Discrete-Event Simulation:
A First Course

Section 3.1 Discrete-Event Simulation

Section 3.1 Discrete-Event Simulation

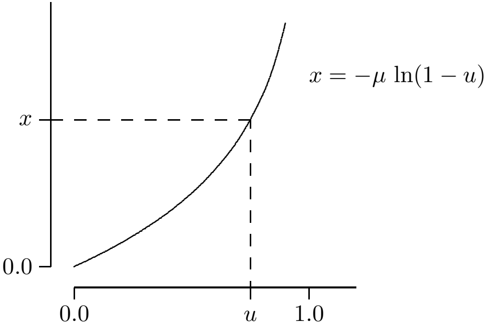
- `ssq1` and `sis1` are trace-driven discrete-event simulations
- Both rely on input data from an external source
- These realizations of naturally occurring stochastic processes are limited
- We cannot perform "what if" studies without modifying the data
- We will convert the single server service node and the simple inventory system to utilize randomly generated input

Single Server Service Node

- We need stochastic assumptions for service times and arrival times
- Assume service times are between 1.0 and 2.0 minutes
  - The distribution within this range is unknown
  - Without further knowledge, we assume no time is more likely than any other
- We will use a *Uniform*(1.0, 2.0) random variate

Exponential Random Variates

- In general, it is unreasonable to assume that all possible values are equally likely.
- Frequently, small values are more likely than large values
- We need a non-linear transformation that maps 0.0 $\rightarrow$ 1.0 to 0.0 $\rightarrow \infty$.



Exponential Random Variates

- The transformation is monotone increasing, one-to-one, and onto

$$
\begin{aligned}
0 < u < 1 \quad &\Longleftrightarrow \quad 0 < (1-u) < 1 \\
&\Longleftrightarrow \quad -\infty < \ln(1-u) < 0 \\
&\Longleftrightarrow \quad 0 < -\mu \ln(1-u) < \infty \\
&\Longleftrightarrow \quad 0 < x < \infty
\end{aligned}
$$

Generating an Exponential Random Variate

```
double Exponential(double μ)      /* use μ > 0.0 */
{
    return (-μ * log(1.0 - Random()));
}
```

- The parameter μ specifies the sample mean
- In the single-server service node simulation, we use *Exponential*(μ) interarrival times

$$a_i = a_{i-1} + \texttt{Exponential}(\mu); \qquad i = 1, 2, 3, \ldots, n$$

Program ssq2

- Program `ssq2` is an extension of `ssq1`
  - Interarrival times are drawn from *Exponential*(2.0)
  - Service times are drawn from *Uniform*(1.0, 2.0)
- The program generates all first-order statistics $\bar{r}$, $\bar{w}$, $\bar{d}$, $\bar{s}$, $\bar{l}$, $\bar{q}$, and $\bar{x}$
- It can be used to study the *steady-state* behavior
  - Will the statistics converge independent of the initial seed?
  - How many jobs does it take to achieve steady-state behavior?
- It can be used to study the *transient* behavior
  - Fix the number of jobs processed and replicate the program with the initial state fixed
  - Each replication uses a different initial `rng` seed

Example 3.1.3

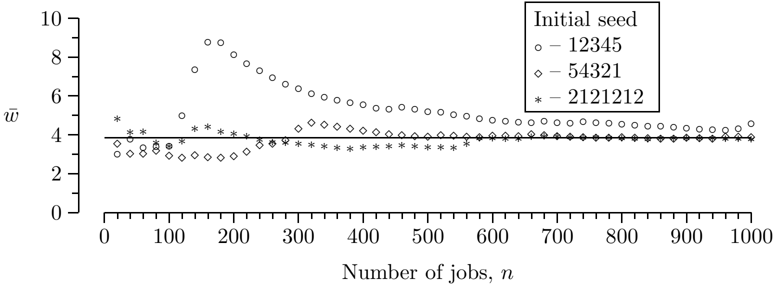
- The theoretical averages for a single-server service node using *Exponential*(2.0) arrivals and *Uniform*(1.0, 2.0) service times are

$\bar{r}$	$\bar{w}$	$\bar{d}$	$\bar{s}$	$\bar{l}$	$\bar{q}$	$\bar{x}$
2.00	3.83	2.33	1.50	1.92	1.17	0.75

- Although the server is busy 75% of the time, on average there are approximately two jobs in the service node
- A job can expect to spend more time in the queue than in service
- To achieve these averages, many jobs must pass through node

Example 3.1.3

- The accumulated average wait was printed every 20 jobs



- The convergence of $\bar{w}$ is slow, erratic, and dependent on the initial seed

Geometric Random Variates

- The *Geometric*(p) random variate is the discrete analog to a continuous *Exponential*(μ) random variate
  Let $x = \textit{Exponential}(\mu) = -\mu \ln(1-u)$
  Let $y = \lfloor x \rfloor$ and let $p = \Pr(y \neq 0)$.

$$
\begin{aligned}
y = \lfloor x \rfloor \neq 0 \quad &\iff \quad x \geq 1 \\
&\iff \quad -\mu \ln(1-u) \geq 1 \\
&\iff \quad \ln(1-u) \leq -1/\mu \\
&\iff \quad 1 - u \leq \exp(-1/\mu)
\end{aligned}
$$

Since $1 - u$ is also *Uniform*$(0.0,1.0)$,
$p = \Pr(y \neq 0) = \exp(-1/\mu)$
Finally, since $\mu = -1/\ln(p)$,

$$y = \lfloor \ln(1-u)/\ln(p) \rfloor$$

Geometric Random Variates

- ANSI C function

Generating a Geometric Random Variate

```
long Geometric(double p)      /* use 0.0 < p < 1.0 */
{
    return ((long) (log(1.0 - Random()) / log(p)));
}
```

- The mean of a *Geometric*(p) random variate is $p/(1-p)$
- If p is close to zero then the mean will be close to zero
- If p is close to one, then the mean will be large

Example 3.1.4

- Assume that jobs arrive at random with a steady-state arrival rate of 0.5 jobs per minute
- Assume that Job service times are composite with two components
  - The *number* of service tasks is $1 + \mathit{Geometric}(0.9)$
  - The *time* (in minutes) per task is $\mathit{Uniform}(0.1, 0.2)$

Get Service Method

```
double GetService(void)
{
    long k;
    double sum = 0.0;
    long tasks = 1 + Geometric(0.9);
    for (k = 0; k < tasks; k++)
        sum += Uniform(0.1, 0.2);
    return (sum);
}
```

Example 3.1.4

- The theoretical steady-state statistics for this model are

$\bar{r}$	$\bar{w}$	$\bar{d}$	$\bar{s}$	$\bar{l}$	$\bar{q}$	$\bar{x}$
2.00	5.77	4.27	1.50	2.89	2.14	0.75

- The arrival rate, service rate, and utilization are identical to Example 3.1.3
- The other four statistics are significantly larger
- Performance measures are sensitive to the choice of service time distribution

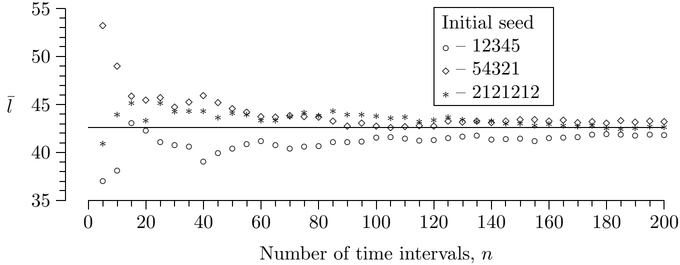
Simple Inventory System

- Program `sis2` has randomly generated demands using an *Equilikely*(a, b) random variate
- Using random data, we can study transient and steady-state behaviors
- If $(a, b) = (10, 50)$ and $(s, S) = (20, 80)$, then the approximate steady-state statistics are

$\bar{d}$	$\bar{o}$	$\bar{u}$	$\bar{l}^{+}$	$\bar{l}^{-}$
30.00	30.00	0.39	42.86	0.26

Example 3.1.6

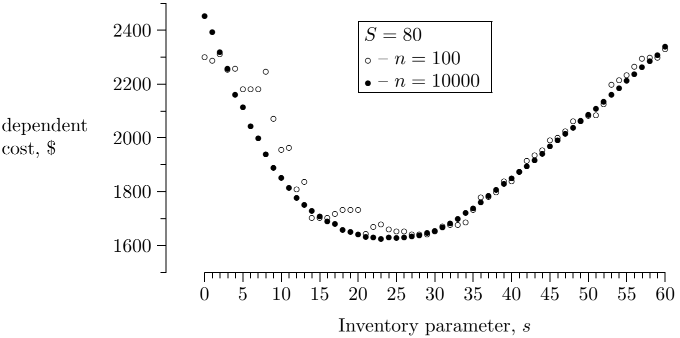
- The average inventory level $\bar{l} = \bar{l}^{+} - \bar{l}^{-}$ approaches steady state after several hundred time intervals



- Convergence is slow, erratic, and dependent on the initial seed

Example 3.1.7

- If we fix S, we can find the optimal cost by varying s
- Recall that the dependent cost ignores the fixed cost of each item



Example 3.1.7

- Using a fixed initial seed guarantees the *exact* same demand sequence
  - Any changes to the system are caused solely by the change of s
- A steady state study of this system is unreasonable
  - All parameters would have to remain fixed for many years
  - When $n = 100$ we simulate approximately 2 years
  - When $n = 10000$ we simulate approximately 192 years

Statistical Considerations

- With Variance Reduction, we eliminate all sources of variance except one
  - Transient behavior will always have some inherent uncertainty
  - We kept the same initial seed and changed only s
- Robust Estimation occurs when a data point that is not sensitive to small changes in assumptions
  - Values of s close to 23 have essentially the same cost
  - Would the cost be more sensitive to changes in S or other assumed values?