Discrete-Event Simulation:

A First Course

Section 2.1: Lehmer Random Number Generators: Introduction

©2006 Pearson Ed., Inc.
0-13-142917-5

Section 2.1: Lehmer Random Number Generators: Introduction

- `ssq1` and `sis1` require input data from an outside source
- The usefulness of these programs is limited by amount of available data
  - What if more data needed?
  - What if the model changed?
  - What if the input data set is small or unavailable?
- A random number generator address all problems
  - It produces real values between 0.0 and 1.0
  - The output can be converted to *random variate* via mathematical transformations

 ©2006 Pearson Ed., Inc. 0-13-142917-5

Random Number Generators

- Historically there are three types of generators
  - table look-up generators
  - hardware generators
  - algorithmic (software) generators
- Algorithmic generators are widely accepted because they meet all of the following criteria:
  - randomness - output passes all reasonable statistical tests of randomness
  - controllability - able to reproduce output, if desired
  - portability - able to produce the same output on a wide variety of computer systems
  - efficiency - fast, minimal computer resource requirements
  - documentation - theoretically analyzed and extensively tested

 ©2006 Pearson Ed., Inc. 0-13-142917-5

Algorithmic Generators

- An *ideal* random number generator produces output such that *each* value in the interval $0.0 < u < 1.0$ is *equally likely* to occur
- A *good* random number generator produces output that is (almost) statistically indistinguishable from and ideal generator
- We will construct a good random number generator satisfying all our criteria

 ©2006 Pearson Ed., Inc. 0-13-142917-5

Conceptual Model

- Conceptual Model:
  - Choose a *large* positive integer m. This defines the set $\mathcal{X}_m = \{1, 2, \ldots, m-1\}$
  - Fill a (conceptual) urn with the elements of $\mathcal{X}_m$
  - Each time a random number u is needed, draw an integer x "at random" from the urn and let $u = x/m$
- Each draw *simulates* a sample of an independent identically distributed sequence of $\mathit{Uniform}(0,1)$
- The possible values are $1/m, 2/m, \ldots (m-1)/m$.
- It is important that m be large so that the possible values are densely distributed between 0.0 and 1.0

 ©2006 Pearson Ed., Inc.

Conceptual Model

- 0.0 and 1.0 are impossible
  - This is important for some random variates
- We would like to draw from the urn with replacement
- For practical reasons, we will draw without replacement
  - If m is large and the number of draws is small relative to m, then the distinction is largely irrelevant

 ©2006 Pearson Ed., Inc. 0-13-142917-5

Lehmer's Algorithm

- *Lehmer's algorithm* for random number generation is defined in terms of two fixed parameters:
  - *modulus* m, a fixed large *prime* integer
  - *multiplier* a, a fixed integer in $\mathcal{X}_m$
- The integer sequence $x_0, x_1, \ldots$ is defined by the iterative equation

$$x_{i+1} = g(x_i)$$

  with

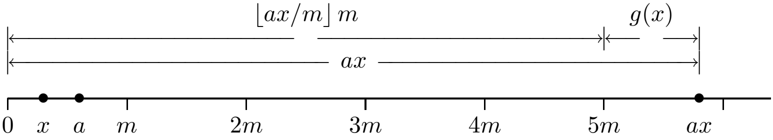
$$g(x) = ax \bmod m$$

- $x_0 \in \mathcal{X}_m$ is called the *initial seed*

Lehmer Generators

- Because of the mod operator, $0 \leq g(x) < m$
- However, 0 must not occur since $g(0) = 0$
  - Since m is prime, $g(x) \neq 0$ if $x \in \mathcal{X}_m$.
  - If $x_0 \in \mathcal{X}_m$, then $x_i \in \mathcal{X}_m$ for all $i \geq 0$.
- *If* the multiplier and prime modulus are chosen properly, a Lehmer generator is statistically indistinguishable from drawing from $\mathcal{X}_m$ with replacement.
- Note, there is nothing random about a Lehmer generator
  - For this reason, it is called a *pseudo-random* generator

Intuitive Explanation



- When ax is divided by m, the remainder is "likely" to be any value between 0 and $m - 1$
- Similar to buying numerous identical items at a grocery store with only dollar bills.
  - a is the price of an item, x is the number of items, and $m = 100$.
  - The change is likely to be any value between 0 and 99 cents

Parameter Considerations

- The choice of m is dictated, in part, by system considerations
  - On a system with 32-bit 2's complement integer arithmetic, $2^{31} - 1$ is a natural choice
  - With 16-bit or 64-bit integer representation, the choice is not obvious
  - In general, we want to choose m to be the largest representable prime integer
- Given m, the choice of a must be made with great care

Example 2.1.1

- If $m = 13$ and $a = 6$ with $x_0 = 1$ then the sequence is
$$1, 6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11, 1, \ldots$$
  - The ellipses indicate the sequence is periodic
- If $m = 13$ and $a = 7$ with $x_0 = 1$ then the sequence is
$$1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2, 1 \ldots$$
  - Because of the 12, 6, 3 and 8, 4, 2, 1 patterns, this sequence appears "less random"
- If $m = 13$ and $a = 5$ then
$$1, 5, 12, 8, 1, \ldots \quad \text{or} \quad 2, 10, 11, 3, 2, \ldots \quad \text{or} \quad 4, 7, 9, 6, 4, \ldots$$
  - This less-than-full-period behavior is obviously undesirable

Central Issues

- For a chosen (a, m) pair, does the function $g(\cdot)$ generate a full-period sequence?
- If a full period sequence is generated, how random does the sequence appear to be?
- Can $ax \bmod m$ be evaluated efficiently and correctly?
    - Integer overflow can occur when computing ax

Full Period Considerations

- From Appendix B, $b \bmod a = b - \lfloor b/a \rfloor a$
- There exists a non-negative integer $c_i = \lfloor ax_i/m \rfloor$ such that

  $$x_{i+1} = g(x_i) = ax_i \bmod m = ax_i - mc_i$$

  Therefore (by induction)

$$
\begin{aligned}
x_1 &= ax_0 - mc_0 \\
x_2 &= ax_1 - mc_1 = a^2x_0 - m(ac_0 + c_1) \\
x_3 &= ax_2 - mc_2 = a^3x_0 - m(a^2c_0 + ac_1 + c_2) \\
&\vdots \\
x_i &= ax_{i-1} - mc_{i-1} = a^ix_0 - m(a^{i-1}c_0 + a^{i-2}c_1 + \ldots + c_{i-1})
\end{aligned}
$$

 ©2006 Pearson Ed., Inc.

Full Period Considerations

- Since $x_i \in \mathcal{X}_m$, we have $x_i = x_i \bmod m$. Therefore, letting $c = a^{i-1}c_0 + a^{i-2}c_1 + \ldots + c_{i-1}$, we have
$$x_i = a^i x_0 - mc = (a^i x_0 - mc) \bmod m = a^i x_0 \bmod m$$

Theorem (2.1.1)

If the sequence $x_0, x_1, x_2, \ldots$ is produced by a Lehmer generator with multiplier a and modulus m then

$$x_i = a^i x_0 \bmod m$$

- It is an eminently bad idea to compute x_i by first computing a^i
- Theorem 2.1.1 has significant theoretical value

Full Period Considerations

$(b_1 b_2 \ldots b_n) \bmod a = (b_1 \bmod a)(b_2 \bmod a) \cdots (b_n \bmod a) \bmod a$

Therefore

$$x_i = a^i x_0 \bmod m = (a^i \bmod m) x_0 \bmod m$$

- Fermat's little theorem states that if p is a prime which does not divide a, then $a^{p-1} \bmod p = 1$.
  Thus, $x_{m-1} = (a^{m-1} \bmod m) x_0 \bmod m = x_0$

Theorem (2.1.2)

If $x_0 \in \mathcal{X}_m$ and the sequence $x_0, x_1, x_2 \ldots$ is produced by a Lehmer generator with multiplier a and (prime) modulus m then there is a positive integer p with $p \leq m - 1$ such that $x_0, x_1, x_2 \ldots x_{p-1}$ are all different and

$$x_{i+p} = x_i \qquad i = 0, 1, 2, \ldots$$

That is, the sequence is periodic *with* fundamental period *p. In addition $(m-1) \bmod p = 0$.*

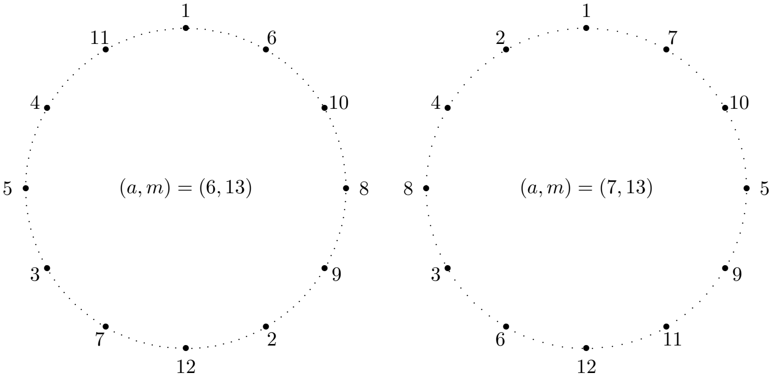
Full Period Multipliers

- If we pick *any* initial seed $x_0 \in \mathcal{X}_m$ and generate the sequence $x_0, x_1, x_2, \ldots$ then x_0 will occur again
- Further x_0 will reappear at index p that is either $m-1$ or a divisor of $m-1$
- The pattern will repeat forever
- We are interested in choosing *full-period multipliers* where $p = m-1$

 ©2006 Pearson Ed., Inc.

Example 2.1.2

- Full-period multipliers generate a virtual *circular list* with $m - 1$ distinct elements.



 ©2006 Pearson Ed., Inc.

Finding Full Period Multipliers

Algorithm 2.1.1

```
p = 1;
x = a;
while (x != 1) {
    p++;
    x = (a * x)% m;        /* beware of a * x overflow */
}
if(p == m - 1)
    /* a is a full-period multiplier */
else
    /* a is not a full-period multiplier */
```

- This algorithm is a slow-but-sure way to test for a full-period multiplier

 ©2006 Pearson Ed., Inc.

Frequency of Full-Period Multipliers

- Given a prime modulus m, how many corresponding full-period multipliers are there?

Theorem (2.1.3)

If m is prime and $p_1, p_2, \ldots, p_r$ are the (unique) prime factors of $m-1$ then the number of full-period multipliers in $\mathcal{X}_m$ is

$$\frac{(p_1-1)(p_2-1)\cdots(p_r-1)}{p_1 p_2 \cdots p_r}(m-1)$$

- **Example 2.1.3** If $m = 13$ then $m - 1 = 12 = 2^2 \cdot 3$. Therefore, there are $\frac{(2-1)(3-1)}{2 \cdot 3}(13-1) = 4$ full-period multipliers (2, 6, 7, and 11)

Example 2.1.4

- If $m = 2^{31} - 1 = 2147483647$ then since the prime decomposition of $m - 1$ is

$$m - 1 = 2^{31} - 2 = 2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331$$

the number of full period multipliers is

$$\left(\frac{1 \cdot 2 \cdot 6 \cdot 10 \cdot 30 \cdot 150 \cdot 330}{2 \cdot 3 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331}\right)(2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331) = 534600000$$

- Therefore, approximately 25% of the multipliers are full-period

Finding All Full-Period Multipliers

- Once one full-period multiplier has been found, then all others can be found in $\mathcal{O}(m)$ time

Algorithm 2.1.2

```
i = 1;
x = a;
while (x != 1) {
    if(gcd(i, m − 1) == 1)
        /* a^i mod m is a full-period multiplier*/
    i++
    x = (a * x) %  m;      /* beware a * x overflow */
}
```

Finding All Full-Period Multipliers

Theorem (2.1.4)

If a is any full-period multiplier relative to the prime modulus m then each of the integers

$$a^i \bmod m \in \mathcal{X}_m \qquad i = 1, 2, 3, \ldots, m-1$$

is also a full-period multiplier relative to m if and only if i and $m-1$ *are relatively prime*

 ©2006 Pearson Ed., Inc. 0-13-142917-5

Example 2.1.5

- If $m = 13$ then we know from Example 2.1.3 there are 4 full period multipliers. From Example 2.1.1 $a = 6$ is one. Then, since 1, 5, 7, and 11 and relatively prime to 13

$$6^1 \bmod 13 = 6 \qquad 6^5 \bmod 13 = 2$$
$$6^7 \bmod 13 = 7 \qquad 6^{11} \bmod 13 = 11$$

- Equivalently, if we knew $a = 2$ is a full-period multiplier

$$2^1 \bmod 13 = 2 \qquad 2^5 \bmod 13 = 6$$
$$2^7 \bmod 13 = 11 \qquad 2^{11} \bmod 13 = 7$$

 ©2006 Pearson Ed., Inc.

Example 2.1.6

- If $m = 2^{31} - 1$ then from Example 2.1.4 there are 534600000 integers relatively prime to $m - 1$. The first first few are $i = 1, 5, 13, 17, 19$. $a = 7$ is a full-period multiplier relative to m and therefore

$$
\begin{aligned}
7^1 \bmod 2147483647 &= 7 \\
7^5 \bmod 2147483647 &= 16807 \\
7^{13} \bmod 2147483647 &= 252246292 \\
7^{17} \bmod 2147483647 &= 52958638 \\
7^{19} \bmod 2147483647 &= 447489615
\end{aligned}
$$

are full-period multipliers relative to m

 ©2006 Pearson Ed., Inc.