Discrete-Event Simulation:

A First Course

Section 8.3: Finite-Horizon and Infinite-Horizon Statistics

Section 8.3: Finite-Horizon and Infinite-Horizon Statistics

Definition 8.3.1: Steady-state statistics

- Produced by simulating the operation of a *stationary* discrete-event system for an infinite length of time
- Steady-state statistics may not exist

  0-13-142917-5

Example 8.3.1

- Use `sis4` to simulate different numbers of time intervals, compute average inventory level $\bar{l}$ for each
- Output from `sis4` for an increasing number of time intervals n

n:	20	40	80	160	320	640	1280	2560
$\bar{l}$:	25.98	26.09	25.49	27.24	26.96	26.36	27.19	26.75

- As $n \to \infty$, $\bar{l}$ seems to converge to 26.75
- Can only approximate $n \to \infty$ convergence
- A better approach is to use *interval* estimation for the steady-state $\bar{l}$, using large but finite n
- One way is to use *batch means* — Section 8.4

Example 8.3.2

- Use ssq4 to simulate different numbers of jobs n, compute $\bar{w}$, $\bar{l}$
- Use an arrival rate of 1/2
- Change *Erlang* parameters so service rate is 1/2 or less
- As n increases, $\bar{w}$ and $\bar{l}$ will tend to increase, without limit
- Why? The server cannot keep up with demand, so the queue will grow as more jobs arrive
- In this case, the steady-state average wait and the average number in the node are both infinite

Finite-Horizon Versus Infinite-Horizon Statistics

- Steady-state statistics are known as *infinite-horizon* statistics

  An infinite-horizon discrete-event simulation is one for which the simulated operational time is effectively infinite
- A *finite-horizon* discrete-event simulation is one for which the simulated operational time is finite
- *Transient* system statistics are those statistics that are produced by a finite-horizon discrete-event simulation
- The initial conditions affect finite-horizon statistics
- The initial conditions *do not* affect infinite-horizon statistics: after enough time, the system loses memory of its initial state

Example 8.3.3

- With minor modifications, program `ssq4` can be used to simulate an initially idle $M/M/1$ service node processing a small number of jobs (say 100) and with relatively high traffic intensity (say 0.8)
- If the program is executed multiple times varying *only* the `rngs` initial seed from replication to replication,
  - the average wait in the node will vary significantly
  - for most replications, the average wait will *not* be close to the steady-state average wait
- If a relatively large number of jobs (say 10000) are used, variability of the average wait will be much less significant and be close to the steady-state average number
- When the number of jobs becomes infinite, the initial condition bias disappears

Another Important Distinction

- In an infinite-horizon simulation, the system “environment” is assumed to remain *static*

  If the system is a single-server service node, both the arrival rate and the service rate are assumed to remain constant in time
- In a finite-horizon simulation, no need to assume a static environment

Relative Importance of Two Statistics

- The “traditional” view: steady-state statistics are most important
  - Steady-state statistics are better understood because they are much more easy to analyze *mathematically*
  - It is frequently difficult to accurately model initial conditions and non-stationary system parameters
- The “pragmatic” view: transient statistics are most important because steady-state is just a convenient fiction
- Depending on the application, both transient and steady-state statistics may be important
- Important to decide which statistics best characterize the system’s performance

  0-13-142917-5

Initial and Terminal Conditions

- Finite-horizon discrete-event simulations are also known as *terminating* simulations
  - In program `ssq4`, the system state is idle at the beginning and at the end of the simulation
  - The terminal condition is specified by the "close the door" time
  - The system state of `sis4` is the current and on-order inventory levels; these states are the same at the beginning and at the end of the simulation
  - The terminal condition is specified by the number of time intervals
- Infinite-horizon discrete-event simulations must be terminated; typically done using whatever stopping conditions are most convenient

Formal Representation

- The state variable $X(\cdot)$ is known formally as a *stochastic process*
  - The typical objective of a finite-horizon simulation of this system would be to estimate the time-averaged *transient* statistic
    $$\bar{X}(\tau) = \frac{1}{\tau}\int_0^\tau X(t)dt$$
    where $\tau > 0$ is the terminal time
  - The typical objective of an infinite-horizon simulation of this sytem would be to estimate the time-averaged *steady-state* statistics
    $$\bar{x} = \lim_{\tau\to\infty} \bar{X}(\tau) = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau X(t)dt$$
- Note that $\bar{X}(\tau)$ is a random variable; $\bar{x}$ is not

Definition 8.3.3

- If a discrete-event simulation is repeated, varying *only* the `rngs` initial states from run to run, each run of the simulation program is a *replication* and the totality of replications is said to be *ensemble*
- Replications are used to generate *estimates* of the same transient statistic
- The initial seed for each replication should be chosen to be no replication-to-replication overlap
- The standard way is to use the final state of each `rngs` stream from *one* replication as the initial state for the *next* replication accomplished by calling `PlantSeeds` once outside the main replication loop

Independent Replications and Interval Estimation

- Suppose the finite-horizon simulation is replicated n times, each time generating a state time history $x_i(t)$

$$\bar{x}_i(\tau) = \frac{1}{\tau}\int_0^\tau x_i(t)dt$$

where $i = 1, 2, \ldots, n$ is the replication index

- Each data point $\bar{x}_i(\tau)$ is an independent observation of the random variable $\bar{X}(\tau)$
- If n is large enough, the pdf of $\bar{X}(\tau)$ can be estimated from a histogram of the $\bar{x}_i(\tau)$

Independent Replications and Interval Estimation (2)

- Want $E[\bar{X}(\tau)]$ — *point* estimate is available as an *ensemble average*

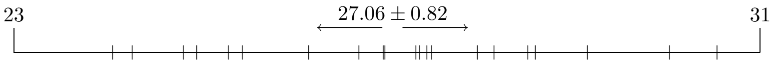
$$\frac{1}{n}\sum_{i=1}^{n}\bar{x}_i(\tau)$$

- An *interval* estimate for $E[\bar{X}(\tau)]$ can be calculated
  - Use the interval estimation technique from Section 8.1
  - Requires ensemble mean and standard deviation of $\bar{x}_i(\tau)$

Example 8.3.5

A modified version of `sis4` was used to produce 21 replications

- 100 time intervals of operation were simulated
- Inventory parameters were $(s, S) = (20, 80)$.
- Measured $\bar{L}(\tau)$, the time-averaged inventory level
- Each of these 21 numbers is a *realization* of $\bar{L}(\tau)$



Example 8.3.5, continued

- The mean and standard deviation are 27.06 and 1.76 respectively
- If a 95% confidence interval estimate is desired,

$$t^* = \texttt{idfStudent}(20, 0.975) = 2.086$$

- The 95% confidence interval estimate is

$$27.06 \pm \frac{(2.086)(1.76)}{\sqrt{20}} = 27.06 \pm 0.82$$

- We are 95% confident that $E[\bar{L}(\tau)]$ is within interval 27.06 ± 0.82

Example 8.3.6

A modified version of ssq2 was used to produce 20 replications

- 100 jobs processed through $M/M/1$ service node
  - Node is initially idle
  - Arrival rate is $\lambda = 1.0$
  - Service rate is $\nu = 1.25$
- The resulting 20 observations of the average wait in the node:

- From Algorithm 8.1.1, the resulting 95% confidence interval estimate is 3.33 ± 1.04

Example 8.3.6, continued

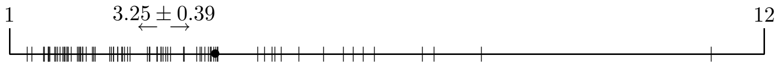
- Steady-state average wait for $M/M/1$ is

$$\frac{1}{\nu - \lambda}$$

  (See Section 8.5)
- In our example
  - Arrival rate is $\lambda = 1.00$
  - Service rate is $\nu = 1.25$
  - Average wait is $\frac{1}{1.25-1.00} = 4.0$
- It is *possible* that 100 jobs are enough to produce steady-state statistics

Example 8.3.7

- The modified version of program `ssq2` was used to produce 60 more replications
- Consistent with $\sqrt{n}$ rule, expect two-fold decrease in the width of the interval estimate
- Based on 80 replications, the resulting 95% confidence interval estimate was 3.25 ± 0.39



- In this case 100 jobs are not enough to produce steady-state statistics

Example 8.3.8

- As a continuation of Example 8.3.6, the number of jobs per replication was increased from 100 to 1000
- 20 replications were used to produce 20 observations of the average wait in the node

- Relative to Example 8.3.6, much more symmetric sample mean in Example 8.3.8

Example 8.3.8, continued

- The 1000-jobs per replication results are more consistent with the underlying theory of interval estimation
  - Requires a sample mean distribution that is approximately *Normal*$(\mu, \sigma/\sqrt{n})$
  - Sample mean distribution is centered on (unknown) population mean
- 1000 jobs may achieve steady-state; 100 jobs cannot
- More on this issue in Section 8.4

  0-13-142917-5