que implican tanto a los clientes como a los servidores, por ejemplo:
   a. ¿Cuál es la probabilidad de que un cliente que llegue tenga que esperar a ser atendido? La medida de rendimiento asociada es la probabilidad de bloqueo, representada por p_b.
   b. En cualquier tiempo particular, ¿cuál es la probabilidad de que un servidor está ocupado? La medida de rendimiento asociada es la utilización, denotada con U. Esta medida indica también la fracción de tiempo que un servidor está ocupado.

   c. ¿Cuál es la probabilidad de que existan n clientes en el sistema? La medida de rendimiento asociada se obtiene calculando la probabilidad P_0 de que no haya clientes en el sistema, la probabilidad P_1 de que haya un cliente en el sistema, y así sucesivamente. Esto tiene como resultado la distribución de probabilidad de estado, representada por P_n, $n = 0, 1, \ldots$
   d. Si el espacio de espera es finito, ¿cuál es la probabilidad de que la cola esté llena y que un cliente que llegue no sea atendido? La medida de rendimiento asociada es la probabilidad de negación de servicio, representada por p_d.
4. Preguntas relacionadas con los costos, como:
   a. ¿Cuál es el costo promedio por unidad de tiempo para operar el sistema?
   b. ¿Cuántas estaciones de trabajo se necesitan para lograr la mayor efectividad de costos?

Distribución de probabilidad de estado Probabilidad de que se encuentren n clientes en el sistema de colas cuando está en estado estable.

Probabilidad de negación de servicio (p_d) Probabilidad de que un cliente que llega no pueda entrar al sistema debido a que la cola está llena.

El cálculo específico de estas medidas de rendimiento depende de la clase de sistema de colas, como se vio en la sección 13.1. Algunas de estas medidas están relacionadas entre sí. Conocer el valor de una medida le permite encontrar el valor de una medida relacionada. Tales relaciones generales se describen primeramente en la sección 13.2.2, antes de que se presenten los métodos utilizados para calcular estas medidas de rendimiento para un sistema de colas dado.

13.2.2 Relaciones entre medidas de rendimiento

El cálculo de muchas medidas de rendimiento depende de los procesos de llegada y de servicio del sistema de colas específico. Recuerde, de la sección 13.1, que en el caso probabilístico, estos procesos son descritos matemáticamente mediante distribuciones de llegada y de servicio. Incluso sin conocer la distribución específica, las relaciones entre algunas de las medidas de rendimiento pueden obtenerse para ciertos sistemas de colas, únicamente mediante el uso de los siguientes parámetros de los procesos de llegada y de servicio:

λ = número promedio de llegadas por unidad de tiempo

μ = número promedio de clientes atendidos por unidad de tiempo en una estación

Suponga una población de clientes infinita y una cantidad ilimitada de espacio de espera en la fila. El tiempo total que un cliente invierte en el sistema es la cantidad de tiempo invertido en esperar en la fila más el tiempo durante el cual es atendido:

$$\left\{\begin{matrix}\text{Tiempo promedio}\\ \text{en el sistema}\end{matrix}\right\} = \left\{\begin{matrix}\text{tiempo promedio}\\ \text{de espera}\end{matrix}\right\} + \left\{\begin{matrix}\text{tiempo promedio}\\ \text{de servicio}\end{matrix}\right\}$$

El tiempo promedio en el sistema y el tiempo promedio de espera están representados por las cantidades W y W_q, respectivamente. El tiempo promedio de servicio puede expresarse en términos del parámetro μ. Por ejemplo, si μ es cuatro clientes por hora, entonces, en promedio, cada cliente requiere 1/4 de hora para ser atendido. En general, el tiempo promedio de servicio es $1/\mu$, lo cual nos conduce a la siguiente relación:

$$W = W_q + \frac{1}{\mu} \qquad (1)$$

44

Considere ahora la relación entre el número promedio de clientes en el sistema y el tiempo promedio que cada cliente pasa en el sistema. Imagine que un cliente acaba de llegar y se espera que permanezca en el sistema un promedio de 1/2 hora. Durante esta media hora, otros clientes siguen llegando a una tasa λ, digamos doce por hora. Cuando el cliente en cuestión abandona el sistema, después de media hora, deja tras de sí un promedio de (1/2) * 12 = 6 clientes nuevos. Es decir, en promedio, existen seis clientes en el sistema a cualquier tiempo dado. En términos de λ y de las medidas de rendimiento, entonces:

$$\left\{\begin{matrix}\text{Tiempo promedio}\\ \text{de clientes}\\ \text{en el sistema}\end{matrix}\right\} = \left\{\begin{matrix}\text{número promedio}\\ \text{de llegadas por}\\ \text{unidad de tiempo}\end{matrix}\right\} * \left\{\begin{matrix}\text{tiempo promedio}\\ \text{en el sistema}\end{matrix}\right\}$$

de modo que:

$$L = \lambda * W \qquad (2)$$

Utilizando una lógica parecida se obtiene la siguiente relación entre el número promedio de clientes que esperan en la cola y el tiempo promedio de espera en la fila:

$$\left\{\begin{matrix}\text{Número promedio}\\ \text{de clientes}\\ \text{en el sistema}\end{matrix}\right\} = \left\{\begin{matrix}\text{número promedio}\\ \text{de llegadas por}\\ \text{unidad de tiempo}\end{matrix}\right\} * \left\{\begin{matrix}\text{tiempo promedio}\\ \text{en la cola}\end{matrix}\right\}$$

de manera que:

$$L_q = \lambda * W_q \qquad (3)$$

Suponiendo que usted conoce los valores de λ y μ para las medidas W, W_q, L y L_q, se pueden encontrar a partir de las ecuaciones (1) a (3), ya que el valor de cualquiera de ellos esté determinado. Por ejemplo, suponga que λ es 12 y μ es 4 y que usted ha determinado que L_q, el número promedio de clientes que esperan en la cola, es 3:

$$W_q = \frac{L_q}{\lambda} \qquad \text{[De (3)]}$$

$$= \frac{3}{12}$$

$$= \frac{1}{4}$$

$$W = W_q + \frac{1}{\mu} \qquad \text{[De (1)]}$$

$$= \frac{1}{4} + \frac{1}{4}$$

$$= \frac{1}{2}$$

$$L = \lambda * W \qquad \text{[De (2)]}$$

$$= 12 * \frac{1}{2}$$

$$= 6$$

CARACTERÍSTICAS CLAVE

En resumen, conociendo λ y μ, se cumple la siguiente relación:

$$W = W_q + \frac{1}{\mu}$$

$$L = \lambda * W$$

$$L_q = \lambda * W_q$$

En la presente sección, usted ha aprendido las medidas de rendimiento utilizadas para evaluar un sistema de colas y las diferentes relaciones entre ellas. Encontrar los valores para tales medidas depende de la clase específica de modelo de colas que usted tenga. En las secciones 13.3 a 13.6 se muestra cómo encontrar estas medidas cuando se obtienen con un paquete de computación.

13.3 ANÁLISIS DE UN SISTEMA DE COLAS DE UN SOLO CANAL DE UNA SOLA LÍNEA CON LLEGADA EXPONENCIAL Y PROCESOS DE SERVICIO (M/M/1)

En la presente sección usted verá cómo calcular las diferentes medidas de rendimiento descritas en la sección 13.2 y cómo interpretar el resultado de computación asociado al análisis de un sistema de colas M/M/1 que consiste en lo siguiente:

1. Una población de clientes finita.
2. Un proceso de llegada en el que los clientes se presentan de acuerdo con un proceso de Poisson con una tasa promedio de λ clientes por unidad de tiempo.
3. Un proceso de colas que consiste en una sola línea de espera de capacidad infinita, con una disciplina de colas de primero en entrar primero en salir.
4. Un proceso de servicio que consiste en un solo servidor que atiende a los clientes de acuerdo con una distribución exponencial con un promedio de μ clientes por unidad de tiempo.

Para que este sistema alcance una condición de estado estable, la *tasa de servicio promedio*, μ, *debe ser mayor que la tasa de llegadas promedio*, λ. Si éste no fuera el caso, la cola del sistema continuaría creciendo debido a que, en promedio, llegarían más clientes que los que pueden ser atendidos por unidad de tiempo. Considere el problema de Ohio Turnpike Commission.

45

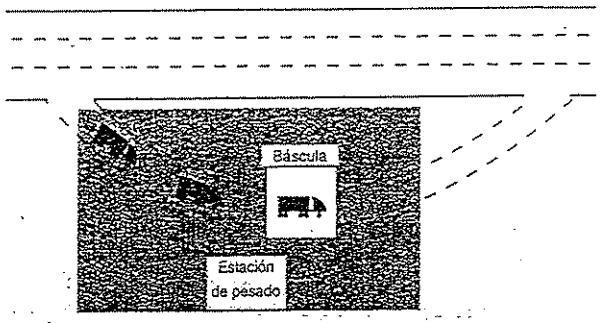


Figura 13.5 Sistema de colas para la estación de pesado en la autopista de Ohio.

EJEMPLO 13.1 EL PROBLEMA DE COLAS DE LA OHIO TURNPIKE COMMISSION La Comisión de la Autopista de Ohio (Ohio Turnpike Commission, OTC) tiene un número de estaciones para el pesado de camiones a lo largo de la autopista de cuota de Ohio, para verificar que el peso de los vehículos cumple con las regulaciones federales. Una de tales estaciones se ilustra en la figura 13.5. La administración de OTC está considerando mejorar la calidad del servicio en sus estaciones de pesado y ha seleccionado una de las instalaciones como modelo a estudiar, antes de instrumentar los cambios. La administración desea analizar y entender el desempeño del sistema actual durante las horas pico, cuando llega a la báscula el mayor número de camiones, suponiendo que el sistema puede desempeñarse bien durante este periodo, el servicio en cualquier otro momento será aún mejor. ■

Formación de cola
EX13_1A.DAT

El gerente de operaciones siente que el sistema actual de la figura 13.5 cumple con las cuatro condiciones presentadas anteriormente. Su siguiente paso es estimar las tasas promedio de llegada y de servicio en dicha estación. De los datos disponibles, suponga que la gerencia determina que los valores son:

λ = número promedio de camiones que llegan por hora = 60

μ = número promedio de camiones que pueden ser pesados por hora = 66

El valor de $\mu = 66$ es mayor que el de $\lambda = 60$, de modo que es posible hacer el análisis de estado estable de este sistema.

13.3.1 Cálculo de las medidas de rendimiento

En términos de los parámetros μ y λ, los investigadores han derivado fórmulas para calcular las diferentes medidas de rendimiento descritas en la sección 13.2 para cualquier sistema de colas M/M/1. Estas fórmulas a menudo se expresan en términos de la intensidad de tráfico, ρ (la letra griega ro), que es el cociente de λ sobre μ. Para el problema de OTC, esta intensidad de tráfico es:

Intensidad de tráfico (ρ) Cociente de la tasa de llegadas, λ, entre la tasa de servicio, μ.

$$\rho = \frac{\lambda}{\mu} = \frac{60}{66} = 0.9091$$

Mientras más cerca esté ρ de 1, más cargado estará el sistema, lo cual tiene como resultado colas más largas y tiempos de espera más grandes.

En términos de ρ, λ y μ, las medidas de rendimiento, para el problema de OTC, se calculan de la manera siguiente:

1. Probabilidad de que no haya clientes en el sistema (P_0):

$$P_0 = 1 - \rho = 1 - 0.9091 = 0.0909$$

Este valor indica que aproximadamente 9% del tiempo un camión que llega no tiene que esperar a que se le proporcione el servicio porque la estación de pesado está vacía. Dicho de otra manera, aproximadamente 91% del tiempo un camión que llega tiene que esperar.

2. Número promedio en la fila (L_q):

$$L_q = \frac{\rho^2}{1-\rho} = \frac{(0.9091)^2}{1-0.9091} = 9.0909$$

En otras palabras, en el estado estable, en promedio, la estación de pesado puede esperar tener aproximadamente nueve camiones esperando para obtener el servicio (sin incluir al que se está pesando).

Cuando ya ha determinado un valor para L_q, usted puede calcular los valores de W_q, W y L, utilizando las relaciones derivadas en la sección 13.2, de la manera siguiente:

3. Tiempo promedio de espera en la cola (W_q):

$$W_q = \frac{L_q}{\lambda} = \frac{9.0909}{60} = 0.1515$$

Este valor indica que, en promedio, un camión tiene que esperar 0.1515 horas, aproximadamente 9 minutos, en la fila antes de que empiece el proceso de pesado.

4. Tiempo promedio de espera en el sistema (W):

$$W = W_q + \frac{1}{\mu} = 0.1515 + \frac{1}{66} = 0.1667$$

46

Este valor indica que, en promedio, un camión invierte 0.1667 horas, 10 minutos, desde que llega hasta que sale.

5. Número promedio en el sistema (L):

$$L = \lambda * W$$
$$= 60 * 0.1667$$
$$= 10$$

Este valor indica que, en promedio, existe un total de 10 camiones en la estación de pesado, ya sea en la báscula o esperando a ser atendidos.

6. Probabilidad de que un cliente que llega tenga que esperar (p_w):

$$p_W = 1 - P_0 = \rho$$
$$= 0.9091$$

Este valor, como se estableció en el paso 1, indica que aproximadamente 91% del tiempo un camión que llega tiene que esperar.

7. Probabilidad de que haya n clientes en el sistema (P_n):

$$P_n = \rho^n * P_0$$

Al utilizar esta fórmula, se obtienen las siguientes probabilidades:

n	P_n
0	0.0909
1	0.0826
2	0.0751
3	0.0683
⋮	⋮

Esta tabla proporciona la distribución de probabilidad para el número de camiones que se encuentran en el sistema. Los números que aparecen en la tabla se pueden utilizar para responder una pregunta como: ¿cuál es la probabilidad de que no haya más de tres camiones en el sistema? En este caso, la respuesta de 0.3169 se obtiene mediante la suma de las primeras cuatro probabilidades de la tabla, para $n = 0, 1, 2$ y 3.

8. Utilización (U):

$$U = \rho$$
$$= 0.9091$$

Este valor indica que aproximadamente 91% del tiempo las instalaciones de pesado están en uso (un camión está siendo pesado). De manera equivalente, aproximadamente 9% del tiempo la estación está sin funcionar, sin que haya camiones que se estén pesando.

Las fórmulas generales para calcular estas diferentes medidas de rendimiento para un sistema de colas M/M/1 con una población de clientes infinita y una capacidad ilimitada de área de espera se resumen en la tabla 13.1, en términos de los parámetros λ, μ y ρ. Ahora que usted ya conoce las fórmulas para las diferentes medidas de rendimiento, puede dejar que la computadora lleve a cabo los cálculos y volver su atención a las cuestiones administrativas, como se describen en la sección 13.3.2.

13.3.2 Interpretación de las medidas de rendimiento

Al evaluar el sistema actual, la gerencia de OTC encuentra que muchas medidas de rendimiento están dentro de los intervalos aceptables. Por ejemplo, un tiempo de espera de $W = 10$ minutos para que un chofer pueda pasar por el proceso de pesado es algo razonable. Se tiene también que un promedio de $L_q = 9$ camiones esperando para ser pesados es tolerable, pues la rampa de salida de la carretera tiene una capacidad de 15 camiones, pero la gerencia está preocupada pues hay ocasiones en que la cola llega hasta la autopista.

TABLA 13.1 Fórmulas para calcular las medidas de rendimiento de un sistema de colas M/M/1

MEDIDA DE RENDIMIENTO	FÓRMULA GENERAL
Número promedio en la fila	$L_q = \frac{\rho^2}{1-\rho}$
Tiempo promedio de espera en la cola	$W_q = L_q / \lambda$
Tiempo promedio de espera en el sistema	$W = W_q + \frac{1}{\mu}$
Número promedio en el sistema	$L = \lambda * W$
Probabilidad de que no haya clientes en el sistema	$P_0 = 1 - \rho$
Probabilidad de que un cliente que llega tenga que esperar	$p_W = 1 - P_0 = \rho$
Probabilidad de que haya n clientes en el sistema	$P_n = \rho^n * P_0$
Utilización	$U = \rho$

Para calcular la probabilidad de que esto suceda, usted debe calcular la probabilidad de que el número de camiones en el sistema sea de 17 o más (uno siendo atendido y 16 o más esperando en la rampa). Este número se obtiene sumando las probabilidades P_n, de que n camiones se encuentren en el sistema, para $n = 17, 18, \ldots$ Esto tiene como resultado un valor de 0.20, es decir, aproximadamente 20% del tiempo los camiones sobrepasarán la rampa completa y llegarán hasta la autopista. Como éste no es un nivel aceptable de desempeño, la gerencia desea mejorar la eficiencia global del sistema, no solamente por la razón anterior, sino también porque se prevé un aumento en el tráfico de camiones sobre la autopista en el futuro cercano. Un informe reciente indica que OTC debería planear una tasa de llegada pico de aproximadamente 70 camiones por hora, en vez del actual valor de 60.

Formación de cola
EX13_1B.DAT

Para atender estas cuestiones, la gerencia de OTC ha propuesto contratar un trabajador adicional, lo cual tendría como resultado un aumento en la eficiencia de aproximadamente 10%. Es decir, con esta persona extra, aproximadamente 73 camiones por hora pueden ser pesados en lugar de los originales 66. Como gerente de operaciones, se le ha pedido a usted que evalúe el impacto de la propuesta.

Este análisis puede llevarse a cabo utilizando las fórmulas de la sección 13.3.1. Solamente cambian la tasa de servicio y de llegada. Los resultados que se obtienen al utilizar la sección de colas del programa STORM para calcular las diferentes medidas de rendimiento para el nuevo sistema, en el cual la tasa de servicio y de llegada de μ se dan en la figura 13.6.

Las primeras tres líneas del informe de la figura 13.6 muestran los datos de entrada. Específicamente, este sistema consiste en un servidor, con una tasa de llegada de 70 camiones por hora, y una tasa de servicio de 73 camiones por hora.

La parte restante de dicho informe enumera los valores de las diferentes medidas de rendimiento. La gerencia está particularmente preocupada tanto por el tiempo promedio que un conductor de camión invierte en el sistema como por el número esperado de camiones que esperan en la rampa. De los resultados que se presentan en la figura 13.6, usted puede informar que, en promedio, un conductor de camión pasa 0.3333 horas (20 minutos) desde el inicio al final del proceso. También que el número promedio de camiones que esperan en la rampa es de aproximadamente 22.

Estas medidas de rendimiento son confirmadas por el resultado obtenido con SQB, presentado en la figura 13.7. La primera línea del informe muestra las tasas de llegada y de servicio. El tiempo promedio que un conductor de camión pasa en el sistema (W) es de 0.332712 horas, que es ligeramente distinto que el presentado en la figura 13.6, debido al error de redondeo. Se tiene también del resultado obtenido con SQB, figura 13.7, que el número promedio de camiones que esperan en la rampa (L_q) es de 22.3308, ligeramente distinto del valor de 22.3744 reportado en la figura 13.6, debido al error de redondeo.

Basándose en estos resultados, la gerencia de OTC encuentra que tal nivel de rendimiento es inaceptable, no sólo porque los conductores se quejarán del hecho de tener que tardar 20 minutos en el sistema, sino también porque la longitud de cola esperada

```
The Problem of the Ohio Turnpike Commission
OTC : M / M / C
Q U E U E   S T A T I S T I C S

Number of identical servers . . . . . . . .        1
Mean arrival rate . . . . . . . . . . . . .   70.0000
Mean service rate per server . . . . . . .    73.0000

Mean server utilization (%) . . . . . . . .   95.8904
Expected number of customers in queue . . . .  22.3744
Expected number of customers in system . . .   23.3333
Probability that a customer must wait . . . .   0.9589
Expected time in the queue . . . . . . . . .    0.3196
Expected time in the system . . . . . . . .     0.3333
```

Figura 13.6 Resultado obtenido con STORM para el problema de colas *M/M/1* de OTC, con $\lambda = 70$ y $\mu = 73$.


```
Final Solution for the Problem of the OTC
M/M/1
With lambda = 70 customers per hour   and f = 73 customers per hour
Overall system effective arrival rate =  69.9994 per hour
Overall system effective service rate =  69.9994 per hour
Overall system effective utilization factor = 0.958904
Average number of customers in the system (L) =  23.2897
Average number of customers in the queue (Lq) =  22.3308
Average time a customer in the system (W) = 0.332712 hour
Average time a customer in the queue (Wq) = 0.319014 hour
The probability that all servers are idle (Po) = 0.041105
The probability an arriving customer waits(Pw) = 0.958895
Probability of n Customers in the System
P(0)   = 0.04110    P(1)   = 0.03942
```

Figura 13.7 Resultado obtenido con QSB para el problema de colas *M/M/1* de OTC con $\lambda = 70$ y $\mu = 73$.

de 22 camiones excede con mucho la capacidad disponible de 15, lo cual podría tener como consecuencia un posible accidente de tráfico en la autopista.

Para obtener niveles de rendimiento aceptables, se ha propuesto otra alternativa, a saber, la construcción de una segunda báscula del otro lado de la estación de pesado. Utilizando el personal actual para que opere ambas básculas, las estimaciones de la gerencia tendrán como resultado una capacidad de peso de aproximadamente 40 camiones por hora en cada báscula.

De nuevo, se le ha pedido que evalúe la presente propuesta. En este caso, sin embargo, usted *no puede* utilizar los resultados obtenidos en la sección 13.3.1. Esto es así debido a que ahora el sistema propuesto tiene *dos* servidores, y el análisis de la sección 13.3.1 se aplica a un sistema con solo *un* servidor. El análisis apropiado se presenta en la sección 13.4.

13.4 ANÁLISIS DE UN SISTEMA DE COLAS DE CANAL MÚLTIPLE DE UNA SOLA LÍNEA CON LLEGADA EXPONENCIAL Y PROCESOS DE SERVICIO (M/M/c)

En la presente sección, usted verá cómo calcular las diferentes medidas de rendimiento descritas en la sección 13.2, y cómo interpretar los resultados asociados obtenidos con computadora para analizar un sistema de colas *M/M/c* consistente en lo siguiente:

1. Una población de clientes infinita.
2. Un proceso de llegada en el que los clientes se presentan de acuerdo a un proceso de Poisson con una tasa promedio de λ clientes por unidad de tiempo.
3. Un proceso de colas que consiste en una sola fila de espera de capacidad infinita, con una disciplina de colas de primero en entrar, primero en salir.
4. Un proceso de servicio que consiste en c servidores idénticos, cada uno de los cuales atiende a los clientes de acuerdo con una distribución exponencial, con una cantidad promedio, μ, de clientes por unidad de tiempo.

48

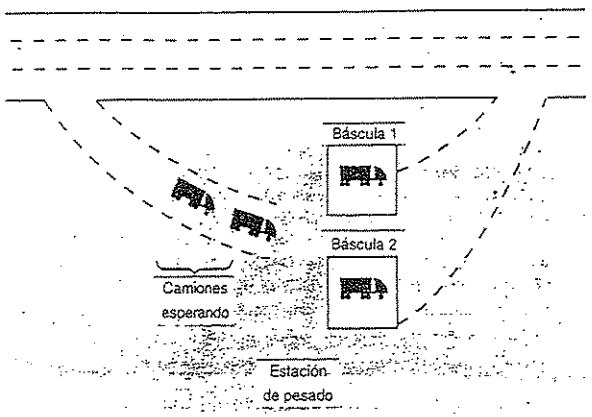


Figura 13.8 Sistema de colas con dos básculas, para el problema de OTC.

Este sistema es distinto al sistema M/M/1 de la sección 13.3 únicamente en el paso 4, que nos permite tener c servidores en lugar de sólo uno. Para que un sistema M/M/c alcance una condición de estado estable, *la tasa total promedio de servicio, $c * \mu$, debe ser estrictamente mayor que la tasa promedio de llegadas, λ.* Si éste no fuera el caso, la cola del sistema continuaría creciendo debido a que, en promedio y por unidad de tiempo, llegarían más clientes que los que pueden ser atendidos.

Formación de colas
OTC_MM2.DAT

Recuerde la última propuesta de OTC de construir una segunda báscula en la estación de pesado, según se describió en la sección 13.3.2 y se ilustró en la figura 13.8. Esta propuesta tiene como resultado un sistema con dos servidores, dos básculas, y la siguiente estimación de llegada, utilizando el personal actual:

$c =$ 2 servidores
$\lambda =$ 70 camiones por hora
$\mu =$ 40 camiones por hora en cada báscula

El valor de $c * \mu = 2 * 40 = 80$, es mayor que el de $\lambda = 70$, de modo que se puede llevar a cabo un análisis de estado estable para este sistema.

13.4.1 Cálculo de las medidas de rendimiento

Los investigadores han derivado fórmulas para calcular las diferentes medidas de rendimiento de un sistema de colas M/M/c, en términos de los parámetros μ y λ. Estas fórmulas, de nueva cuenta, se expresan en términos de ρ, que es el cociente de λ sobre μ. Para el problema de OTC:

$$\rho = \frac{\lambda}{\mu} = \frac{70}{40} = 1.75$$

En términos de ρ, λ y μ, las medidas de rendimiento para el problema de OTC se calculan de la manera siguiente:

1. Probabilidad de que ningún cliente esté en el sistema (P_0):

$$P_0 = \frac{1}{\left(\sum_{n=0}^{c-1} \frac{\rho^n}{n!}\right) + \left(\frac{\rho^c}{c!}\right) * \left(\frac{c}{c-\rho}\right)}$$

donde

$$\sum_{n=0}^{c-1} \frac{\rho^n}{n!} = \frac{\rho^0}{0!} + \frac{\rho^1}{1!} + \ldots + \frac{\rho^{c-1}}{(c-1)!}$$

y $k! = k(k-1)\ldots 1$ (y $0! = 1$). Para el problema de OTC en el cual $\rho = 1.75$ y $c = 2$,

$$\sum_{n=0}^{c-1} \frac{\rho^n}{n!} = \frac{(1.75)^0}{0!} + \frac{(1.75)^1}{1!} = 1 + 1.75 = 2.75$$

$$\frac{\rho^c}{n!} * \frac{c}{c-\rho} = \frac{(1.75)^2}{2!} + \frac{2}{2-1.75} = 1.53125 * 8 = 12.25$$

$$P_0 = \frac{1}{2.75 + 12.25} = \frac{1}{15} = 0.06667$$

Este valor de P_0 indica que aproximadamente 7% del tiempo, la estación de pesado está vacía.

2. Número promedio en la fila (L_q):

$$L_q = \frac{\rho^{c+1}}{(c-1)!} * \frac{1}{(c-\rho)^2} * P_0 = \frac{(1.75)^3}{1!} * \frac{1}{(2-1.75)^2} * 0.06667 = 5.359375 * 16 * 0.06667 = 5.7167$$

Dicho con palabras, en promedio, la estación de pesado puede esperar tener aproximadamente seis camiones esperando a ser atendidos (sin incluir al que ya está en la báscula).

49

Ahora que ya se ha determinado un valor para L_q, los valores de W_q, W y L pueden calcularse utilizando la relación derivada en la sección 13.2:

3. Tiempo promedio de espera en la cola (W_q):

$$W_q = \frac{L_q}{\lambda}$$
$$= \frac{5.7167}{70}$$
$$= 0.081667$$

Este valor indica que en promedio, un camión tiene que esperar 0.0817 horas, aproximadamente 5 minutos, en la fila antes de iniciar el proceso de pesado.

4. Tiempo promedio de espera en el sistema(W):

$$W = W_q + \frac{1}{\mu}$$
$$= 0.081667 + \frac{1}{40}$$
$$= 0.081667 + 0.025$$
$$= 0.10667$$

Este valor indica que en promedio, un camión tiene que esperar 0.10667 horas, aproximadamente 7 minutos, desde que llega hasta que sale de la estación.

5. Número promedio en el sistema (L):

$$L = \lambda * W$$
$$= 70 * 0.10667$$
$$= 7.4667$$

Este valor indica que, en promedio, se tienen entre siete y ocho camiones esperando en la estación, ya sea en la báscula o en espera de ser atendidos.

6. Probabilidad de que un cliente que llega tenga que esperar (p_w):

$$p_W = \frac{1}{c!} * \rho^c * \frac{c}{c-\rho} * P_0$$
$$= \frac{1}{2!} * (1.75)^2 * \frac{2}{2-1.75} * 0.06667$$
$$= 0.5 * 3.0625 * 8 * 0.06667$$
$$= 0.81667$$

Este valor indica que aproximadamente 82% de las veces un camión que llega tiene que esperar o, de manera equivalente, aproximadamente 18% de las veces un camión que llega es pesado sin que tenga que esperar.

7. Probabilidad de que haya n clientes en el sistema (P_n):
Si $n \le c$:

$$P_n = \frac{\rho^n}{n!} * P_0$$

Al utilizar esta fórmula se obtienen las siguientes probabilidades:

n	P_n
0	0.06667
1	0.11667
2	0.10210

Si $n > c$

$$P_n = \frac{\rho^n}{(c!)c^{n-c}} * P_0$$

Al utilizar esta fórmula, se obtienen las siguientes probabilidades:

n	P_n
3	0.08932
4	0.07816
⋮	⋮

Estas tablas proporcionan la distribución de probabilidad para el número de camiones que hay en el sistema. Las cantidades que aparecen en tales tablas se pueden utilizar para responder preguntas como: ¿cuál es la probabilidad de que al menos una báscula no esté funcionando? Esta probabilidad es la misma que la probabilidad de que haya menos de dos camiones en el sistema. Sumando las dos primeras probabilidades de la tabla para $n = 0$ y 1, se obtiene la respuesta: 0.18334.

8. Utilización (U):

$$U = 1 - \left[P_0 + \left(\frac{c-1}{c}\right)P_1 + \left(\frac{c-2}{c}\right)P_2 + \ldots + \left(\frac{1}{c}\right)P_{c-1}\right]$$
$$= 1 - \left[P_0 + \left(\frac{1}{2}\right)P_1\right]$$
$$= 1 - [0.06667 + (0.5 * 0.11667)]$$
$$= 1 - 0.125$$
$$= 0.875$$

Este valor indica que cada báscula está ocupada 87% del tiempo.

En la tabla 13.2 se resumen las fórmulas para un sistema de colas M/M/c con una población infinita de clientes y un área de espera de capacidad ilimitada, en términos de los parámetros λ, μ y ρ. Observe que cuando $c = 1$, estas fórmulas tienen como resultado los mismos valores de las medidas de rendimiento de un sistema M/M/1, derivadas en la sección 13.3. Usted puede ahora dejar que la computadora efectúe estos cálculos y dirigir su atención a cuestiones gerenciales.

TABLA 13.2 Fórmulas para calcular las medidas de rendimiento de un sistema de colas M/M/c

MEDIDA DE RENDIMIENTO	FÓRMULA GENERAL
Probabilidad de que no haya clientes en el sistema	$P_0 = \dfrac{1}{\left(\sum_{n=0}^{c-1} \dfrac{\rho^n}{n!}\right) + \left(\dfrac{\rho^c}{c!}\right) * \left(\dfrac{c}{c-\rho}\right)}$
Número promedio en la fila	$L_q = \dfrac{\rho^{c+1}}{(c-1)!} * \dfrac{1}{(c-\rho)^2} * P_0$
Tiempo promedio de espera en la cola	$W_q = \dfrac{L_q}{\lambda}$
Tiempo promedio de espera en el sistema	$W = W_q + \dfrac{1}{\mu}$
Número promedio en el sistema	$L = \lambda * W$
Probabilidad de que un cliente que llega tenga que esperar	$P_W = \dfrac{1}{c!} * \rho^c * \dfrac{c}{c-\rho} * P_0$
Probabilidad de que haya n clientes en el sistema ($n \leq c$)	$P_n = \dfrac{\rho^n}{n!} * P_0$
Probabilidad de que haya n clientes en el sistema ($n > c$)	$P_n = \dfrac{\rho^n}{(c!)c^{n-c}} * P_0$
Utilización	$U = 1 - \left[P_0 + \left(\dfrac{c-1}{c}\right) P_1 + \left(\dfrac{c-2}{c}\right) P_2 + \ldots + \left(\dfrac{1}{c}\right) P_{c-1}\right]$

13.4.2 Interpretación de las medidas de rendimiento

Formación de cola
OTC_MM2.DAT

Los resultados de la evaluación de las fórmulas de la tabla 13.2 con el paquete de cómputo STORM, para el sistema de colas propuesto para OTC, se muestran en la figura 13.9. Las primeras tres líneas del informe de la figura corresponden a los datos de entrada. Este sistema tiene una tasa de llegada de 70 camiones por hora y dos servidores, con una tasa promedio de servicio de 40 camiones por hora en cada servidor.

El informe de la figura 13.9 también enumera los valores de las medidas de rendimiento. Usted puede informar a la gerencia sobre el tiempo promedio que un

conductor de camión tiene que invertir en el sistema y el número esperado de camiones que esperan en la rampa. En la última línea del informe de la figura 13.9, usted puede observar que, en promedio, un conductor espera 0.1067 horas (aproximadamente 7 minutos) desde que entra hasta que sale. También, que el número promedio de camiones que esperan en la rampa es de aproximadamente 5.7167.

La gerencia de OTC encuentra aceptable este nivel de rendimiento. Sin embargo, la gerencia de nuevo se pregunta si la fila de camiones llegará hasta la autopista. Este suceso se presenta cuando hay dos camiones en la báscula y más de 15 esperando en la rampa. ¿Cuál es la probabilidad de que más de 17 camiones estén en el sistema en cualquier momento?

Se puede utilizar el informe de STORM, figura 13.10, para responder a esta pregunta. Específicamente, la probabilidad de que se presente este caso se obtiene sumando las probabilidades de la figura correspondientes a cada valor de $n = 18, 19, \ldots$ La probabilidad resulta ser de 9.6%. Si tal valor no es aceptable, deben sugerirse modelos alternativos. Por ejemplo, la contratación de una persona más para aumentar la tasa de servicio, o el aumento de la capacidad del área de espera teniendo dos filas en lugar de una sola, podrían ser sugerencias apropiadas.

Las medidas de rendimiento para este problema son confirmadas por el resultado obtenido con el paquete de computación QSB, mostrado en la figura 13.11. La primera línea de tal informe muestra la tasa de llegada de 70 camiones por hora y la tasa de servicio de 40 camiones por hora en cada báscula. La cantidad promedio de tiempo que un conductor tiene que invertir en el sistema (W) es de 0.106667, la misma reportada en la figura 13.9. El número promedio de camiones que esperan en la rampa para ser pesados (L_q) es de 5.716664, que también es el mismo que se muestra en la figura 13.9.

Usted ha visto cómo calcular e interpretar las medidas de rendimiento para un sistema de colas M/M/c, tanto a mano como con una computadora. Cuando solamente hay uno o dos sistemas alternativos para analizar, a menudo, se puede hacer una elección aceptable basándose en las medidas de rendimiento. Sin embargo, cuando se tienen disponibles muchas alternativas, a veces debe incurrirse en costos de información adicionales para seleccionar la mejor alternativa, según se describe en la sección 13.5.

```
The Problem of the Ohio Turnpike Commission
OTC : M / M / C
Q U E U E   S T A T I S T I C S

Number of identical servers . . . . . . . .        2
Mean arrival rate . . . . . . . . . . . . .   70.0000
Mean service rate per server . . . . . . .    40.0000

Mean server utilization (%) . . . . . . . .   87.5000
Expected number of customers in queue . . .    5.7167
Expected number of customers in system . . .   7.4667
Probability that a customer must wait . . . .  0.8167
Expected time in the queue . . . . . . . . .   0.0817
Expected time in the system . . . . . . . . .  0.1067
```

Figura 13.9 Medidas de rendimiento obtenidas con STORM para el problema de dos servidores de OTC.

```
The Problem of the Ohio Turnpike Commission
OTC : M / M / C
PROBABILITY DISTRIBUTION OF NUMBER IN SYSTEM
Number  Prob  0   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9   1
                +----+----+----+----+----+----+----+----+----+----+
   0   0.0667|***+                                                |
   1   0.1167|******---                                            |
   2   0.1021|*****+--------                                       |
   3   0.0893|****+-------------                                   |
   4   0.0782|****-----------------                                |
   5   0.0684|***+-------------------                              |
   6   0.0598|***----------------------                             |
   7   0.0524|**+-------------------------                          |
   8   0.0458|**+--------------------------                         |
   9   0.0401|**+----------------------------                       |
  10   0.0351|*+-------------------------------                    |
  11   0.0307|*+--------------------------------                   |
  12   0.0269|*+----------------------------------                 |
  13   0.0235|*+----------------------------------                 |
  14   0.0206|*+------------------------------------               |
  15   0.0180|*+------------------------------------               |
  16   0.0157|*+------------------------------------               |
  17   0.0138|*+------------------------------------               |
  18   0.0121|*+------------------------------------               |
  19   0.0105|*+------------------------------------               |
  20   0.0092|+-------------------------------------               |
  21   0.0081|+-------------------------------------               |
  22   0.0071|+-------------------------------------               |
  23   0.0062|+-------------------------------------               |
  24   0.0054|+-------------------------------------               |
OVER   0.0376|**+-----------------------------------               |
                +----+----+----+----+----+----+----+----+----+----+
```

Figura 13.10 Probabilidad obtenida con STORM de que haya n camiones en el sistema de

```
Final Solution for the Problem of the OTC
M/M/2
With lambda = 70 customers per hour   and f = 40 customers per hour
        Overall system effective arrival rate =  70.0000 per hour
        Overall system effective service rate =  70.0000 per hour
  Overall system effective utilization factor = 0.875001
 Average number of customers in the system (L) = 7.466666
 Average number of customers in the queue (Lq) = 5.716664
    Average time a customer in the system (W) = 0.106667 hour
    Average time a customer in the queue (Wq) = 0.081667 hour
 The probability that all servers are idle (Po)= 0.066667
 The probability an arriving customer waits(Pw)= 0.816667
        Probability of n Customers in the System
 P(0)   = 0.06667    P(1)   = 0.11667
```

Figura 13.11 Medidas de rendimiento obtenidas con QSB para el problema de dos servidores de OTC.

13.5 ANÁLISIS ECONÓMICO DE LOS SISTEMAS DE COLAS

En la sección 13.4, usted vio la ventaja de tener más de un servidor, a saber, la reducción del tiempo de espera y del número de clientes que esperan a ser atendidos. Claramente, mientras más servidores se tengan, mejor será el servicio a los clientes. Sin embargo, cada servidor implica costos de operación. ¿De qué manera evalúa usted este equilibrio entre nivel de servicio y costo?

En el ejemplo de la Ohio Turnpike Commission de la sección 13.4, la decisión de poner en operación dos básculas, es decir, tener dos servidores, está basada exclusivamente en el logro de un nivel aceptable de servicio, lo que en este caso significa asegurar tiempos de espera y colas de tamaño razonables. En algunos problemas, es posible utilizar información sobre costos para llevar a cabo un análisis económico del equilibrio entre el número de servidores y el nivel de servicio al cliente. Considere el problema de American Weavers, Inc.

EJEMPLO 13.2 PROBLEMA DE COLAS DE AMERICAN WEAVERS, INC. American Weavers, Inc., tiene una planta de manufactura de tela en Georgia. La planta tiene un gran número de máquinas tejedoras que con frecuencia se atascan. Estas máquinas son reparadas basándose en el procedimiento de la primera en entrar, la primera en ser revisada, por uno de los siete miembros del personal de reparación. Durante varios recorridos, la gerente de producción ha observado que, en promedio, aproximadamente de 10 a 12 máquinas están fuera de operación en cualquier momento debido a que están atascadas. Ella sabe que contratar personal de reparaciones adicional bajaría el número de máquinas sin funcionar, lo cual traería como consecuencia un aumento en la producción, pero no sabe a cuántas personas más debería contratar. Como asesor administrativo, se le ha mandado llamar a usted para que ayude a determinar dicho número. ■

Formación de cola
EX13_2A.DAT

13.5.1 Modelo y análisis del sistema de colas actual

El primero paso que debe dar consiste en analizar las condiciones de operación actuales. Debe reconocer que las máquinas tejedoras conforman un modelo de colas. Los clientes están constituidos por las máquinas que se atascan de vez en cuando. Existe un gran número de tales máquinas, de modo que podría suponer, razonablemente, que la población de clientes es infinita. Se tienen siete servidores independientes e idénticos que reparan las máquinas basándose en una estrategia de primera en entrar, primera en darle servicio. Usted puede pensar en estas máquinas formando una sola fila en espera de pasar con el siguiente servidor que esté disponible.

Para modelar esta operación, el siguiente paso consiste en reunir y analizar los datos correspondientes a los procesos de llegada y de servicio. Suponga que se tiene que:

1. La aparición de máquinas atascadas puede ser aproximada por un proceso de llegada de Poisson con una tasa promedio de 25 por hora.
2. Cada máquina atascada requiere una cantidad aleatoria de tiempo para su reparación, que puede ser aproximada por una distribución exponencial con un tiempo promedio de servicio de 15 minutos, lo cual, para cada servidor, significa una tasa promedio de cuatro máquinas por hora.

Con estas observaciones, el sistema actual puede modelarse como un sistema de colas M/M/7, con $\lambda = 25$, $\mu = 4$ y una población y un área de espera infinitas.

```
M/M/7 : M / M / C
QUEUE STATISTICS

Number of identical servers . . . . . .          7
Mean arrival rate . . . . . .               25.0000
Mean service rate per server . . . . . .     4.0000

Mean server utilization (%) . . . . . .     89.2857
Expected number of customers in queue . . . .  5.8473
Expected number of customers in system . . .  12.0973
Probability that a customer must wait . . . .  0.7017
Expected time in the queue . . . . . .       0.2339
Expected time in the system . . . . . .      0.4839
```

Figura 13.12. Medidas de rendimiento obtenidas con STORM para el problema de American Weavers, Inc., con siete reparadores.

Los resultados obtenidos con el paquete STORM con respecto a las medidas de rendimiento se presentan en la figura 13.12. Como puede ver, el gerente de producción había estimado con bastante precisión el hecho de que entre 10 y 12 máquinas están atascadas, en promedio, en cualquier momento. De hecho, ese número en el informe es de 12.09. La última línea del reporte indica que las máquinas atascadas están fuera de operación durante un tiempo promedio de 0.4839 horas, aproximadamente 29 minutos.

Como asesor, se le ha pedido a usted que recomiende el número de reparadores adicionales que se necesitarían contratar. Usted conoce las medidas de rendimiento de un total de siete trabajadores. ¿De qué manera cambian las medidas de rendimiento si se aumenta el personal de reparación? Las medidas de rendimiento asociadas para un número entre 7 y 11 reparadores se muestran en la tabla 13.3.

A medida que aumenta el tamaño del personal de 7 a 11, el número promedio de máquinas fuera de operación disminuye de aproximadamente 12 a 6.333. Similarmente, la cantidad promedio de tiempo que una máquina está fuera de operación disminuye de 0.4839 horas (aproximadamente 29 minutos) a 0.2533 horas (aproximadamente 15 minutos). Ahora necesita información sobre los costos para determinar cuántos reparadores adicionales, si se requieren, deben contratarse.

TABLA 13.3 *Medidas de rendimiento para el problema de American Weavers, Inc., con diferentes tamaños de personal de reparación*

	NÚMERO DE REPARADORES				
	7	8	9	10	11
Utilización (%)	89.2857	78.1250	69.4444	62.5000	56.8182
Número esperado en la cola	5.8473	1.4936	0.5363	0.2094	0.0830
Número esperado en el sistema	12.0973	7.7436	6.7863	6.4594	6.3330
Probabilidad de que un cliente tenga que esperar	0.7017	0.4182	0.2360	0.1257	0.0630
Tiempo esperado en la cola	0.2339	0.0597	0.0215	0.0084	0.0033
Tiempo esperado en el sistema	0.4839	0.3097	0.2715	0.2584	0.2533

BIBLIOTECA

13.5.2 Análisis de costos del sistema de colas

Al analizar los méritos de contratar personal de reparación adicional en American Weavers, Inc., usted debería identificar dos componentes importantes:

1. Un costo por hora basado en el tamaño del personal,

$$\left\{\begin{matrix}\text{Costo total de}\\ \text{personal por hora}\end{matrix}\right\} = \left\{\begin{matrix}\text{costo por hora para}\\ \text{cada reparador}\end{matrix}\right\} * \left\{\begin{matrix}\text{número de}\\ \text{reparadores}\end{matrix}\right\}$$

2. Un costo por hora basado en el número de máquinas fuera de operación:

$$\left\{\begin{matrix}\text{Costo total}\\ \text{por la}\\ \text{espera}\end{matrix}\right\} = \left\{\begin{matrix}\text{costo por hora para}\\ \text{cada máquina fuera}\\ \text{de operación}\end{matrix}\right\} * \left\{\begin{matrix}\text{número promedio}\\ \text{de máquinas fuera}\\ \text{de operación}\end{matrix}\right\}$$

Para seguir adelante, necesita ahora conocer el costo por hora de cada miembro del personal de reparación (denotado con c_r) y el costo por hora de una máquina fuera de operación (denotado por c_m), que es el costo de una hora de producción perdida. Suponga que el departamento de contabilidad le informa que cada reparador le cuesta a la compañía $50 por hora, incluyendo impuestos, prestaciones, etc. El costo de una hora de producción perdida deberá incluir costos explícitos, como la cantidad de ganancias no obtenidas, y costos implícitos, como la pérdida de voluntad por parte del cliente si no se cumple con la fecha límite de entrega. Estos costos implícitos son difíciles de estimar. Sin embargo, suponga que el departamento de contabilidad estima que la compañía pierde $100 por cada hora que una máquina esté fuera de operación. Ahora ya puede calcular un costo total para cada uno de los tamaños de personal. Para un personal de siete reparadores, el número esperado de máquinas en el sistema es 12.0973 (véase la tabla 13.3), de modo que:

$$\begin{aligned}\text{Costo total} &= (\text{costo del personal}) + (\text{costo de la espera})\\ &= \left\{\left(\begin{matrix}\text{costo por hora}\\ \text{por persona}\end{matrix}\right) * \left(\begin{matrix}\text{número de}\\ \text{reparadores}\end{matrix}\right)\right\} + \left\{\left(\begin{matrix}\text{costo por hora por}\\ \text{cada máquina fuera}\\ \text{de operación}\end{matrix}\right) * \left(\begin{matrix}\text{número esperado}\\ \text{de máquinas fuera}\\ \text{de operación}\end{matrix}\right)\right\}\\ &= (50 * 7) + (100 * 12.0973)\\ &= \$1559.73 \text{ por hora}\end{aligned}$$

Realizando cálculos parecidos para cada uno de los tamaños de personal restantes se tiene como resultado los costos por hora de cada alternativa que presentamos en la tabla 13.4.

De los resultados, usted puede ver que la alternativa que tiene el menor costo por hora, $1128.63, es tener un total de nueve reparadores. En consecuencia, su recomendación a la gerencia de producción de American Weavers, Inc., es contratar a dos reparadores adicionales. Estos dos nuevos empleados tendrán un costo de $100 por hora, pero este costo adicional está más que justificado por los ahorros que se tendrán con menos máquinas fuera de operación. La recomendación reducirá el costo por hora de $1559.73 a $1128.63, un ahorro de aproximadamente $430 por hora, mayor que la cantidad que cubre sus honorarios.

53

TABLA 13.4 Costo por hora para diferentes tamaños de personal de reparación de American Weavers, Inc.

TAMAÑO DE PERSONAL	NÚMERO ESPERADO EN EL SISTEMA	COSTO POR HORA ($)				
7	12.0973	(50 * 7)	+	(100 * 12.0973)	=	1559.73
8	7.7436	(50 * 8)	+	(100 * 7.7436)	=	1174.36
9	6.7863	(50 * 9)	+	(100 * 6.7863)	=	1128.63
10	6.4594	(50 * 10)	+	(100 * 6.4594)	=	1145.94
11	6.3330	(50 * 11)	+	(100 * 6.3330)	=	1183.30

CARACTERÍSTICAS CLAVE

En resumen, para evaluar un sistema de colas en el que usted controla el número de servidores o su tasa de servicio, se necesitan las siguientes estimaciones de costo y medidas de rendimiento:

✓ El costo por servidor por unidad de tiempo (c_s).
✓ El costo por unidad de tiempo por cliente esperando en el sistema (c_w).
✓ El número promedio de clientes en el sistema (L).

Por cada alternativa que implique c servidores, calcule el siguiente costo total por unidad de tiempo:

Costo total por unidad de tiempo con c servidores

$$= (\text{costo de los servidores}) + (\text{costo de la espera})$$

$$= \left\{\begin{pmatrix}\text{costo por servidor}\\ \text{por unidad de tiempo}\end{pmatrix} * \begin{pmatrix}\text{número de}\\ \text{servidores}\end{pmatrix}\right\} + \left\{\begin{pmatrix}\text{costo por cliente}\\ \text{por unidad}\\ \text{de tiempo}\end{pmatrix} * \begin{pmatrix}\text{número esperado}\\ \text{de clientes en}\\ \text{el sistema}\end{pmatrix}\right\}$$

$$= (c_s * c) + (c_w * L)$$

Por último, seleccione la alternativa que ofrezca el costo total mínimo por unidad de tiempo.

13.6 ANÁLISIS DE OTROS MODELOS DE COLAS USANDO LA COMPUTADORA

En la sección 13.1 usted aprendió que existen diferentes modelos de colas basados en las características del sistema. Usted sabe que la población de posibles clientes puede

ser finita o infinita, la área de espera puede ser limitada o ilimitada en su capacidad, y el proceso de servicio puede seguir o no una distribución exponencial.

Los modelos M/M/c y los ejemplos presentados en las secciones 13.2 a 13.5, todos suponen una población infinita de clientes, un área de espera ilimitada, una distribución de Poisson en las llegadas y una distribución exponencial en el servicio. Sobre la base de estas suposiciones, usted efectúa los cálculos de las medidas de rendimiento utilizando las fórmulas de la sección 13.4. ¿Qué sucede cuando una o varias de tales suposiciones no cumplen con el sistema de colas que se está investigando? En algunos casos, sigue siendo posible calcular las medidas de rendimiento. Sin embargo, las fórmulas se vuelven bastante complejas y se necesita un paquete de cómputo para llevar a cabo los cálculos de una variedad de modelos de colas que se encuentran comúnmente y en los cuales los clientes que llegan esperan en una sola línea.

13.6.1 Un sistema M/M/c con una población de clientes finita (M/M/c/K)

En los modelos de colas que usted ha visto hasta este punto, se ha supuesto que existe una población infinita de clientes. A pesar de que en la realidad esto nunca es verdadero, para muchas situaciones prácticas la suposición es razonable. Por ejemplo, cuando la población real es muy grande, como en el caso de clientes que llegan a un supermercado o a un banco, tal suposición es bastante justificable. En algunos modelos, sin embargo, la suposición de una población infinita no es apropiada. Por ejemplo:

1. Personal de mantenimiento proporciona servicio de reparación a un laboratorio de computación conformado por 50 microcomputadoras. En este caso, las 50 computadoras son clientes y los miembros del personal de reparaciones son los servidores.
2. Una compañía da mantenimiento a los elevadores de 30 edificios de oficinas. Aquí, los 30 edificios son clientes y el personal de reparaciones de la compañía son los servidores.
3. Una flotilla de automóviles de una compañía se encuentran disponibles para 20 directivos. En este caso, los 20 directivos son los clientes y los automóviles de la flotilla son los servidores.

En cada uno de los ejemplos anteriores, la población de clientes es bastante limitada en tamaño. Obtener medidas de rendimiento utilizando la suposición de una población de clientes infinita puede producir resultados no válidos. Recuerde el problema enfrentado por el gerente de producción de American Weavers, Inc., en el que las máquinas tejedoras se atascan de tiempo en tiempo y requieren servicio. Al realizar el análisis, a usted, como asesor, se le hizo creer que había un número suficiente de máquinas tejedoras, clientes, de modo que la suposición de una población infinita era válida. Los resultados que obtuvo en la sección 13.5, sobre la base de esta suposición, llevaron a la recomendación de contratar dos reparadores adicionales que se sumen a los siete reparadores actuales.

Al analizar con más detalle la situación con el gerente de producción, sin embargo, usted se ha enterado de que solamente tienen 100 máquinas tejedoras. Antes de escribir su informe final, usted necesita ver si la consideración de la población de clientes como finita tiene un impacto significativo en su recomendación.

En general, la suposición de una población finita afecta el proceso de llegada. Con una población infinita, la tasa de llegadas permanece igual, sin importar cuántos clientes hayan llegado. Éste *no* es el caso con una población finita. Suponga que usted estima que los clientes de una población infinita llegan a una tienda de abarrotes con

una tasa de, digamos, 20 por hora. Incluso si ya hay 60 clientes en espera, resulta razonable suponer que los clientes nuevos continuarán llegando a un ritmo de 20 por hora, porque hay un número infinito de clientes que aún no están en la tienda. Sin embargo, suponga que la tienda tiene una base de 100 clientes y 60 ya están dentro. Ya no resulta razonable suponer que los 40 clientes restantes llegarán con una tasa de 20 por hora, pues existen muy pocos de ellos que aún no llegan.

CARACTERÍSTICAS CLAVE

En general, con un número finito de clientes, la tasa de llegadas disminuye conforme aumenta el número de clientes en el sistema, porque existen menos clientes restantes que aún no llegan.

Los procesos de llegada para una población finita no se pueden describir de manera matemática mediante una tasa de llegada *fija*, debido a que la tasa cambia según el número de clientes que se encuentren en el sistema. Cuantos más clientes haya en el sistema, menor será la tasa de llegada de clientes. Considere los extremos. Si el sistema no tiene clientes, la tasa de llegada estará en nivel más alto. Si todos los clientes están en el sistema en un momento dado, la tasa de llegada bajará a cero. ¿De qué modo, entonces, se puede especificar la tasa de llegadas?

El proceso de llegada se describe considerando la *tasa de llegada de cada cliente individual*. Esto es, usted debe identificar con qué frecuencia llega un cliente en particular. En el problema de American Weavers, con 100 máquinas, usted debe determinar la tasa a la cual cada máquina requiere reparación. Suponga que la frecuencia es de una vez cada cuatro horas. Esta frecuencia se convierte en una tasa por hora:

$$\lambda = 1/4 = 0.25 \text{ atascadas por hora por máquina}$$

```
M/M/7/K : M / M / C / K / K
QUEUE  STATISTICS

Number of identical servers . . . . . . . . .        7
Mean arrival rate per customer . . . . . . .    0.2500
Mean service rate per customer . . . . . . .    4.0000
Size of the source population . . . . . . . .      100

Mean server utilization (%) . . . . . . . . .  82.5102
Expected number of customers in queue . . . .   1.8128
Expected number of customers in system . . .    7.5885
Probability that a customer must wait . . . .   0.5254
Expected time in the queue . . . . . . . . .    0.0785
Expected time in the system . . . . . . . . .   0.3285
```

Figura 13.13 Medidas de rendimiento obtenidas con STORM para el problema de American Weavers, Inc., con una población finita.

Formación de cola
EX13_2B.DAT

Recuerde que, actualmente, existe un personal de siete reparadores, cada uno capaz de reparar una máquina en un tiempo promedio de 15 minutos. La tasa de servicio por servidor es $\mu = 4$ máquinas por hora. Recuerde también que el costo por hora de cada reparador es de $50, y que el costo por hora de producción perdida cuando una máquina se atora es de $100. Al introducir estos datos en el paquete de software STORM, junto con el hecho de que el tamaño de la población es de 100, produce los valores para las medidas de rendimiento mostrados en la figura 13.13.

Compare las medidas de rendimiento de la figura 13.13 con las de la figura 13.12 presentada en la sección 13.5, correspondiente a una población infinita. Usted puede ver que hay algunas diferencias. Por ejemplo, con una población infinita, el número esperado de máquinas fuera de operación es de 12.0973; con una población finita de 100, la misma estadística es de 7.5885. ¿Qué es lo que causa esta significativa diferencia? Con una población infinita, la tasa de llegada se fija en 25 máquinas por hora, independientemente de cuántas máquinas estén en reparación en cualquier momento. Pero considere una población finita de 100 máquinas. Si las 100 máquinas tejedoras están trabajando, la tasa de descomposturas también es de 25 por hora (0.25 atascadas por máquina * 100 máquinas). Pero, ¿qué sucede cuando, digamos, diez máquinas se atascan? Solamente hay 90 máquinas operando, de modo que la tasa baja a 22.5 (0.25 atascadas por máquina * 90 máquinas) tejedoras atascadas por hora. El número menor de llegadas tiene como resultado un menor número de máquinas atascadas que requieren servicio.

El análisis económico de STORM para este ejemplo, con una población finita, se muestra en la figura 13.14. Los costos por cada servidor y por espera son los mismos aquí que antes. La última línea de dicho informe proporciona el costo total de $1108.85 por hora para el sistema actual con siete reparadores. STORM lleva a cabo un análisis económico parecido con diferentes cantidades de reparadores e informa el tamaño del personal con menor costo por hora en la última columna, etiquetada con "Optimal system" (Sistema óptimo), del informe de la figura 13.14. Como puede ver, el tamaño óptimo de reparadores es de solamente ocho, con una población finita de 100 máquinas. Este tamaño de personal es menor que el número óptimo de nueve que se obtiene cuando se supone una población infinita, debido a que menos máquinas tejedoras están atascadas. Por consiguiente, usted deberá modificar su recomendación y sugerir que solamente se contrate un reparador adicional.

```
M/M/7/K : M / M / C / K / K
COST ANALYSIS PER UNIT TIME

                        Current System        Optimal System *
Number of servers  |        7          |        8          |
Cost per server    |   50.0000         |   50.0000         |
Cost of service    |            350.0000|            400.0000|
Mean number in system |  7.5885         |   6.5145          |
Waiting cost/customer |100.0000         |  100.0000         |
Cost of waiting    |            758.8500|            651.4500|
                   |          ----------|          ----------|
TOTAL COST                     1108.8500            1051.4500
                   * Optimization is over number of servers
```

Figura 13.14 Análisis económico hecho con STORM para el problema de American Weavers, Inc., con una población finita.

13.6.2 Un sistema M/M/c con capacidad de espera limitada (M/M/c/K)

¿Es válida la suposición de un área de espera ilimitada para los clientes? Los modelos de colas vistos hasta aquí, han utilizado esta suposición. De hecho, en muchas situaciones prácticas, esta suposición es razonable. En un banco, el área de espera es limitada, pero los clientes que esperan nunca necesitan más de tal espacio. Incluso cuando la fila se hace muy grande, el espacio de espera puede extenderse hasta los pasillos o la calle. Así pues, para todos los propósitos prácticos, el área de espera puede suponerse ilimitada. En algunos modelos, sin embargo, esta suposición no resulta apropiada.

1. Un sistema de reservaciones por teléfono puede mantener un número limitado de llamadas. Aquí, las llamadas que llegan son los clientes y los recepcionistas son los servidores.
2. En una planta de producción, las partes que llegan de una etapa previa de producción a una máquina en donde se les hará cierto proceso esperan en una banda transportadora con una capacidad limitada. Si las partes que esperan alcanzan la capacidad de la banda, la producción en la etapa anterior deberá detenerse. En este caso, las partes que llegan de la etapa anterior son los clientes y la máquina el servidor.
3. Un estacionamiento, una vez lleno a toda su capacidad, debe rechazar a los automóviles que llegan. En este caso, los autos que llegan son los clientes, cada cajón de estacionamiento es un servidor y no hay espacio de espera.

En cada uno de estos ejemplos, no hay área de espera o la capacidad de ésta es limitada. Cuando se llena el área de espera, los clientes que llegan son rechazados y podrían, o no, regresar. En tales casos, las medidas de rendimiento obtenidas utilizando la suposición de un área de espera ilimitada pueden no ser válidas. Al modificar las fórmulas para calcular las medidas de rendimiento para tomar en cuenta el espacio limitado de espera, se pueden obtener resultados válidos.

CARACTERÍSTICAS CLAVE

Estos sistemas dan lugar a cuestiones adicionales:

- ✓ ¿Cuál es la probabilidad de que un cliente que llegue sea rechazado y se le niegue el servicio porque el área de espera está llena? A esta medida de rendimiento se le conoce como la *probabilidad de negación de servicio*, denotada con p_d.
- ✓ Cuando se efectúa un análisis económico, debe tomarse en consideración un tercer componente, un costo asociado con la pérdida de un cliente, junto con el costo por servidor y el costo por esperar.

Considere el problema enfrentado por National Public TV (NPTV).

EJEMPLO 13.3 EL PROBLEMA DE CAPACIDAD DE ESPERA LIMITADA DE NATIONAL PUBLIC TV El gerente de la estación local de NPTV, una red de televisión no lucrativa, está planeando un maratón telefónico (teletón) especial de cinco días para la obtención de fondos, y está tratando de determinar el tipo de sistema telefónico que debe alquilar para recibir las promesas de donaciones. La compañía telefónica local proporciona sistemas de 15 o de 20 líneas. Con cada sistema, se tiene disponible una opción de espera de 0, 5 o 10 llamadas, costos diarios totales dados a continuación:

SISTEMA	NÚMERO DE TELÉFONOS	LLAMADAS ESPERADAS	COSTO TOTAL ($/DÍA)
1	15	0	150
2	20	0	220
3	15	5	180
4	20	5	264
5	15	10	225
6	20	10	330

Como gerente de la estación, usted desea determinar el sistema más económico que podría utilizar. ■

Al igual que con cualquier sistema de colas, su primera tarea consiste en identificar los procesos de llegada y de servicio apropiados. En este caso, un proceso de llegada de Poisson y una tasa de servicio exponencial han resultado ser, históricamente, razonables para sistemas telefónicos. Suponga que su investigación revela los siguientes datos:

1. Tasa de llegadas = λ = 150 llamadas por hora.
2. Tasa de servicio por línea telefónica = μ = 12 llamadas por hora.

Para analizar el rendimiento de uno de estos seis sistemas diferentes, usted debe introducir estos datos en un paquete de computación capaz de proporcionar medidas para modelos M/M/c con capacidad de espera limitada. Los resultados obtenidos con el paquete STORM para el primer sistema, con 15 líneas y sin mantenimiento de llamadas, se muestran en la figura 13.15. Excepto por la última línea del informe, todas las medidas de rendimiento son interpretadas del mismo modo que en cualquier modelo

Formación de cola
EX13_3.DAT

```
        M/M/15/0 : M / M / C / K
      Q U E U E   S T A T I S T I C S

Number of identical servers . . . . . . . .      15
Mean arrival rate . . . . . . . . . . . . .   150.0000
Mean service rate per server . . . . . . .     12.0000
Waiting room capacity . . . . . . . . . . .        0

Mean server utilization (%) . . . . . . . .   74.9592
Expected number of customers in queue . . . .  *0.0000
Expected number of customers in system . . .   11.2439
Probability that a customer must wait . . . .   0.1005
Probability of service denial . . . . . . . .   0.1005
```

Figura 13.15 Medidas de rendimiento obtenidas con STORM para el problema M/M/15/0 de National Public TV.

56

M/M/c. Observe la medida de rendimiento de la probabilidad de negación de servicio (p_d), una nueva e importante estadística, en la última línea del informe. El valor de 0.1005 indica que con este sistema existe 10% de probabilidad de que un observador que llame obtenga señal de ocupado porque las 15 líneas están ocupadas. Este cliente puede llamar o no de nuevo. En el último caso, se pierden entradas.

Para realizar un análisis económico de tales sistemas, usted necesita saber el costo de los servidores y el costo de la espera. También necesita estimar el costo de perder un cliente cuando el espacio de espera está lleno. Para el problema de NPTV, estos tres componentes del costo se estiman de la manera siguiente:

1. *Costo por servidor:* Cada servidor corresponde a una línea telefónica. El costo total para NPTV puede convertirse en un costo por línea telefónica por hora. Suponga que el teletón se lleva a cabo durante ocho horas diarias. El costo asociado para el primer sistema es:

$$c_s = (\$150/\text{día})/(8 \text{ horas/día})/(15 \text{ líneas telefónicas})$$
$$= 1.25$$

Por consiguiente, con este sistema de 15 servidores (líneas telefónicas), este costo por hora es:

$$\text{Costo total de los servidores} = (\text{costo por servidor}) * (\text{número de servidores})$$
$$= c_s * c$$
$$= 1.25 * 15$$
$$= \$18.75 \text{ por hora}$$

2. *Costo de espera:* En este caso, no existe un costo directo correspondiente a un contribuyente que pierde tiempo en la línea para prometer una donación, de modo que:

$$c_w = 0$$

Así pues, el costo por hora de los clientes en el sistema es:

$$\left\{\begin{matrix}\text{costo total de}\\ \text{la espera}\end{matrix}\right\} = \left\{\begin{matrix}\text{costo de la}\\ \text{espera}\end{matrix}\right\} * \left\{\begin{matrix}\text{número de clientes}\\ \text{en el sistema}\end{matrix}\right\}$$
$$= c_w * L$$
$$= 0 * 11.2439$$
$$= \$0 \text{ por hora}$$

3. *Costo por pérdida de un cliente:* Aquí es necesario estimar cuánto dinero se pierde cuando una persona llama y obtiene una señal de ocupado y no puede hacer una contribución. La gerencia de NPTV sabe, por experiencias pasadas, que la donación promedio por llamada es de $50. Sin embargo, esta cantidad no siempre se pierde cuando un contribuyente no puede hacer la llamada, pues el 80% de ellos intentará llamar de nuevo. El costo por la pérdida de un cliente en este caso es, entonces:

$$c_d = (\text{por llamada}) * (\text{probabilidad de perder la llamada})$$
$$= 50 * 0.20$$
$$= \$10.00 \text{ por negación}$$

Esta cifra representa un costo de negación de servicio *por cada cliente*. Para calcular el costo total por hora, es necesario saber a *cuántos clientes* se les niega el servicio por hora. Recuerde que la tasa de llegadas es $\lambda = 150$ llamadas por hora y que la probabilidad de que a un cliente se le niegue el servicio es $p_d = 0.1005$ (véase la figura 13.15). En promedio, entonces, 150 * 0.1005 = 15.075 clientes no obtienen servicio cada hora. Así pues, el costo por hora de pérdida de clientes es:

$$\left\{\begin{matrix}\text{Costo total}\\ \text{por negación}\end{matrix}\right\} = \left\{\begin{matrix}\text{costo por}\\ \text{negación}\end{matrix}\right\} * \left\{\begin{matrix}\text{número de}\\ \text{llegadas}\end{matrix}\right\} * \left\{\begin{matrix}\text{probabilidad}\\ \text{de negación}\\ \text{de servicio}\end{matrix}\right\}$$
$$= c_d * \lambda * p_d$$
$$= 10 * 150 * 0.1005$$
$$= \$150.75 \text{ por hora}$$

Estos tres componentes de costo se suman para obtener el costo total por hora para el primer sistema con 15 líneas y sin capacidad de espera:

$$\text{Costo total} = (\text{costo de los servidores}) + (\text{costo de la espera}) + (\text{costo de la negación del servicio})$$
$$= (c_s * c) + (c_w * L) + (c_d * \lambda * p_d)$$
$$= (1.25 * 15) + (0 * 11.2439) + (10.00 * 150 * 0.1005)$$
$$= 18.75 + 0 + 150.75$$
$$= \$169.50 \text{ por hora}$$

Un análisis parecido se puede efectuar, ahora, para los restantes cinco sistemas telefónicos. Los resultados se resumen en la tabla 13.5. Observe el costo total por hora de cada sistema en la línea final de dicha tabla. Los costos indican que resulta más económico tener un sistema con 20 líneas y capacidad de espera de hasta cinco llamadas, con un costo total por hora de $34.90 (sistema 4).

TABLA 13.5 *Análisis económico de los sistemas de seis teléfonos para el problema de NPTV*

	SISTEMA					
	1	2	3	4	5	6
Número de filas	15	20	15	20	15	20
Capacidad de espera	0	0	5	5	10	10
c_s	1.25	1.375	1.50	1.65	1.875	2.0625
c_w	0.00	0.00	0.00	0.00	0.00	0.00
c_d	10.00	10.00	10.00	10.00	10.00	10.00
L	11.235	12.331	12.722	12.527	13.565	12.555
p_d	0.1005	0.0125	0.0311	0.0013	0.0114	0.0001
Costo total ($/hr)	169.50	47.78	69.08	34.90	45.26	41.43

13.6.3 Un sistema de colas con una distribución de tiempo de servicio general (M/G/c)

En todos los sistemas de colas analizados hasta este punto, el tiempo de servicio se supone que sigue una distribución exponencial con una tasa de servicio media conocida de μ. En algunos modelos, sin embargo, esta suposición puede no ser válida. Un ejemplo extremo es cuando el tiempo de servicio es determinístico, esto es, cuando cada cliente requiere la misma cantidad conocida de tiempo de servicio (como en el caso de una línea de ensamblaje con un ciclo de tiempo fijo). Incluso cuando el tiempo de servicio es probabilístico, usted puede no conocer su distribución o ésta puede no ser exponencial. En tales casos, se puede utilizar un análisis de colas apropiado mediante la identificación del proceso de servicio como "general" (G).

CARACTERÍSTICAS CLAVE

Para obtener medidas de rendimiento en cuanto a tales sistemas, además de la tasa de llegada promedio de λ, usted debe estimar:

- ✓ La cantidad promedio de tiempo por servicio.
- ✓ La desviación estándar del tiempo de servicio, que proporciona una medida de su variabilidad. (Observe que una desviación estándar de cero corresponde a un tiempo de servicio determinístico.)

Considere el problema de la división Los Álamos de la Oficina de Control de Vehículos de Motor de Texas.

Formación de cola
EX13_4.DAT

EJEMPLO 13.4 EL PROBLEMA DE COLAS DE LA OFICINA DE CONTROL DE VEHÍCULOS DE MOTOR DE TEXAS La división de Los Álamos actualmente tiene tres servidores públicos que procesan el registro de automóviles. Recientemente, han recibido quejas de los clientes que tienen que esperar demasiado durante la hora del almuerzo, de 11:30 a 13:30 horas. Para minimizar el problema, usted, como administrador de la oficina, está tratando de determinar cuántos empleados adicionales debe contratar para este periodo de dos horas, de modo que el tiempo de espera sea menor a los 10 minutos. ■

La llegada de clientes podría suponerse, razonablemente, que sigue un proceso de Poisson. Basándose en datos históricos, usted estima que la tasa promedio de llegadas es $\lambda = 46$ personas por hora. A pesar de que no tiene certeza sobre la distribución del tiempo de servicio, un estudio del tiempo ha revelado que cada servidor necesita un promedio de cinco minutos (0.08333 horas) para atender a un cliente, con una desviación estándar de dos minutos (0.0333 horas).

Estos datos indican que cada servidor puede procesar un promedio de $\mu = 12$ clientes por hora. Así pues, para manejar la estimación pico de 46 clientes por hora, es decir, asegurar que la tasa total de servicio, $c * \mu$, excede a la tasa total de llegades, λ, es necesario tener al menos cuatro ventanillas en servicio. Al introducir estos datos en el paquete de software STORM, utilizando un modelo M/G/4 (para indicar que la distribución de tiempo de servicio no es exponencial) produce las medidas de rendimiento dadas en la figura 13.16. Usted puede ver que existe un promedio de 12 clientes en la cola y que cada uno de ellos tiene que esperar un promedio de 0.2636 horas (aproximadamente 16 minutos) antes de ser atendidos. En total, cada cliente tiene que invertir 0.3470 horas (aproximadamente 21 minutos) en la oficina.


```
M/G/4 : M / G / C
QUEUE STATISTICS

Number of identical servers . . . . . .          4
Mean arrival rate . . . . . .            46.0000
Mean service rate per server . . . . . .  12.0000
Standard deviation of service time . . . . .  0.0333

Mean server utilization (%) . . . . . .  95.8330
Expected number of customers in queue . . . .  12.1272
Expected number of customers in system . . .  15.9605
Probability that a customer must wait . . . .  0.9092
Expected time in the queue . . . . . .   0.2636
Expected time in the system . . . . . .  0.3470
```

Figura 13.16 Medidas de rendimiento obtenidas con STORM para el problema M/G/4 de Texas BMV.

Este nivel de servicio no es aceptable porque el tiempo promedio de espera de 16 minutos excede al objetivo de 10 minutos. Por consiguiente, es necesario tener al menos cinco ventanillas en funcionamiento. Al cambiar el número de servidores de 4 a 5 y resolver el nuevo modelo M/G/5, se obtienen los resultados mostrados en la figura 13.17. Usted puede ver que con cinco ventanillas abiertas, el tiempo promedio de espera en la cola disminuye a 0.0204 horas, un poco más de un minuto. Esto está completamente dentro del propósito de los diez minutos, de modo que usted decide aumentar el número de servidores públicos de 3 a 5, durante el tiempo de almuerzo.

En la presente sección, usted ha visto cómo la computadora obtiene medidas de rendimiento para algunos de los sistemas de colas que se presentan más comúnmente

```
M/G/5 : M / G / C
QUEUE STATISTICS

Number of identical servers . . . . . .          5
Mean arrival rate . . . . . .            46.0000
Mean service rate per server . . . . . .  12.0000
Standard deviation of service time . . . . .  0.0333

Mean server utilization (%) . . . . . .  76.6664
Expected number of customers in queue . . . .   0.9369
Expected number of customers in system . . .   4.7702
Probability that a customer must wait . . . .  0.4916
Expected time in the queue . . . . . .   0.0204
Expected time in the system . . . . . .  0.1037
```

Figura 13.17 Medidas de rendimiento obtenidas con STORM para el problema M/G/5 de Texas BMV.

58

y que son distintos a los sistemas *M/M/c*. Estos otros modelos incluyen un sistema *M/M/c* con población finita, un sistema *M/M/c* con capacidad de espera limitada y un sistema *M/G/c* en el cual el tiempo de servicio sigue una distribución general cuyas media y desviación estándar deben estimarse. A continuación se analizarán otras cuestiones sobre las que un administrador debe preocuparse cuando analiza o diseña sistemas de colas.

CONSIDERACIONES GERENCIALES COMPLEMENTARIAS

Usted ha aprendido a diseñar y analizar algunos sistemas de colas típicos, mediante la evaluación de varias medidas de rendimiento y mediante la realización de un análisis económico apropiado. En esta sección, se revisan varias cuestiones importantes con respecto a dicho análisis, desde un punto de vista gerencial.

Elección de un modelo adecuado

A pesar de que el objetivo terminal de un modelo de colas es evaluar varias medidas de rendimiento, la capacidad de calcular estas medidas está restringida a un número limitado de modelos diferentes (como los sistemas *M/M/c* o *M/G/c*). El problema particular de colas que esté tratando puede no adaptarse a los modelos que su paquete de computación es capaz de manejar. En estos casos, usted debe hacer algo de lo siguiente:

1. Obtener un paquete de computación que sea capaz de analizar su modelo.
2. Localizar las fórmulas adecuadas de algún libro especializado en teoría de colas para calcular las medidas de rendimiento necesarias, y esto no siempre puede ser posible.
3. Hacer algunas suposiciones acerca de su problema que le permitan aproximarlo con uno de los modelos de colas para los cuales las fórmulas de las medidas de rendimiento están disponibles.

Tome en consideración una máquina embotelladora de bebidas no alcohólicas. Las botellas vacías llegan al mismo tiempo y son llenadas en grupos de 24 a la vez. A diferencia de los modelos anteriores, en los que cada cliente de la cola es atendido de manera individual, aquí la máquina procesa un grupo de 24 botellas de manera simultánea. Si hay menos de 24 botellas en la cola, la máquina debe esperar.

Una manera de aproximación para este sistema consiste en identificar cada grupo de 24 botellas como un solo "cliente". Esto se hace modificando el proceso de llegada para expresar una tasa de llegada asociada con *cada lote*. Por ejemplo, si originalmente las botellas llegan con una rapidez promedio de 48 botellas por minuto, entonces cada grupo de 24 botellas llega con una rapidez aproximada de dos lotes por minuto. Similarmente, el tiempo de servicio debe expresarse como la cantidad de tiempo necesaria para que la máquina llene las 24 botellas, no una de manera individual. Las medidas de rendimiento asociadas deben interpretarse de acuerdo con lo anterior. Por ejemplo, si el número promedio de clientes que esperan en la cola es de, digamos, 5, esto significa que en promedio hay aproximadamente 5 * 24 = 120 botellas esperando para ser llenadas.

Cuando utilice un modelo de aproximación, tenga en cuenta que las medidas de rendimiento que usted obtenga pueden *no* ser lo que se ve en la práctica. Antes de instrumentar decisiones basadas en resultados modelados, deberá intentar validarlos. Para validar la aproximación utilizada en el ejemplo de la embotelladora, suponga que el modelo tiene como resultado un promedio de cinco clientes en el sistema. Si esta aproximación es válida, usted deberá, efectivamente, observar en la práctica un promedio de aproximadamente 120 botellas esperando para ser llenadas. Únicamente si usted determina que el modelo de aproximación es válido, entonces deberá considerar la instrumentación de las decisiones, basándose en las medidas de rendimiento obtenidas con el modelo.

Sistemas de colas adicionales

Como usted ha aprendido, un sistema de colas tiene muchos componentes y características individuales, incluyendo un proceso de llegada y de servicio, una disciplina de colas, etcétera. Los resultados específicos de este capítulo se aplican *exclusivamente* a un sistema con las siguientes características:

1. Los clientes llegan y continúan esperando en una sola fila hasta que son atendidos, uno a la vez.
2. La disciplina de cola es primero en entrar, primero en salir, es decir, el primer cliente que llega es el primero que es atendido.
3. Una vez que el cliente es atendido, abandona para siempre el sistema, es decir, solamente existe una estación de trabajo.

En muchas aplicaciones, sin embargo, estas características pueden variar, por ejemplo:

1. Los clientes *pueden llegar en lotes* y/o esperar en *múltiples filas*. En este último caso, un cliente que llega, probablemente se forma en la fila más corta, pero más adelante puede *maniobrar* entre las colas, a medida que cambia su longitud.
2. Los clientes que esperan pueden *ser atendidos en grupos*, en vez de hacerlo de manera individual. El grupo puede tener un tamaño fijo, como en el caso del ejemplo anterior de la embotelladora, o puede variar de tamaño, como cuando un autobús recoge a los pasajeros que esperan.
3. La elección de cuál cliente se va a atender puede ser diferente al procedimiento de primero en entrar, primero en salir, como en la sala de urgencias de un hospital.
4. Los clientes que llegan, al ver la longitud de la cola, pueden decidir no esperarse y se retiran. Alternativamente, pueden decidir esperar en la cola durante un tiempo y luego retirarse, por lo que *renuncian*.
5. Algunos clientes, después de ser atendidos, *pueden no abandonar el sistema*, sino que, en vez de ello, pasar a otra estación de trabajo para otro tipo de atención. Incluso pueden regresar a la primera estación de trabajo una segunda ocasión. Esto tiene como resultado un sistema con una *red* de estaciones de trabajo en lugar de una sola estación.

Estas características variables traen como consecuencia sistemas más complejos que, en muchos casos, no pueden ser analizados de manera directa porque sus fórmulas matemáticas no pueden derivarse. Para tales situaciones, se pueden uti-

lizar las técnicas de las ciencias de la administración conocidas como *simulación*, y que se presentan en los capítulos 14 y 15, para fines de diseño y de análisis.

Análisis de sensibilidad

Su paquete de computación puede ser capaz de manejar el modelo que está proponiendo y proporcionarle medidas de rendimiento exactas. Sin embargo, debe mantener en su mente que los resultados que obtenga dependen de estimaciones de las tasas de llegada y de servicio. Estas estimaciones pueden ser o no precisas, de modo que debe aprender a hacerse preguntas del tipo "¿qué sucede si . . . ? Por ejemplo, en un sistema de colas M/M/c, usted podría haber estimado que 50 clientes llegan para ser atendidos cada hora. Basándose en esta información y en otros datos de costos, suponga que ha determinado que el sistema de menor costo requiere cinco servidores. Antes de instrumentar esta decisión, deberá hacerse algunas de las preguntas siguientes:

1. ¿Cuántos servidores deben contratarse si la tasa de llegada es de 52, 55 o 60 clientes por hora?
2. ¿En qué cantidad debe aumentarse o disminuirse la tasa de llegada, dejando todos los demás datos igual, antes de que deje de ser óptimo el tener cinco servidores?
3. ¿En qué cantidad debe aumentarse o disminuirse la tasa de servicio, dejando todos los demás datos igual, antes de que deje de ser óptimo el tener cinco servidores?
4. ¿En qué cantidad puede aumentar o disminuir el costo de cada servidor antes de que deje de ser óptimo el tener cinco servidores?

Esta preguntas se responden calculando repetidamente las medidas de rendimiento y los análisis económicos para el modelo, cambiando en cada ocasión los datos de interés. Si los resultados son sensibles a un valor de dato particular, ese valor deberá recibir particular atención. Su tiempo puede ser bien invertido si centra su atención en este valor, con el objetivo de obtener una estimación más precisa.

Los análisis de sensibilidad pueden llevarse a cabo en prácticamente todas las medidas de rendimiento, dependiendo de cuál o cuáles de ellas le preocupan más. Por ejemplo, incluso con una cola M/M/1, usted puede hacerse (y responderse) cada una de las siguientes preguntas:

1. ¿En qué cantidad debería incrementarse la tasa de llegadas antes de que la longitud promedio y/o el tiempo de espera promedio en la cola excedan algún límite aceptable?
2. ¿En qué cantidad podría disminuirse la tasa de servicio antes de que la probabilidad de que haya más de 10 clientes en el sistema exceda un nivel aceptable?

Solamente después de un cuidadoso análisis de estas preguntas de sensibilidad debería instrumentarse una solución. Pero, incluso entonces, la instrumentación debería revisarse muy de cerca para asegurarse de que los resultados obtenidos en la práctica son lo que se esperaba del modelo.

Análisis de equilibrio

Al tratar de determinar el "mejor" sistema de colas, debe centrarse en una medición de rendimiento en particular. Por ejemplo, puede desear un sistema que tenga una alta utilización, esto es, la fracción de tiempo que cada servidor está ocupado debería ser cercana a uno. Puede lograr este resultado si tiene menos servidores, de modo que los que tenga estén ocupados más tiempo, o podría intentar un incremento de la tasa de llegada, de manera que la cola casi nunca esté vacía. Por ejemplo, en un ambiente de producción, podría asignar al sistema más tareas con más rapidez.

En la mayoría de los casos, sin embargo, típicamente, encontrará que existe un cierto intercambio. A medida que mejora una medida de rendimiento, otras se deterioran. Por ejemplo, en la tabla 13.6 se muestra el contraste de la utilización contra el tiempo de espera, cuando se cambia la tasa de llegadas, λ, de un sistema de colas M/M/1, con una tasa de servicio $\mu = 20$ clientes por hora. Conforme la tasa de llegadas, λ, aumenta de 10 a 18, la utilización aumenta de 50% a 90%, un cambio deseable. Sin embargo, simultáneamente, el tiempo de espera de un cliente en la cola aumenta de 0.05 horas (3 minutos) a 0.45 horas (27 minutos). Como administrador, usted debe escoger un valor de λ que produzca una utilización aceptable de sus servidores y un tiempo de espera razonable para sus clientes.

TABLA 13.6 Equilibrio entre la utilización y el tiempo promedio de espera para un sistema de colas M/M/1 con $\mu = 20$ por hora

λ	ρ	UTILIZACIÓN	TIEMPO PROMEDIO EN LA COLA (HR)
10	0.5	0.50	0.050
12	0.6	0.60	0.075
14	0.7	0.70	0.116
16	0.8	0.80	0.200
18	0.9	0.90	0.450

RESUMEN

En el presente capítulo usted ha aprendido cómo aplicar la teoría de colas en el diseño y análisis de sistemas en que los productos llegan a una estación, esperan en una fila, obtienen algún tipo de servicio y luego salen del sistema. También aprendió que analizar estos modelos de colas implica el cálculo de algunas de las siguientes medidas de rendimiento:

1. El tiempo promedio de espera que un cliente que llega tiene que invertir en una fila antes de ser atendido (W_q).
2. El tiempo promedio que un cliente invierte en el sistema (W).
3. La longitud promedio en la cola (L_q).
4. El número promedio de clientes que hay en el sistema (L).
5. La probabilidad de que un cliente que llega tenga que esperar a ser atendido (p_w).
6. La utilización promedio de un servidor (U).
7. La probabilidad de que haya n clientes en el sistema (P_n).

60

Las fórmulas necesarias para calcular estas medidas dependen de las características específicas de los siguientes cuatro componentes:

1. La población de clientes.
2. El proceso de llegada.
3. El proceso y la disciplina de colas.
4. El proceso de servicio.

También ha visto cómo utilizar la computadora para obtener estas medidas para muchos modelos de uso común y cómo interpretar y utilizar los resultados para tomar decisiones administrativas. El hacerlo, a menudo, implica la evaluación de sistemas alternativos. La elección puede basarse en el logro de algún nivel deseable de rendimiento o en un análisis económico que asocie un costo global a cada alternativa, lo cual le permite escoger la de menor costo.

Las fórmulas para calcular las diferentes medidas de rendimiento están disponibles solamente para ciertos métodos sencillos, como el *M/M/c* o el *M/G/c*. Si su sistema es demasiado complejo para ser aproximado por uno de los modelos para los cuales las fórmulas matemáticas están disponibles, a menudo puede utilizarse la técnica de simulación, que se presenta en los capítulos 14 y 15.

CHAPTER 9

OUTPUT DATA ANALYSIS FOR A SINGLE SYSTEM

Recommended sections for a first reading: 9.1 through 9.3, 9.4.1, 9.4.3, 9.5.1, 9.5.2, 9.8

9.1 INTRODUCTION

In many simulation studies a great deal of time and money is spent on model development and programming, but little effort is made to analyze the simulation output data appropriately. As a matter of fact, a very common mode of operation is to make a single simulation run of somewhat arbitrary length and then to treat the resulting simulation estimates as the "true" model characteristics. Since random samples from probability distributions are typically used to drive a simulation model through time, these estimates are just particular realizations of random variables that may have large variances. As a result, these estimates could, in a particular simulation run, differ greatly from the corresponding true characteristics for the model. The net effect is, of course, that there could be a significant probability of making erroneous inferences about the system under study.

Historically, there are several reasons why output data analyses have not been conducted in an appropriate manner. First, users often have the unfortunate impression that simulation is just an exercise in computer programming,

albeit a complicated one. Consequently, many simulation "studies" begin with heuristic model building and coding, and end with a single run of the program to produce "the answers." In fact, however, a simulation is a computer-based statistical sampling experiment. Thus, if the results of a simulation study are to have any meaning, appropriate statistical techniques must be used to design and analyze the simulation experiments. A second reason for inadequate statistical analyses is that the output processes of virtually all simulations are nonstationary and autocorrelated (see Sec. 5.5.3). Thus, classical statistical techniques based on IID observations are not directly applicable. At present, there are still several output-analysis problems for which there is no completely accepted solution, and the methods that are available are often complicated to apply. Another impediment to obtaining precise estimates of a model's true parameters or characteristics is the cost of the computer time needed to collect the necessary amount of simulation output data. Indeed, there are situations where an appropriate statistical procedure is available, but the cost of collecting the quantity of data dictated by the procedure is prohibitive. This latter difficulty is becoming less severe since many analysts now have their own high-speed microcomputers or engineering work stations. These computers are relatively inexpensive to buy and can be run overnight or on weekends to produce large amounts of simulation output data, at essentially zero marginal cost.

We now describe more precisely the random nature of simulation output. Let $Y_1, Y_2, \ldots$ be an output stochastic process (see Sec. 4.3) from a *single* simulation run. For example, Y_i might be the throughput (production) in the ith hour for a manufacturing system. The Y_i's are random variables that will, in general, be neither independent nor identically distributed. Thus, most of the formulas of Chap. 4, which assume independence [e.g., the confidence interval given by (4.12)], do not apply *directly*.

Let $y_{11}, y_{12}, \ldots, y_{1m}$ be a realization of the random variables $Y_1, Y_2, \ldots, Y_m$ resulting from making a simulation run of length m observations using the random numbers $u_{11}, u_{12}, \ldots$. (The ith random number used in the jth run is denoted u_{ji}.) If we run the simulation with a different set of random numbers $u_{21}, u_{22}, \ldots$, then we will obtain a different realization $y_{21}, y_{22}, \ldots, y_{2m}$ of the random variables $Y_1, Y_2, \ldots, Y_m$. (The two realizations are not the same since the different random numbers used in the two runs produce different samples from the input probability distributions.) In general, suppose that we make n independent replications (runs) of the simulation (i.e., different random numbers are used for each replication, the statistical counters are reset at the beginning of each replication, and each replication uses the same initial conditions; see Sec. 9.4.3) of length m, resulting in the observations:

$$
\begin{array}{ccc}
y_{11}, \ldots, & y_{1i}, \ldots, & y_{1m} \\
y_{21}, \ldots, & y_{2i}, \ldots, & y_{2m} \\
\vdots & \vdots & \vdots \\
y_{n1}, \ldots, & y_{ni}, \ldots, & y_{nm}
\end{array}
$$

69

The observations from a particular replication (row) are clearly not IID. However, note that $y_{1i}, y_{2i}, \ldots, y_{ni}$ (from the ith column) are IID observations of the random variable Y_i, for $i = 1, 2, \ldots, m$. This *independence across runs* (see Prob. 9.1) is the key to the relatively simple output-data-analysis methods described in later sections of this chapter. Then, roughly speaking, the goal of output analysis is to use the observations y_{ji} ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$) to draw inferences about the (distributions of the) random variables $Y_1, Y_2, \ldots, Y_m$. For example, $\bar{y}_i(n) = \Sigma_{j=1}^{n} y_{ji}/n$ is an unbiased estimate of $E(Y_i)$.

Example 9.1. Consider a bank with five tellers and one queue, which opens its doors at 9 A.M., closes its doors at 5 P.M., but stays open until all customers in the bank at 5 P.M. have been served. Assume that customers arrive in accordance with a Poisson process at rate 1 per minute (i.e., IID exponential interarrival times with mean 1 minute), that service times are IID exponential random variables with mean 4 minutes, and that customers are served in a FIFO manner. Table 9.1 shows several typical output statistics from 10 independent replications of a simulation of the bank, assuming that no customers are present initially. Note that results from various replications can be quite different. Thus, one run clearly does not produce "the answers."

Our goal in this chapter is to discuss methods for statistical analysis of simulation output data and to present the material with a practical focus that should be accessible to a reader having a basic understanding of probability and statistics. (Reviewing Chap. 4 might be advisable before reading this chapter.) We will discuss what we believe are all the important methods for output analysis; however, the emphasis will be on statistical procedures that are relatively easy to understand and implement, have been shown to perform well in practice, and have applicability to real-world problems.

TABLE 9.1
Results for 10 independent replications of the bank model

Replication	Number served	Finish time (hours)	Average delay in queue (minutes)	Average queue length	Proportion of customers delayed < 5 minutes
1	484	8.12	1.53	1.52	0.917
2	475	8.14	1.66	1.62	0.916
3	484	8.19	1.24	1.23	0.952
4	483	8.03	2.34	2.34	0.822
5	455	8.03	2.00	1.89	0.840
6	461	8.32	1.69	1.56	0.866
7	451	8.09	2.69	2.50	0.783
8	486	8.19	2.86	2.83	0.782
9	502	8.15	1.70	1.74	0.673
10	475	8.24	2.60	2.50	0.779

In Secs. 9.2 and 9.3 we discuss types of simulations with regard to output analysis, and also measures of performance or parameters θ for each type. Sections 9.4 through 9.6 show how to get a point estimator $\hat{\theta}$ and confidence interval for each type of parameter θ, with the confidence interval typically requiring an estimate of the variance of $\hat{\theta}$, namely, $\widehat{\mathrm{Var}}(\hat{\theta})$. Each of the analysis methods discussed may suffer from one or both of the following problems:

1. $\hat{\theta}$ is not an unbiased estimator of θ, that is, $E(\hat{\theta}) \neq \theta$; see, for example, Sec. 9.5.2
2. $\widehat{\mathrm{Var}}(\hat{\theta})$ is not an unbiased estimator of $\mathrm{Var}(\hat{\theta})$; see, for example, Sec. 9.5.3

Section 9.7 extends the above analyses to confidence-interval construction for several different parameters simultaneously. Finally, in Sec. 9.8 we show how time plots of important variables may provide insight into a system's dynamic behavior.

We will not attempt to give every reference on the subject of output data analysis, since a very comprehensive set of references was given in the survey paper by Law (1983). Also see the book chapter by Welch (1983).

9.2 TRANSIENT AND STEADY-STATE BEHAVIOR OF A STOCHASTIC PROCESS

Consider the output stochastic process $Y_1, Y_2, \ldots$. Let $F_i(y|I) = P(Y_i \leq y|I)$ for $i = 1, 2, \ldots$, where y is a real number and I represents the initial conditions used to start the simulation at time 0. [The conditional probability $P(Y_i \leq y|I)$ is the probability that the event $\{Y_i \leq y\}$ occurs *given* the initial conditions I.] For a manufacturing system, I might specify the number of jobs present, and whether each machine is busy or idle, at time 0. We call $F_i(y|I)$ the *transient distribution* of the output process at (discrete) time i for initial conditions I. Note that $F_i(y|I)$ will, in general, be different for each value of i and each set of initial conditions I. The density functions for the transient distributions corresponding to the random variables $Y_{i_1}, Y_{i_2}, Y_{i_3}$, and Y_{i_4} are shown in Fig. 9.1 for a particular set of initial conditions I and increasing time indices i_1, i_2, i_3, and i_4, where it is assumed that the random variable Y_{i_j} has density function $f_{Y_{i_j}}$. The density $f_{Y_{i_j}}$ specifies how the random variable Y_{i_j} can vary from one replication to another.

For fixed y and I, the probabilities $F_1(y|I), F_2(y|I), \ldots$ are just a sequence of numbers. *If* $F_i(y|I) \to F(y)$ as $i \to \infty$ for all y and for any initial conditions I, then $F(y)$ is called the *steady-state distribution* of the output process $Y_1, Y_2, \ldots$. Strictly speaking, the steady-state distribution $F(y)$ is only obtained in the limit as $i \to \infty$. In practice, however, there will often be a finite time index, say, $k + 1$, such that the distributions from this point on will be approximately the same as each other; "steady state" is figuratively said to start at time $k + 1$ as shown in Fig. 9.1. Note that steady state does *not* mean

70

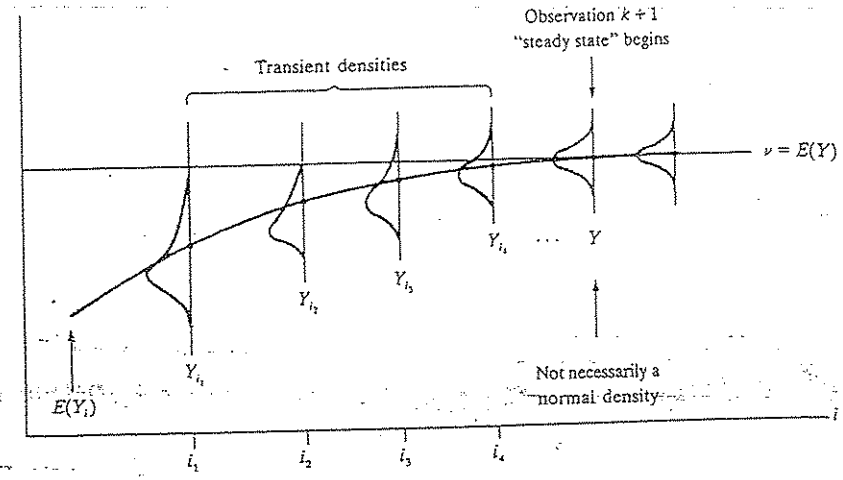


FIGURE 9.1
Transient and steady-state density functions for a particular stochastic process $Y_1, Y_2, \ldots$ and initial conditions I.

that the random variables $Y_{k+1}, Y_{k+2}, \ldots$ will all take on the same value in a particular simulation run; rather, it means that they will all have approximately the same *distribution*. Furthermore, these random variables will not be independent, but will approximately constitute a covariance-stationary stochastic process (see Sec. 4.3). See Welch (1983) for an excellent discussion of transient and steady-state distributions.

The steady-state distribution $F(y)$ does not depend on the initial conditions I; however, the rate of convergence of the transient distributions $F_i(y|I)$ to $F(y)$ does, as the following example shows.

Example 9.2. Consider the stochastic process $D_1, D_2, \ldots$ for the $M/M/1$ queue with $\rho = 0.9$ ($\lambda = 1$, $\omega = 10/9$), where D_i is the delay in queue of the ith customer. In Fig. 9.2 we plot the convergence of the transient mean $E(D_i)$ to the steady-state mean $d = E(D) = 8.1$ as i gets large for various values of number in system at time 0, s. (The random variable D has the steady-state delay in queue distribution.) Note that the convergence of $E(D_i)$ to d is, surprisingly, much faster for $s = 15$ than for $s = 0$ (see Prob. 9.11). The values for $E(D_i)$ were derived in Kelton and Law (1985); see also Kelton (1985) and Murray and Kelton (1988). The distribution function of D is given by (4.14) in App. 4A.

Example 9.3. Consider the stochastic process $C_1, C_2, \ldots$ for the inventory problem of Example 4.23, where C_i is the total cost in the ith month. In Fig. 9.3 we plot the convergence of $E(C_i)$ to the steady-state mean $c = E(C) = 112.11$ [see Wagner (1969, p. A19)] as i gets large for an initial inventory level of 57. Note that the convergence is clearly not monotone.

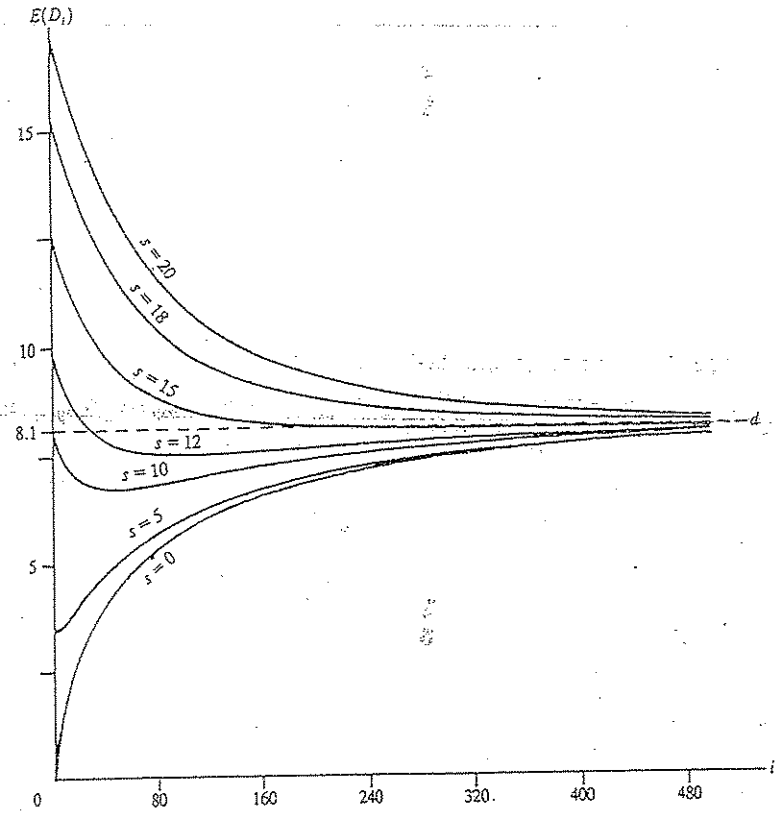


FIGURE 9.2
$E(D_i)$ as a function of i and the number in system at time 0, s, for the $M/M/1$ queue with $\rho = 0.9$.

In Examples 9.2 and 9.3 we plotted the convergence of the *expected value* $E(Y_i)$ to the steady-state mean $E(Y)$. It should be remembered, however, that the entire *distribution* of Y_i is also converging to the distribution of Y as i gets large.

9.3 TYPES OF SIMULATIONS WITH REGARD TO OUTPUT ANALYSIS

The options available in designing and analyzing simulation experiments depend on the type of simulation at hand, as depicted in Fig. 9.4. Simulations may be either terminating or nonterminating, depending on whether there is an obvious way for determining run length. Furthermore, measures of perfor-

71

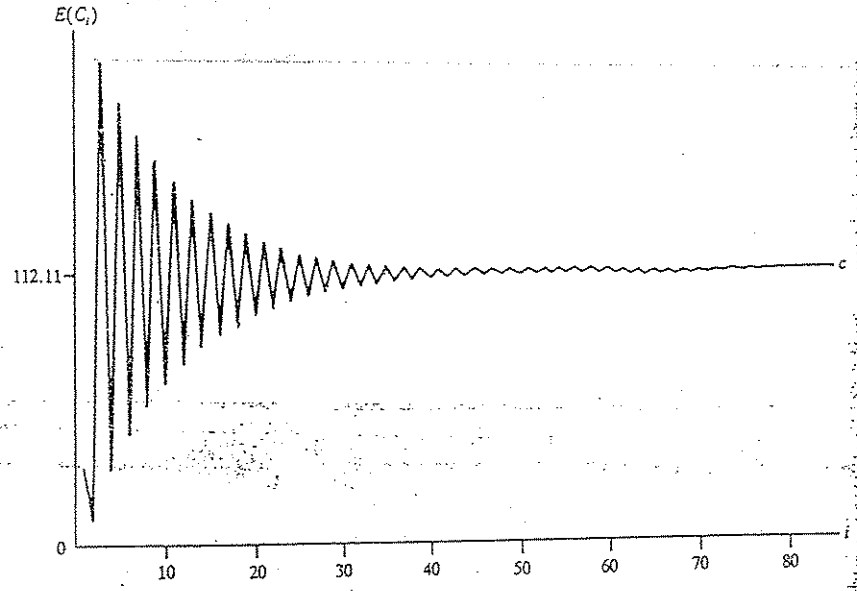


FIGURE 9.3
$E(C_i)$ as a function of i for the (s,S) inventory system.

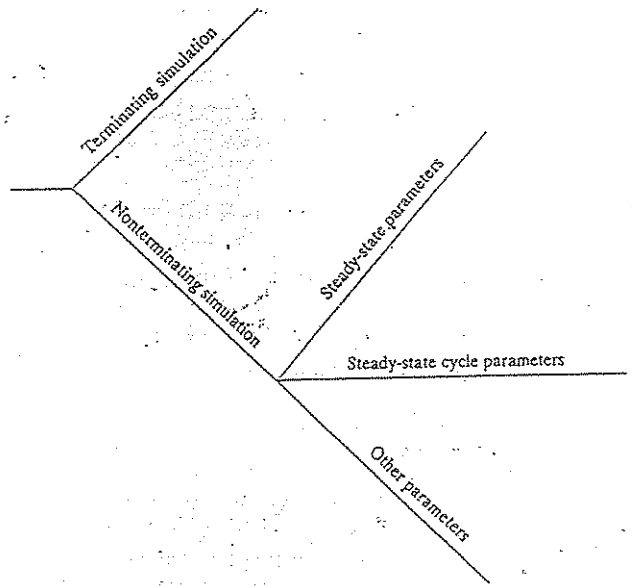


FIGURE 9.4
Types of simulations with regard to output analysis.

mance or parameters for nonterminating simulations may be of several types, as shown in the figure. These concepts are defined more precisely below.

A *terminating simulation* is one for which there is a "natural" event E that specifies the length of each run (replication). Since different runs use independent random numbers and the same initialization rule, this implies that comparable random variables from the different runs are IID (see Sec. 9.4). The event E often occurs at a time point beyond which no useful information is obtained or at a time point when the system is "cleaned out." It is specified before any runs are made, and the time of occurrence of E for a particular run may be a random variable. Since the *initial conditions for a terminating simulation generally affect the desired measures of performance*, these conditions should be representative of those for the actual system (see Sec. 9.4.3).

Example 9.4. A retail/commercial establishment, e.g., a bank, closes each evening. If the establishment is open from 9 to 5, the objective of a simulation might be to estimate some measure of the quality of customer service over the period beginning at 9 A.M. and ending when the last customer who entered before the doors closed at 5 P.M. has been served. In this case $E = \{$at least 8 hours of simulated time have elapsed and the system is empty$\}$, and the initial conditions for the simulation are the number of customers present at time 0 (see Sec. 9.4.3).

Example 9.5. Consider a military ground confrontation between a blue force and a red force. Relative to some initial force strengths, the goal of a simulation might be to determine the (final) force strengths when the battle ends. In this case $E = \{$either the blue force or the red force has "won" the battle$\}$. An example of a condition that would end the battle is one side losing 30 percent of its force, since this side would no longer be considered viable. The choice of initial conditions, e.g., the number of troops and tanks for each force, for the simulation is generally not a problem here, since they are specified by the military scenario under consideration.

Example 9.6. An aerospace manufacturer receives a contract to produce 100 airplanes, which must be delivered within 18 months. The company would like to simulate various manufacturing configurations to see which one can meet the delivery deadline at least cost. In this case $E = \{$100 airplanes have been completed$\}$.

Example 9.7. Consider a manufacturing company that operates 16 hours a day (two shifts) with work in process carrying over from one day to the next. Would this qualify as a terminating simulation with $E = \{$16 hours of simulated time have elapsed$\}$? No, since this manufacturing operation is essentially a continuous process, with the ending conditions for one day being the initial conditions for the next day.

Example 9.8. A company that sells a single product would like to decide how many items to have in inventory during a planning horizon of 120 months (see Sec. 1.5). Given some initial inventory level, the objective might be to determine how much to order each month so as to minimize the expected average cost per month of operating the inventory system. In this case $E = \{$120 months have been simulated$\}$, and the simulation is initialized with the current inventory level.

72

A *nonterminating simulation* is one for which there is no natural event E to specify the length of a run. A measure of performance for such a simulation is said to be a *steady-state parameter* if it is a characteristic of the steady-state distribution of some output stochastic process $Y_1, Y_2, \ldots$. In Fig. 9.1, if the random variable Y has the steady-state distribution, then we might be interested in estimating the steady-state mean $\nu = E(Y)$.

Example 9.9. Consider a company that is going to build a new manufacturing system and would like to determine the long-run (steady-state) mean hourly throughput of their system after it has been running long enough for the workers to know their jobs and for mechanical difficulties to have been worked out. Assume that:

(*a*) The system will operate 16 hours a day for 5 days a week.
(*b*) There is negligible loss of production at the end of one shift or at the beginning of the next shift (see Prob. 9.3).
(*c*) There are no breaks (e.g., lunch) that shut down production at specified times each day.

This system could be simulated by "pasting together" 16-hour days, thus ignoring the system idle time at the end of each day and on the weekend. Let N_i be the number of parts manufactured in the ith hour. If the stochastic process $N_1, N_2, \ldots$ has a steady-state distribution with corresponding random variable N, then we are interested in estimating the mean $\nu = E(N)$ (see Prob. 9.4).

It should be mentioned that stochastic processes for most *real* systems do not have steady-state distributions, since the characteristics of the system change over time. For example, in a manufacturing system the production-scheduling rules and the facility layout (e.g., number and location of machines) may change from time to time. On the other hand, a simulation model (which is an abstraction of reality) may have steady-state distributions, since characteristics of the *model* are often assumed not to change over time.

If, in Example 9.9, the manufacturing company wanted to know the time required for the system to go from startup to operating in a "normal" manner, this would be a terminating simulation with terminating event E = {simulated system is running "normally"} (if such can be defined). Thus, a simulation for a particular system might be either terminating or nonterminating, depending on the objectives of the simulation study.

Example 9.10. Consider a simulation model for a computer (or communication) system that does not currently exist. Since there are typically no representative data available on the arrival mechanism for jobs, it is common to assume that jobs arrive in accordance with a Poisson process with *constant* rate equal to the *predicted* arrival rate of jobs during the period of peak loading. (When the system is actually built, the arrival rate will vary as a function of time and the period of peak loading may be relatively short.) Since the state of the system during "normal operation" is unknown, initial conditions must be chosen somewhat arbitrarily (e.g., no jobs present at time 0). Then the goal is to run the simulation long enough so that the arbitrary choice of initial conditions is no longer having a significant effect on the estimated measures of performance (e.g., mean response time for a job).

In performing the above steady-state analysis of the proposed computer system, we are essentially trying to determine how the system will respond to a peak load of infinite duration. If, however, the peak period in the actual system is short or if the arrival rate before the peak period is considerably lower than the peak rate, our analysis may overestimate the congestion level during the peak period in the system. This might result in purchasing a computer configuration that is more powerful than actually needed.

Consider a stochastic process $Y_1, Y_2, \ldots$ for a nonterminating simulation that does not have a steady-state distribution. Suppose that we divide the time axis into equal-length, contiguous time intervals called *cycles*. (For example, in a manufacturing system a cycle might be an 8-hour shift.) Let Y_i^C be a random variable defined on the ith cycle, and assume that $Y_1^C, Y_2^C, \ldots$ are comparable. Suppose that the process $Y_1^C, Y_2^C, \ldots$ has a steady-state distribution F^C and that $Y^C \sim F^C$. Then a measure of performance is said to be a *steady-state cycle parameter* if it is a characteristic of Y^C such as the mean $\nu^C = E(Y^C)$. Thus, a steady-state cycle parameter is just a steady-state parameter of the appropriate cycle process $Y_1^C, Y_2^C, \ldots$.

Example 9.11. Suppose for the manufacturing system in Example 9.9 that there is a half-hour lunch break at the beginning of the fifth hour in each 8-hour shift. Then the process of hourly throughputs $N_1, N_2, \ldots$ has no steady-state distribution (see Prob. 9.6). Let N_i^C be the average hourly throughput in the ith 8-hour shift (cycle). Then we might be interested in estimating the steady-state expected average hourly throughput over a cycle, $\nu^C = E(N^C)$, which is a steady-state cycle parameter.

Example 9.12. Consider a national long-distance telephone company for which one must dial a local phone number to gain access to the system. Suppose that the arrival rate of calls to the system varies with the time of day and day of the week, but assume that the pattern of arrival rates is identical from week to week. Let D_i be the delay experienced by the ith arriving call between dialing the local phone number and actually gaining access to the system, at which time the desired long-distance number can be entered. The stochastic process $D_1, D_2, \ldots$ does not have a steady-state distribution. Let D_i^C be the average delay over the ith week. Then we might be interested in estimating the steady-state expected average delay over a week, $\nu^C = E(D^C)$.

For a nonterminating simulation, suppose that the stochastic process $Y_1, Y_2, \ldots$ does not have a steady-state distribution, and that there is no appropriate cycle definition such that the corresponding process $Y_1^C, Y_2^C, \ldots$ has a steady-state distribution. This can occur, for example, if the parameters

43

for the model continue to change over time. In Example 9.12, if the arrival rate of calls changes from week to week and from year to year, then steady-state (cycle) parameters will probably not be well defined. In these cases, however, there will typically be a fixed amount of data describing how input parameters change over time. This provides, in effect, a terminating event E for the simulation and, thus, the analysis techniques for terminating simulations in Sec. 9.4 are appropriate. This is why we do not treat this situation as a separate case later in the chapter. Measures of performance or parameters for such simulations usually change over time and are included in the category "other parameters" in Fig. 9.4.

Example 9.13. Consider a manufacturing system for microcomputers consisting of an assembly line and a test area. There is a 3-month build schedule available from marketing, which describes the types and numbers of computers to be produced each week. The schedule changes from week to week because of changing sales and the introduction of new computers. In this case, weekly or monthly throughputs do not have steady-state distributions. We therefore perform a terminating simulation of length 3 months and estimate the actual mean throughput for each week.

9.4 STATISTICAL ANALYSIS FOR TERMINATING SIMULATIONS

Suppose that we make n independent replications of a terminating simulation, where each replication is terminated by the event E and is begun with the "same" initial conditions (see Sec. 9.4.3). The independence of replications is accomplished by using different random numbers for each replication. (For a discussion of how this can easily be accomplished if the n replications are made in more than one execution, see Sec. 7.2.) Assume for simplicity that there is a single measure of performance of interest. (This assumption is dropped in Sec. 9.7.) Let X_j be a random variable defined on the jth replication for $j = 1, 2, \ldots, n$; it is assumed that the X_j's are comparable for different replications. Then the X_j's are IID random variables. For the bank of Examples 9.1 and 9.4, X_j might be the average delay $\Sigma_{i=1}^{N} D_i/N$ over a day from the jth replication, where N (a random variable) is the number of customers served in a day. For the combat model of Example 9.5, X_j might be the number of red tanks destroyed on the jth replication. Finally, for the inventory system of Example 9.8, X_j could be the average cost $\Sigma_{i=1}^{120} C_i/120$ from the jth replication.

9.4.1 Estimating Means

Suppose that we would like to obtain a point estimate and confidence interval for the mean $\mu = E(X)$, where X is random variable defined on a replication as described above. Make n independent replications of the simulation and let $X_1, X_2, \ldots, X_n$ be the resulting IID random variables. Then, by substituting the X_j's into (4.3) and (4.12), we get that $\bar{X}(n)$ is an unbiased point estimator

for μ, and an approximate $100(1-\alpha)$ percent $(0 < \alpha < 1)$ confidence interval for μ is given by

$$\bar{X}(n) \pm t_{n-1,1-\alpha/2}\sqrt{\frac{S^2(n)}{n}} \tag{9.1}$$

where the sample variance $S^2(n)$ is given by Eq. (4.4). We will call the confidence interval based on (9.1) the *fixed-sample-size procedure*.

Example 9.14. For the bank of Example 9.1, suppose that we want to obtain a point estimate and an approximate 90 percent confidence interval for the expected average delay of a customer over a day, which is given by

$$E(X) = E\left(\frac{\sum_{i=1}^{N} D_i}{N}\right)$$

(Note that we estimate the expected *average* delay, since each delay has, in general, a different mean.) From the 10 replications given in Table 9.1 we obtained

$$\bar{X}(10) = 2.03, \qquad S^2(10) = 0.31$$

and

$$\bar{X}(10) \pm t_{9,0.95}\sqrt{\frac{S^2(10)}{10}} = 2.03 \pm 0.32$$

Thus, subject to the correct interpretation to be given to confidence intervals (see Sec. 4.5), we can claim with approximately 90 percent confidence that $E(X)$ is contained in the interval [1.71, 2.35] minutes.

Example. 9.15. For the inventory system of Sec. 1.5 and Example 9.8, suppose that we want to obtain a point estimate and an approximate 90 percent confidence interval for the expected average cost over the 120-month planning horizon, which is given by

$$E(X) = E\left(\frac{\sum_{i=1}^{120} C_i}{120}\right)$$

We made 10 independent replications and obtained the following X_j's:

129.35	127.11	124.03	122.13	120.44
118.39	130.17	129.77	125.52	133.75

which resulted in

$$\bar{X}(10) = 126.07, \qquad S^2(10) = 23.55$$

and the 90 percent confidence interval

$$126.07 \pm 2.81 \quad \text{or, alternatively,} \quad [123.26, 128.88]$$

Note that the estimated coefficient of variation (see Table 6.5), a measure of variability, is 0.04 for the inventory system and 0.27 for the bank model. Thus the X_j's for the bank model are inherently more variable than those for the inventory system.

44

Example 9.16. For the bank of Example 9.1, suppose that we would like to obtain a point estimate and an approximate 90 percent confidence interval for the expected proportion of customers with a delay less than 5 minutes over a day, which is given by

$$E(X) = E\left(\frac{\sum_{i=1}^{N} I_i(0,5)}{N}\right)$$

where the *indicator function* $I_i(0,5)$ is defined as

$$I_i(0,5) = \begin{cases} 1 & \text{if } D_i < 5 \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, 2, \ldots, N$. From the last column of Table 9.1, we obtained

$$\bar{X}(10) = 0.853, \quad S^2(10) = 0.004$$

and the 90-percent confidence interval

$$0.853 \pm 0.036 \quad \text{or} \quad [0.817, 0.889]$$

The correctness of the confidence interval given by (9.1) (in terms of having coverage close to $1-\alpha$) depends on the assumption that the X_j's are normal random variables; this is why we called the confidence intervals in Examples 9.14, 9.15, and 9.16 *approximate* 90 percent confidence intervals. Since this assumption will rarely be satisfied in practice, we now use several simple stochastic models with *known* means to investigate empirically the robustness of the confidence interval to departures from normality. Our goal is to provide the simulation practitioner with some guidance as to how well the confidence interval will perform, in terms of coverage, in practice.

We first made 500 independent simulation experiments for the $M/M/1$ queue with $\rho = 0.9$. For each experiment we considered $n = 5$, 10, 20, 40, and for each n we used (9.1) to construct an approximate 90 percent confidence interval for

$$d(25|s=0) = E\left(\frac{\sum_{i=1}^{25} D_i}{25}\middle| s=0\right) = 2.12$$

where s is the number of customers present at time 0 [see Heathcote and Winer (1969) and Example 9.2]. Table 9.2 gives the proportion, $\hat{p}$, of the 500 confidence intervals that covered the true $d(25|s=0)$, a 90 percent confidence interval for the true coverage p [the proportion of a very large number of confidence intervals that would cover $d(25|s=0)$], and the average value of the confidence-interval half-length [that is, $t_{n-1,1-\alpha/2}\sqrt{S^2(n)/n}$] divided by the point estimate $\bar{X}(n)$ over the 500 experiments, which is a measure of the precision of the confidence interval; see below for further discussion. The 90 percent confidence interval for the true coverage is computed from

$$\hat{p} \pm z_{0.95}\sqrt{\frac{\hat{p}(1-\hat{p})}{500}}$$

TABLE 9.2
Fixed-sample-size results for $d(25|s=0) = 2.12$ based on 500 experiments, $M/M/1$ queue with $\rho = 0.9$

n	Estimated coverage	Average of (confidence interval half-length)/$\bar{X}(n)$
5	0.880 ± 0.024	0.67
10	0.864 ± 0.025	0.44
20	0.886 ± 0.023	0.30
40	0.914 ± 0.021	0.21

and is based on the fact that $(\hat{p}-p)/\sqrt{\hat{p}(1-\hat{p})/500}$ is approximately distributed as a standard normal random variable [e.g., Hogg and Craig (1970, p. 187)].

From Table 9.2 it can be seen that 86.4 percent of the 500 confidence intervals based on $n = 10$ replications covered $d(25|s=0)$, and we know with approximately 90 percent confidence that the true coverage for $n = 10$ is between 0.839 and 0.889. Considering that a simulation model is always just an approximation to the corresponding real-world system, we believe that the estimated coverages presented in Table 9.2 are close enough to the desired 0.9 to be useful. Note also from the last column of the table that four times as many replications are required to increase the precision of the confidence interval by a factor of approximately 2. This is not surprising since there is a $\sqrt{n}$ in the denominator of the expression for the confidence-interval half-length in (9.1).

To show that the confidence interval given by (9.1) does not always produce coverages close to $1-\alpha$, we considered a second example. A reliability model consisting of three components will function as long as component 1 works and either component 2 or 3 works. If G is the time to failure of the whole system and G_i is the time to failure of component i (where $i = 1, 2, 3$), then $G = \min\{G_1, \max\{G_2, G_3\}\}$. We further assume that the G_i's are independent random variables and that each G_i has a Weibull distribution with shape parameter 0.5 and scale parameter 1 (see Sec. 6.2.2). This particular Weibull distribution is extremely skewed and nonnormal. Once again we performed 500 independent simulation experiments; for each experiment we considered $n = 5$, 10, 20, 40, and for each n we used (9.1) to construct a 90 percent confidence interval for $E(G|\text{all components new}) = 0.78$ (which was calculated by analytic reasoning). The results from these experiments are given in Table 9.3. Note that for small values of n there is significant coverage degradation. Also, as n gets large, the coverage appears to be approaching 0.9, as guaranteed by the central limit theorem.

We can see from Tables 9.2 and 9.3 that the coverage actually obtained from the confidence interval given by (9.1) depends on the simulation model under consideration (actually, on the distribution of the resulting X_j's) and also on the sample size n. It is therefore natural to ask why the confidence interval

TABLE 9.3
Fixed-sample-size results for $E(G|\text{all components new}) = 0.78$ based on 500 experiments, reliability model

n	Estimated coverage	Average of (confidence interval half-length)/$\bar{X}(n)$
5	0.708 ± 0.033	1.16
10	0.750 ± 0.032	0.82
20	0.800 ± 0.029	0.60
40	0.840 ± 0.027	0.44

worked better for the $M/M/1$ queue than it did for the reliability model. Two possible reasons come to mind. First, an X_j for the queueing system is actually an average of 25 individual delays, while an X_j for the reliability model is computed from the three individual times to failure by a formula involving a minimum and a maximum. There are central limit theorems for certain types of correlated data which state that averages of these data become approximately normally distributed as the number of points in the average gets large. (See Sec. 9.5.3 for further discussion.) We therefore expect that if X_j is the average of a large number of individual points (even though correlated), the degradation in coverage of the confidence interval may not be severe. Our experience indicates that many real-world simulations produce X_j's of this type. A second reason is that the delays for the queueing system are themselves more normal-like than are the times to failure for the reliability model. In fact, recall that the distribution of the times to failure of the individual components was purposely chosen to be extremely nonnormal.

Obtaining a Specified Precision. One disadvantage of the fixed-sample-size procedure based on n replications is that the analyst has no control over the confidence-interval half-length [or the precision of $\bar{X}(n)$]; for fixed n, the half-length will depend on $\mathrm{Var}(X)$, the population variance of the X_j's. In what follows we discuss procedures for determining the number of replications required to estimate the mean $\mu = E(X)$ with a specified error or precision.

We begin by defining two ways of measuring the error in the estimate $\bar{X}$. (The dependence on n is suppressed, since the number of replications may be a random variable.) If the estimate $\bar{X}$ is such that $|\bar{X} - \mu| = \beta$, then we say that $\bar{X}$ has an *absolute error* of β. If we make replications of a simulation until the half-length of the $100(1-\alpha)$ percent confidence interval given by (9.1) is less than or equal to β (where $\beta > 0$), then

$$\begin{aligned} 1-\alpha &\approx P(\bar{X} - \text{half-length} \le \mu \le \bar{X} + \text{half-length}) \\ &= P(|\bar{X} - \mu| \le \text{half-length}) \\ &\le P(|\bar{X} - \mu| \le \beta) \end{aligned}$$

[If A and B are events with A being a subset of B, then $P(A) \le P(B)$.] Thus, $\bar{X}$ has an absolute error of at most β with a probability of approximately $1-\alpha$. In other words, if we construct 100 independent 90 percent confidence intervals using the above stopping rule, we would expect $\bar{X}$ to have an absolute error of at most β in about 90 out of the 100 cases; in about 10 cases the absolute error would be greater than β.

Suppose that we have constructed a confidence interval for μ based on a fixed number of replications n. If we assume that our estimate $S^2(n)$ of the population variance will not change (appreciably) as the number of replications increases, an *approximate* expression for the total number of replications, $n_a^*(\beta)$, required to obtain an absolute error of β is given by

$$n_a^*(\beta) = \min\left\{ i \ge n\colon t_{i-1,1-\alpha/2}\sqrt{\frac{S^2(n)}{i}} \le \beta \right\} \tag{9.2}$$

(The colon ":" is read "such that.") We can determine $n_a^*(\beta)$ by iteratively increasing i by 1 until a value of i is obtained for which $t_{i-1,1-\alpha/2}\sqrt{S^2(n)/i} \le \beta$. [Alternatively, $n_a^*(\beta)$ can be approximated as the smallest integer i satisfying $i \ge S^2(n)(z_{1-\alpha/2}/\beta)^2$.] If $n_a^*(\beta) > n$ and if we make $n_a^*(\beta) - n$ additional replications of the simulation, then the estimate $\bar{X}$ based on all $n_a^*(\beta)$ replications should have an absolute error of approximately β. The accuracy of Eq. (9.2) depends on how close the variance estimate $S^2(n)$ is to $\mathrm{Var}(X)$.

Example 9.17. For the bank of Example 9.14, suppose that we would like to estimate the expected average delay with an absolute error of 0.25 minute and a confidence level of 90 percent. From the 10 available replications, we get

$$n_a^*(0.25) = \min\left\{ i \ge 10\colon t_{i-1,0.95}\sqrt{\frac{0.31}{i}} \le 0.25 \right\} = 16$$

We now discuss another way of measuring the error in $\bar{X}$. If the estimate $\bar{X}$ is such that $|\bar{X} - \mu|/|\mu| = \gamma$, then we say that $\bar{X}$ has a *relative error* of γ, or that the *percentage error* in $\bar{X}$ is 100γ percent. Suppose that we make replications of a simulation until the half-length of the confidence interval given by (9.1), divided by $|\bar{X}|$, is less than or equal to γ $(0 < \gamma < 1)$. This ratio is an estimate of the actual relative error. Then

$$\begin{aligned} 1-\alpha &\approx P(|\bar{X} - \mu|/|\bar{X}| \le \text{half-length}/|\bar{X}|) \\ &\le P(|\bar{X} - \mu| \le \gamma|\bar{X}|) && [(\text{half-length}/|\bar{X}|) \le \gamma] \\ &= P(|\bar{X} - \mu| \le \gamma|\bar{X} - \mu + \mu|) && (\text{add, subtract } \mu) \\ &\le P(|\bar{X} - \mu| \le \gamma(|\bar{X} - \mu| + |\mu|)) && (\text{triangle inequality}) \\ &= P((1-\gamma)|\bar{X} - \mu| \le \gamma|\mu|) && (\text{algebra}) \\ &= P(|\bar{X} - \mu|/|\mu| \le \gamma/(1-\gamma)) && (\text{algebra}) \end{aligned}$$

Thus, $\bar{X}$ has a relative error of at most $\gamma/(1-\gamma)$ with a probability of approximately $1-\alpha$. In other words, if we construct 100 independent 90 percent confidence intervals using the above stopping rule, we would expect $\bar{X}$ to have a relative error of at most $\gamma/(1-\gamma)$ in about 90 of the 100 cases; in

76

about 10 cases the relative error would be greater than $\gamma/(1-\gamma)$. Note that we get a relative error of $\gamma/(1-\gamma)$ rather than the desired γ, since we *estimate* $|\mu|$ by $|\bar{X}|$.

Suppose once again that we have constructed a confidence interval for μ based on a fixed number of replications n. If we assume that our estimates of both the population mean and population variance will not change (appreciably) as the number of replications increases, an *approximate* expression for the number of replications, $n_r^*(\gamma)$, required to obtain a relative error of γ is given by

$$n_r^*(\gamma) = \min\left\{ i \geq n: \frac{t_{i-1,1-\alpha/2}\sqrt{S^2(n)/i}}{|\bar{X}(n)|} \leq \gamma' \right\} \tag{9.3}$$

where $\gamma' = \gamma/(1+\gamma)$ is the "adjusted" relative error needed to get an *actual* relative error of γ. {Again, $n_r^*(\gamma)$ is approximated as the smallest integer i satisfying $i \geq S^2(n)[z_{i-\alpha/2}/\gamma'\bar{X}(n)]^2$.} If $n_r^*(\gamma) > n$ and if we make $n_r^*(\gamma) - n$ additional replications of the simulation, then the estimate $\bar{X}$ based on all $n_r^*(\gamma)$ replications should have a relative error of approximately γ.

Example 9.18. For the bank of Example 9.14, suppose that we would like to estimate the expected average delay with a relative error of 0.10 and a confidence level of 90 percent. From the 10 available replications, we get

$$n_r^*(0.10) = \min\left\{ i \geq 10: \frac{t_{i-1,0.95}\sqrt{0.31/i}}{2.03} \leq 0.09 \right\} = 27$$

where $\gamma' = 0.1/(1+0.1) = 0.09$.

The difficulty with using Eq. (9.3) directly to obtain an estimate $\bar{X}$ with a relative error of γ is that $\bar{X}(n)$ and $S^2(n)$ may not be precise estimates of their corresponding population parameters. If $n_r^*(\gamma)$ is greater than the number of replications actually required, then a significant number of unnecessary replications may be made, resulting in a waste of computer resources. Conversely, if $n_r^*(\gamma)$ is too small, then an estimate $\bar{X}$ based on $n_r^*(\gamma)$ replications may not be as precise as we think. We now present a *sequential* procedure (new replications are added one at a time) for obtaining an estimate of μ with a specified relative error that takes only as many replications as are actually needed. The procedure assumes that $X_1, X_2, \ldots$ is a sequence of IID random variables that need not be normal.

The specific objective of the procedure is to obtain an estimate of μ with a relative error of γ $(0 < \gamma < 1)$ and a confidence level of $100(1-\alpha)$ percent. Choose an initial number of replications $n_0 \geq 2$ and let

$$\delta(n, \alpha) = t_{n-1,1-\alpha/2}\sqrt{\frac{S^2(n)}{n}}$$

be the usual confidence-interval half-length. Then the sequential procedure is as follows:

0. Make n_0 replications of the simulation and set $n = n_0$.
1. Compute $\bar{X}(n)$ and $\delta(n,\alpha)$ from $X_1, X_2, \ldots, X_n$.
2. If $\delta(n,\alpha)/|\bar{X}(n)| \leq \gamma'$, use $\bar{X}(n)$ as the point estimate for μ and stop. Equivalently,

$$I(\alpha,\gamma) = [\bar{X}(n) - \delta(n,\alpha),\ \bar{X}(n) + \delta(n,\alpha)] \tag{9.4}$$

is an approximate $100(1-\alpha)$ percent confidence interval for μ with the desired precision. Otherwise, replace n by $n+1$, make an additional replication of the simulation, and go to step 1.

Note that the procedure computes a new estimate of $\mathrm{Var}(X)$ after *each* replication is obtained, and that the total number of replications required by the procedure is a random variable.

Example 9.19. For the bank of Example 9.14, suppose that we would like to obtain an estimate of the expected average delay with a relative error of $\gamma = 0.1$ and a confidence level of 90 percent. Using the previous $n_0 = 10$ replications as a starting point, we obtained

number of replications at termination = 74
$\bar{X}(74) = 1.76, \quad S^2(74) = 0.67$
90 percent confidence interval: [1.60, 1.92]

Note that the number of replications actually required, 74, is considerably larger than the 27 predicted in Example 9.18, due mostly to the imprecise variance estimate based on 10 replications.

Although the sequential procedure described above is intuitively appealing, the question naturally arises as to how well it performs in terms of producing a confidence interval with coverage close to the desired $1-\alpha$. In Law, Kelton, and Koenig (1981), it is shown that if $\mu \neq 0$ [and $0 < \mathrm{Var}(X) < \infty$], then the coverage of the confidence interval given by Eq. (9.4) will be arbitrarily close to $1-\alpha$, provided the desired relative error is sufficiently close to 0. Based on sampling from a large number of stochastic models and probability distributions (including the $M/M/1$ queue and the above reliability model) for which the true values of μ are known, our recommendation is to use the sequential procedure with $n_0 \geq 10$ and $\gamma \leq 0.15$. It was found that if these recommendations are followed, the estimated coverage (based on 500 independent experiments for each model) for a desired 90 percent confidence interval was never less than 0.864.

Analogous to the sequential procedure described above is a sequential procedure due to Chow and Robbins (1965) for constructing a $100(1-\alpha)$ percent confidence interval for μ with a small absolute error β. Furthermore, it can be shown that the coverage actually produced by the procedure will be arbitrarily close to $1-\alpha$ provided the desired absolute error β is sufficiently close to 0. However, since the meaning of "*absolute error* sufficiently small" is

extremely model-dependent, and since the coverage results in Law (1980) indicate that the procedure is very sensitive to the choice of β, we do not recommend the use of the Chow and Robbins procedure in general.

Recommended Use of the Procedures. We now make our recommendations on the use of the fixed-sample-size and sequential procedures for terminating simulations. If one is performing an exploratory experiment where the precision of the confidence interval may not be overwhelmingly important, we recommend using the fixed-sample-size procedure. However, if the X_j's are highly nonnormal and the number of replications n is too small, the actual coverage of the constructed confidence interval may be somewhat lower than desired.

From an exploratory experiment consisting of n replications, one can estimate the cost per replication and the population variance of the X_j's, and then obtain from Eq. (9.2) a *rough estimate* of the number of replications, $n_a^*(\beta)$, required to estimate μ with a desired absolute error β. Alternatively, one can obtain from Eq. (9.3) a *rough estimate* of the number of replications, $n_r^*(\gamma)$, required to estimate μ with a desired relative error γ. Sometimes the choice of β or γ may have to be tempered by the cost associated with the required number of replications. If it is finally decided to construct a confidence interval with a small relative error γ, we recommend use of the sequential procedure with $\gamma \le 0.15$ and $n_0 \ge 10$. If one wants a confidence interval with a relative error γ greater than 0.15, we recommend several successive applications of the fixed-sample-size approach. In particular, one might estimate $n_r^*(\gamma)$, collect, say $[n_r^*(\gamma) - n]/2$ more replications, and then use (9.1) to construct a confidence interval based on the existing $[n + n_r^*(\gamma)]/2$ replications. If the estimated relative error of the resulting confidence interval is still greater than γ', then $n_r^*(\gamma)$ can be reestimated based on a new variance estimate, and some portion of the necessary additional replications may be collected, etc. To construct a confidence interval with a small absolute error β, we once again recommend several successive applications of the fixed-sample-size approach. It should be mentioned that all of the statistical analyses [except the calculation of $n_a^*(\beta)$] for terminating simulations thus far discussed can be performed in SIMSCRIPT II.5 using an optional library routine called STAT.R [see Law (1979)].

Regardless of the cost per replication, we recommend always making at least three to five replications of a stochastic simulation to assess the variability of the X_j's. If this is not possible due to time or cost considerations, then the simulation study should probably not be done at all.

9.4.2 Estimating Other Measures of Performance

In this section we discuss estimating measures of performance other than means. As the following example shows, comparing two or more systems by some sort of mean system response may result in misleading conclusions.

Example 9.20. Consider the bank of Example 9.14, where the utilization factor $\rho = \lambda/(5\omega) = 0.8$. We compare the policy of having one queue for each teller (and jockeying) with the policy of having one queue feed all tellers on the basis of *expected average delay in queue* (see Example 9.14) and *expected time-average number of customers in queue*, which is defined by

$$E\left[\frac{\int_0^T Q(t)\,dt}{T}\right]$$

where $Q(t)$ is the number of customers in queue at time t and T is the bank's operating time ($T \ge 8$ hours). Table 9.4 gives the results of making one simulation run of each policy. [These simulation runs were performed so that the time of arrival of the ith customer ($i = 1, 2, \ldots, N$) was identical for both policies and so that the service time of the ith customer to begin service ($i = 1, 2, \ldots, N$) was the same for both policies.] Thus, on the basis of "average system response," it would appear that the two policies are equivalent. However, this is clearly not the case. Since customers need not be served in the order of their arrival with the multiqueue policy, we would expect this policy to result in greater variability of a customer's delay. Table 9.5 gives estimates, computed from the same two simulation runs used above, of the expected proportion of customers with a delay in the interval [0,5) (in minutes), the expected proportion of customers with a delay in [5,10), . . . , the expected proportion of customers with a delay in [40,45) for both policies. (We did not estimate variances from these runs since, as pointed out in Sec. 4.4, variance estimates computed from correlated simulation output data are highly biased.) Observe from Table 9.5 that a customer is more likely to have a large delay with the multiqueue policy than with the single-queue policy. In particular, if 480 customers arrive in a day, then 33 and 6 of them would be expected to have delays greater than or equal to 20 minutes for the five-queue and one-queue policies, respectively. (For larger values of ρ, the differences between the two policies would be even greater.) This observation together with the greater equitability of the single-queue policy has probably led many organizations, e.g., banks and airlines, to adopt this policy.

We conclude from the above example that comparing alternative systems of policies on the basis of average system behavior alone can sometimes result in misleading conclusions and, furthermore, that proportions can be a useful measure of system performance. In Example 9.16 we showed how to obtain a point estimate and a confidence interval for an expected proportion. In this

TABLE 9.4
Simulation results for the two bank policies: averages

Measure of performance	Estimates: Five queues	Estimates: One queue
Expected operating time, hours	8.14	8.14
Expected average delay, minutes	5.57	5.57
Expected average number in queue	5.52	5.52

TABLE 9.5
Simulation results for the two bank policies: proportions

Interval (minutes)	Estimates of expected proportions of delays in interval: Five queues	One queue
[0,5)	0.626	0.597
[5,10)	0.182	0.188
[10,15)	0.076	0.107
[15,20)	0.047	0.095
[20,25)	0.031	0.013
[25,30)	0.020	0
[30,35)	0.015	0
[35,40)	0.003	0
[40,45)	0	0

section we show how to perform similar analyses for probabilities and quantiles in the context of terminating simulations.

Let X be a random variable defined on a replication as described in Sec. 9.4.1. Suppose that we would like to estimate the probability $p = P(X \in B)$, where B is a set of real numbers. (For example, B could be the interval $[20,\infty)$ in Example 9.20.) Make n independent replications and let $X_1, X_2, \ldots, X_n$ be the resulting IID random variables. Let S be the number of X_i's that fall in the set B. Then S has a binomial distribution (see Sec. 6.2.3) with parameters n and p, and an unbiased point estimator for p is given by

$$\hat{p} = \frac{S}{n}$$

Furthermore, a confidence interval for p may be constructed using procedures described in Welch (1983, pp. 285–287) and Conover (1980, pp. 99–104) (see also Prob. 9.9).

Example 9.21. For the bank of Example 9.14, suppose that we would like to get a point estimate for

$$p = P(X \le 15) \qquad \text{where } X = \max_{0 \le t \le T} Q(t)$$

In this case $B = [0,15]$. We made 100 independent replications of the bank simulation and obtained $\hat{p} = 0.77$. Thus, for approximately 77 out of every 100 days, we expect the maximum queue length during a day to be less than or equal to 15 customers.

Suppose now that we would like to estimate the q-quantile ($100q$th percentile) x_q of the distribution of the random variable X (see Sec. 6.4.3 for the definition). For example, the 0.5-quantile is the median. If $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$ are the order statistics corresponding to the X_i's from n

independent replications, then a point estimator for x_q is the sample q-quantile $\hat{x}_q$, which is given by

$$\hat{x}_q = \begin{cases} X_{(nq)} & \text{if } nq \text{ is an integer} \\ X_{(\lfloor nq+1 \rfloor)} & \text{otherwise} \end{cases}$$

A confidence interval for x_q can also be obtained; see Welch (1983, pp. 287–288) and Conover (1980, pp. 111–116).

Example 9.22. For the bank of Example 9.14, suppose that we would like to decide how large a lobby is needed to accommodate customers waiting in the queue. If we let X be the maximum queue length as defined in Example 9.21, then we might want to build a lobby large enough to hold $x_{0.95}$ customers, the 0.95-quantile of X. From the 100 replications in the previous example, we obtained $\hat{x}_{0.95} = X_{(95)} = 20$. Thus, if the lobby has room for 20 customers, this will be sufficient for approximately 95 out of every 100 days. Note also that $\hat{x}_{0.99} = X_{(99)} = 23$.

The interested reader may also want to consult Conover (1980, pp. 117–121) for a discussion of *tolerance limits*, which is an interval that contains a specified proportion of the *values* of the random variable X (and does so with a certain prescribed confidence level).

9.4.3 Choosing Initial Conditions

As stated in Sec. 9.3, the measures of performance for a terminating simulation depend explicitly on the state of the system at time 0; thus, care must be taken in choosing appropriate initial conditions. Let us illustrate this potential problem by means of an example. Suppose that we would like to estimate the expected average delay of all customers who arrive and complete their delays between 12 noon and 1 P.M. (the busiest period) in a bank. Since the bank will probably be quite congested at noon, starting the simulation then with no customers present (the usual initial conditions for a queueing simulation) will cause our estimate of expected average delay to be biased low. We now discuss two heuristic approaches to this problem, the first of which appears to be used widely (see Sec. 9.5.1).

For the first approach, let us assume that the bank opens at 9 A.M. with no customers present. Then we can start the simulation at 9 A.M. with no customers present and run it for 4 simulated hours. In estimating the desired expected average delay, we use only the delays of those customers who arrive and complete their delays between noon and 1 P.M. The evolution of the simulation between 9 A.M. and noon (the "warmup period") determines the appropriate conditions for the simulation at noon. A disadvantage of this approach is that 3 hours of simulated time are not used directly in the estimate. As a result, one might compromise and start the simulation at some other time, say 11 A.M., with no customers present. However, there is no guarantee that the conditions in the simulation at noon will be representative of the actual

79

conditions in the bank at noon. This approach can be carried out in SIMLIB (see Chap. 2) by reinitializing the statistical counters for subroutines SAMPST, TIMEST, and FILEST (see Prob. 2.7) at noon.

An alternative approach is to collect data on the number of customers present in the bank at noon for several different days. Let $\hat{p}_i$ be the proportion of these days that i customers ($i = 0, 1, \ldots$) are present at noon. Then we simulate the bank from noon to 1 P.M. with the number of customers present at noon being randomly chosen from the distribution $\{\hat{p}_i\}$. (All customers who are being served at noon might be assumed to be just beginning their services. Starting all services fresh at noon results in an approximation to the actual situation in the bank, since the customers who are in the process of being served at noon would have partially completed their services. However, the effect of this approximation should be negligible for a simulation of length 1 hour.)

If more than one simulation run from noon to 1 P.M. is desired, then a different sample from $\{\hat{p}_i\}$ is drawn for each run. The X_j's that result from these runs are still IID, since the initial conditions for each run are chosen independently from the same distribution.

80

CHAPTER 10

COMPARING ALTERNATIVE SYSTEM CONFIGURATIONS

Recommended sections for a first reading: 10.1 through 10.3, 10.4.1

10.1 INTRODUCTION

In Chap. 9 we saw the importance of applying appropriate statistical analyses to the output from a simulation model of a *single* system. In this chapter we discuss statistical analyses of the output from several *different* simulation models that might represent competing system designs or alternative operating policies. This is a very important subject, since the real utility of simulation lies in comparing such alternatives before implementation. As the following example illustrates, appropriate statistical methods are essential if we are to avoid making serious errors leading to fallacious conclusions and, ultimately, poor decisions. We hope that this example will demonstrate the danger inherent in making decisions based on the output from a *single* run (or replication) of each alternative system.

Example 10.1. A bank planning to install an automated teller station must choose between buying one Zippytel machine or two Klunkytel machines. Although one Zippy costs twice as much to purchase, install, and operate as one Klunky, the Zippy works twice as fast. Since the total cost to the bank is thus the same regardless of its decision, the managers would like to install the system that will provide the best service.

From available data, it appears that during a certain rush period, customers arrive one at a time according to a Poisson process with rate 1 per minute. The Zippy could provide service times that are IID exponential random variables with mean 0.9 minute. Alternatively, if two Klunkies are installed, each will yield service times that are IID exponential random variables with mean 1.8 minutes; in this case a single FIFO queue will be formed instead of two separate lines. Thus, we are comparing an $M/M/1$ queue with an $M/M/2$ queue, each with utilization factor $\rho = 0.9$, as shown in Fig. 10.1. The performance measure of interest is the expected average delay in queue of the first 100 customers, assuming that the first customer arrives to an empty and idle system; we denote these (expected) quantities by $d_Z(100)$ and $d_K(100)$ for the one-Zippy and two-Klunky cases, respectively. (The bank decided to ignore customer service times, since waiting in line is the most irritating part of the experience and customers are reasonably pacified as long as they are being served; see Prob. 10.1 for further consideration of this issue.) The bank's intrepid systems analyst decided to make a simulation run of length 100 customer delays for each system (using independent random numbers) and to use the average of the 100 delays in each case to infer whether $d_Z(100)$ or $d_K(100)$ is smaller, and thus make a recommendation.

How likely is it that the analyst will make the right recommendation? To find out, we performed 100 independent experiments of the analyst's entire scheme and noted how many times the best system would have been recommended. The best system is actually the two-Klunky installation, since $d_Z(100) = 4.13$ and $d_K(100) = 3.70$. [These values were determined from the queueing-theoretic results in Kelton and Law (1985).] Our experiment was, thus, to

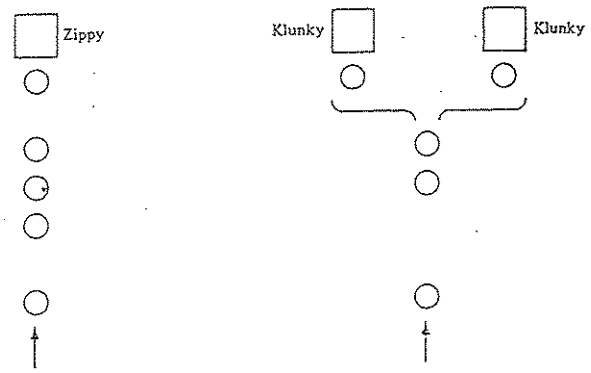


FIGURE 10.1
One Zippy or two Klunkies?

84

perform 100 independent pairs of independent simulations of the two systems, and average the delays in each simulation to obtain $\hat{d}_Z(100)$ and $\hat{d}_K(100)$, say, and then recommend the Zippy or Klunky system according as $\hat{d}_Z(100)$ or $\hat{d}_K(100)$ was smaller; some of the results are in Table 10.1. In only 48 of our 100 experiments was $\hat{d}_K(100) < \hat{d}_Z(100)$, so the analyst would not really appear to have any better chance of making the right decision than making the wrong one.

We have an uneasy feeling that many simulation studies are carried out in a manner similar to that described in Example 10.1. The difficulty is that the simulation output data are stochastic, so comparing the two systems on the basis of only a single run of each is a very unreliable approach.

The following example indicates how the comparison in Example 10.1 could be improved.

Example 10.2. To illuminate the problem with the one-run-of-each approach in Example 10.1, we plotted all 100 $\hat{d}_Z(100)$'s and $\hat{d}_K(100)$'s in the "$n = 1$" pair of horizontal dot plots in Fig. 10.2; each circle (solid or hollow) represents the average of the 100 delays in a single simulation, positioned according to the scale at the bottom. Even though the *expected* average delay for the two-Klunky system is smaller than that for the one-Zippy system, the distributions of the *observed* average delays overlap substantially. This accounts for the distressingly large probability of making the wrong choice noted at the end of Example 10.1.

Instead, we could make some number, n, of complete independent replications of each alternative system, and compare the systems on the basis of their averages across replications. Specifically, let X_{1j} be the average of the 100 delays in the one-Zippy system on the jth independent replication of this system, and let X_{2j} be the average of the 100 delays in the two-Klunky system on its jth replication, for $j = 1, 2, \ldots, n$. (We also made the simulations so that the X_{1j}'s and the X_{2j}'s are independent.) Then if $\bar{X}_1(n)$ and $\bar{X}_2(n)$ are the sample means of the X_{1j}'s and X_{2j}'s, respectively, we would recommend the system with the smaller $\bar{X}_i(n)$. (The method of Example 10.1 is thus a special case, taking $n = 1$.)

TABLE 10.1
Testing the analyst's decision rule

Experiment	$\hat{d}_Z(100)$	$\hat{d}_K(100)$	Recommendation	
1	3.80	4.60	Zippy	(wrong)
2	3.17	8.37	Zippy	(wrong)
3	3.96	4.18	Zippy	(wrong)
4	1.91	5.77	Zippy	(wrong)
5	1.71	2.23	Zippy	(wrong)
6	6.16	4.72	Klunky	(right)
7	5.67	1.39	Klunky	(right)
⋮	⋮	⋮	⋮	
98	8.40	9.39	Zippy	(wrong)
99	7.70	1.54	Klunky	(right)
100	4.64	1.17	Klunky	(right)

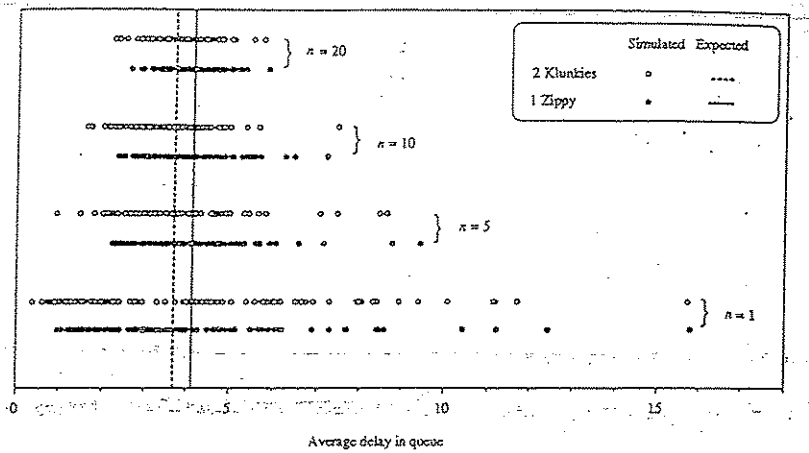


FIGURE 10.2
One Zippy vs. two Klunkies, as described in Examples 10.1 and 10.2.

Table 10.2 shows the proportion of 100 independent pairs of n-replication averages in which the one-Zippy system appeared better, i.e., would result in the wrong recommendation, for $n = 1, 5, 10$, and 20. The chance of making an error falls as n increases, but at a corresponding higher cost of simulating. The four pairs of plots in Fig. 10.2 also indicate that as n rises, the distributions of the n-replication averages (each circle represents such an average) tighten up around their expectations, but there is still considerable overlap even for $n = 20$, where the proportion of incorrect recommendations is still 0.34.

Examples 10.1 and 10.2 illustrate the need for careful design and analysis of comparative simulations. Indeed, even with $n = 20$ replications of each system design, Example 10.2 indicates that there is substantial room for error. One way of sharpening the comparison will be discussed in Sec. 11.2, and the above examples will be reworked in that context; see Example 11.2 in Sec. 11.2.4.

TABLE 10.2
Proportion of wrong recommendations in the n-replication method of Example 10.2

n	Proportion of experiments favoring the one-Zippy system
1	0.52
5	0.43
10	0.38
20	0.34

85

Note that both Examples 10.1 and 10.2 dealt with terminating simulations (see Secs. 9.3 and 9.4). As we shall see in this chapter, a basic requirement for using many statistical methods for comparing alternative configurations is the ability to collect IID observations with expectation equal to the desired measure of performance. For terminating simulations, this is easily accomplished by simply making independent replications; e.g., a basic unit of observation in Examples 10.1 and 10.2 was the average of the 100 delays in a single *entire* replication of the model. If we want to compare alternative systems on the basis of steady-state behavior (see Secs. 9.3 and 9.5), however, the situation becomes more complicated since we cannot easily obtain IID observations having expectation (approximately) equal to the desired steady-state measure of performance. There are different ways of dealing with steady-state comparisons, which will be discussed throughout the chapter, specifically in Secs. 10.2.4 and 10.4.4.

Our purpose in this chapter is to present several different types of comparison and selection problems that have been found useful in simulation, together with appropriate statistical procedures for their solution, and numerical examples. We assume for this chapter that the various alternative systems are simply *given*. In many situations care should be taken in choosing *which* particular system variants to simulate; see Chap. 12 for discussion of how to choose appropriate alternative systems for comparison.

In Sec. 10.2 we treat the special but important case of comparing just two systems by constructing a confidence interval for the difference between their performance measures. These ideas are extended in Sec. 10.3 to confidence-interval comparisons of more than two systems. Section 10.4 introduces some procedures for selecting the "best" of several alternative systems, as well as other goals involving choice of certain "good" subsets from among the set of all alternatives. Appendixes 10A and 10B treat certain technical issues related to the selection procedures of Sec. 10.4.

10.2 CONFIDENCE INTERVALS FOR THE DIFFERENCE BETWEEN PERFORMANCE MEASURES OF TWO SYSTEMS

Here we consider the special case of comparing two systems on the basis of some performance measure, or expected *response*. We effect this comparison by forming a confidence interval for the *difference* in the two expectations, rather than by doing a hypothesis test to see whether the observed difference is significantly different from zero. Whereas a test results in only a "reject" or "fail-to-reject" conclusion, a confidence interval gives us this information (according as the interval misses or contains zero, respectively) as well as quantifies how much the measures differ, if at all. Also, we shall take a parametric, i.e., normal-theory, approach here, even though nonparametric analogues could be used instead [see, for example, Conover (1980, pp. 223–225)]. The parametric approach is simple and familiar, and moreover

should be quite robust in this context, since troublesome skewness (see Sec. 9.4.1) in the underlying distributions of the output random variables should be ameliorated upon subtraction (assuming the two output distributions are skewed in the same direction).

For $i = 1, 2$, let $X_{i1}, X_{i2}, \ldots, X_{in_i}$ be a sample of n_i IID observations from system i, and let $\mu_i = E(X_{ij})$ be the expected response of interest; we want to construct a confidence interval for $\zeta = \mu_1 - \mu_2$. Whether or not X_{1j} and X_{2j} are independent depends on how the simulations are executed, and could determine which of the two confidence-interval approaches discussed in Secs. 10.2.1 and 10.2.2 are used.

10.2.1 A Paired-t Confidence Interval

If $n_1 = n_2$ ($=n$, say), or we are willing to discard some observations from the system on which we actually have more data, we can pair X_{1j} with X_{2j} to define $Z_j = X_{1j} - X_{2j}$, for $j = 1, 2, \ldots, n$. Then the Z_j's are IID random variables and $E(Z_j) = \zeta$, the quantity for which we want to construct a confidence interval. Thus, we can let

$$\bar{Z}(n) = \frac{\sum_{j=1}^{n} Z_j}{n}$$

and

$$\widehat{\mathrm{Var}}[\bar{Z}(n)] = \frac{\sum_{j=1}^{n} [Z_j - \bar{Z}(n)]^2}{n(n-1)}$$

and form the (approximate) $100(1-\alpha)$ percent confidence interval

$$\bar{Z}(n) \pm t_{n-1,1-\alpha/2}\sqrt{\widehat{\mathrm{Var}}[\bar{Z}(n)]} \tag{10.1}$$

If the Z_j's are normally distributed, this confidence interval is exact, i.e., it covers ζ with probability $1-\alpha$; otherwise, we rely on the central limit theorem (see Sec. 4.5), which implies that this coverage probability will be *near* $1-\alpha$ for large n. An important point here is that we did *not* have to assume that X_{1j} and X_{2j} are independent; nor did we have to assume that $\mathrm{Var}(X_{1j}) = \mathrm{Var}(X_{2j})$. Allowing positive correlation between X_{1j} and X_{2j} can be of great importance, since this leads to a reduction in $\mathrm{Var}(Z_j)$ (see Prob. 4.13) and thus to a smaller confidence interval. Section 11.2 discusses a method (*common random numbers*) that can often induce this positive correlation between the observations on the different systems. The confidence interval in (10.1) will be called the *paired-t confidence interval*, and in its derivation we essentially reduced the two-system problem to one involving a single sample, namely, the Z_j's. In this sense, the paired-t approach is the same as the method discussed in Sec. 9.4.1 for analysis of a single system. (Thus, the sequential confidence-interval

86

procedures of Sec. 9.4.1 could be applied here.) It is important to note that the X_{ij}'s are random variables defined over an entire *replication*; for example, X_{1j} might be the average of the 100 delays on the jth replication of the Zippytel system of Example 10.2; it is *not* the delay of some individual customer.

Example 10.3. For the inventory model of Sec. 1.5, suppose we want to compare two different (s,S) policies in terms of their effect on the expected average total cost per month for the first 120 months of operation, where we assume that the initial inventory level is 60. For the first policy $(s,S) = (20,40)$, and the second policy sets $(s,S) = (20,80)$. Here, X_{ij} is the average total cost per month of policy i on the jth independent replication. We made the runs for policy 1 and policy 2 independently of each other and made $n = n_1 = n_2 = 5$ independent replications of the model under each policy; Table 10.3 contains the results. Using the paired-t approach, we obtained $\bar{Z}(5) = 4.98$ and $\widehat{\mathrm{Var}}[\bar{Z}(5)] = 2.44$, leading to the (approximate) 90 percent confidence interval [1.65, 8.31] for $\zeta = \mu_1 - \mu_2$. Thus, with approximately 90 percent confidence, we can say that μ_1 differs from μ_2; and it furthermore appears that policy 2 is superior, since it leads to a lower average operating cost (between 1.65 and 8.31 lower, which would *not* have been evident from a hypothesis test). We must use the word "approximate" to describe the confidence level, since $n_1 = n_2 = 5$ may or may not be "large" enough for this model for the central limit theorem to have taken effect.

10.2.2 A Modified Two-Sample-t Confidence Interval

A second approach to forming a confidence interval for ζ does not pair up the observations from the two systems, but *does* require that the X_{1j}'s be independent of the X_{2j}'s. However, n_1 and n_2 can now be different.

To apply the classical two-sample-t approach [see, for example, Devore (1982, pp. 287–291)], we *must* have $\mathrm{Var}(X_{1j}) = \mathrm{Var}(X_{2j})$; if these variances are not equal, the two-sample-t confidence interval can exhibit serious coverage degradation. [If, however, $n_1 = n_2$, the two-sample-t approach is fairly safe even if the variances differ; see Scheffé (1970) for further discussion.] Since equality of variances is probably not a safe assumption when simulating real systems, we would recommend against using the two-sample-t approach.

TABLE 10.3
Average total cost per month for five independent replications of two inventory policies, and the differences

j	X_{1j}	X_{2j}	Z_j
1	126.97	118.21	8.76
2	124.31	120.22	4.09
3	126.68	122.45	4.23
4	122.66	122.68	−0.02
5	127.23	119.40	7.83

Instead, we shall give an old but reliable approximate solution, due to Welch (1938), to this problem of comparing two systems with unequal and unknown variances, called the *Behrens-Fisher problem* when the X_{ij}'s are normally distributed [see also Scheffé (1970)]. As usual, let

$$\bar{X}_i(n_i) = \frac{\sum_{j=1}^{n_i} X_{ij}}{n_i}$$

and

$$S_i^2(n_i) = \frac{\sum_{j=1}^{n_i} [X_{ij} - \bar{X}_i(n_i)]^2}{n_i - 1}$$

for $i = 1, 2$. Then compute the *estimated* degrees of freedom

$$\hat{f} = \frac{[S_1^2(n_1)/n_1 + S_2^2(n_2)/n_2]^2}{[S_1^2(n_1)/n_1]^2/(n_1 - 1) + [S_2^2(n_2)/n_2]^2/(n_2 - 1)}$$

and use

$$\bar{X}_1(n_1) - \bar{X}_2(n_2) \pm t_{\hat{f},1-\alpha/2}\sqrt{\frac{S_1^2(n_1)}{n_1} + \frac{S_2^2(n_2)}{n_2}} \tag{10.2}$$

as an approximate $100(1-\alpha)$ percent confidence interval for ζ. Since $\hat{f}$ will not, in general, be an integer, interpolation in the t tables will probably be necessary. The confidence interval given by (10.2), which we will call the *Welch confidence interval*, can also be used to validate a simulation model of an existing system (see Sec. 5.6.2). If "system 1" is the real-world system on which we have physically collected data and "system 2" is the corresponding simulation model from which we have simulation output data, it is likely that n_1 will be far less than n_2. Finally, if we are comparing two simulated systems and want a "small" confidence interval, a sequential procedure due to Robbins, Simons, and Starr (1967) can be used, which is efficient in the sense of minimizing the final value of $n_1 + n_2$. It is also asymptotically correct in the sense that the confidence interval will have approximately the correct coverage probability as the prespecified confidence-interval width becomes small.

Example 10.4. Since the runs for the two different inventory policies of Example 10.3 were done independently, we can apply the Welch approach to form an approximate 90 percent confidence interval for ζ; we use the same X_{ij} data as given in Table 10.3. We get $\bar{X}_1(5) = 125.57$, $\bar{X}_2(5) = 120.59$, $S_1^2(5) = 4.00$, $S_2^2(5) = 3.76$, and $\hat{f} = 7.99$. Interpolating in the t tables leads to $t_{7.99,0.95} = 1.860$. Thus, the Welch confidence interval is [2.66, 7.30].

10.2.3 Contrasting the Two Methods

Since the inventory data of Table 10.3 were collected so that $n_1 = n_2$ and the X_{1j}'s were independent of the X_{2j}'s, we could apply either the paired-t or Welch approach to construct a confidence interval for ζ. It happened that the

87