Random Number Generation and Monte Carlo Simulation

Lawrence M. Leemis and Stephen K. Park, Discrete-Event Simul A First Course, Prentice Hall, 2006

Hui Chen

Department of Mathematics and Computer Science
Virginia State University
Petersburg, Virginia

January 29, 2015

Need for Random Number Generators

- Single Server Queue and Simple Inventory System
- Two trace-driven simulation programs: *ssq1* and *sis1*
- The usefulness of these programs depends on the availability of the traces
  - What if more data is needed?
  - What if the input data set is small or unavailable?
  - What if the model changes?
- A random number generator addresses all the problems
  - It produces random real values between 0.0 and 1.0
  - The output can be converted to *random variate* via mathematical transformations

Random Number Generators: Applications

- Computer games
- Computer networking
- Computer and network security
- ...

Random Number Generators (RNG)

- Types of generators
  - Table look-up generators
  - Hardware generators
  - Algorithmic (software) generators
- Desired 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
- Algorithmic generators meet the above criteria and are widely accepted

Algorithmic Generators

- An *ideal* RNG produces output such that each value in the interval $0.0 < u < 1.0$ is equally likely to occur
- A *good* RNG produces output that is *almost* statistically indistinguishable from an ideal RNG
- We will construct a good RNG satisfying all our criteria
  - Lehmer Random Number Generators

Random Number Generators in Programming Languages

- C/C++: rand()
- Java: the java.util.Random class
- Perl: rand()
- Matlab: rand()

Open Exercises

- C/C++: test rand()
- Java: test java.util.Random class

Seeding Random Number Generators

- C/C++: srand()
- Java: java.util.Random class

Seeding RNGs: System Dependent

- Windows: use CrytoAPI's CryptGenRandom()
- Linux/Unix: use /dev/random

Open Exercises

- Windows: use CryptoAPI's CryptGenRandom()
- Linux/Unix: use /dev/random