Introduction to Computer Simulation

Simulation – The process of designing a model of a real system, implementing the model as a computer program, and conducting experiments with the model for the purpose of understanding the behavior of the system, or evaluating strategies for the operation of the system.

What is simulation used for?

(1) Analyze the system
- Change inputs
- Keep system the same
- Evaluate the output (response)

(2) Alternative system design
- Keep input the same
- Change operation of the system
- Make decisions based on output

(3) Alternative control
- Specify system
- Specify the output
- Change the input

Advantages and Disadvantages of Simulation

- **Advantages**
  - Easy to build the model
  - Can handle otherwise intractable problems
  - Not subject to so many assumptions
  - Easy for decision maker to understand
  - Cheap and safe
  - Good for comparisons
- **Disadvantages**
  - Hard to debug (randomness)
  - No guarantee of optimality
  - Hard to establish validity
  - Costly to build and maintain complex models
  - Can't produce absolute numbers

BOTTOM LINE: ***USE SIMULATION ONLY WHEN ADVANTAGES OUTWEIGH DISADVANTAGES***

Simulation Modeling

The First Law of Mentat (from Dune by Herbert):

A process cannot be understood by stopping it. Understanding must move with the process, must join it and flow with it.

Def: System – A group of objects or sets of objects united by some form of regular interaction or interdependence to perform a specified function.

Random Number Generation

- Random numbers from RNG's are not really random
- Desirable characteristics of RNG's
  - (1) Know distribution (usually U(0,1))
  - (2) Independent
  - (3) Reproducible but changeable
  - (4) Fast computation
  - (5) Small memory requirements
  - (6) Portable
  - (7) Easy to implement
  - (8) Numbers should be dense

Random Number Algorithm

- Linear Congruential Random Number Generator
  - – Additive
  - – Multiplicative
  - – Mixed
- Algorithm

$$X_i = (aX_{i-1} + c)\text{mod}(m)$$

note: mod function is the remainder function

ex. 12 mod 5 = 2

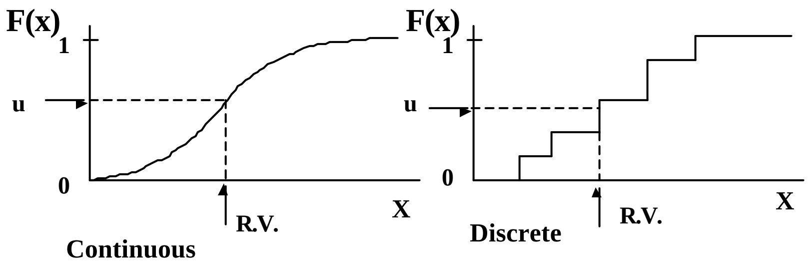
  - – X's are usually integers
  - – Values of a, c, and m determine quality of RNG
  - – Start with "seed" X_o

Non-Uniform Random Numbers

- These algorithms used to generate U(0,1) variates
- Need to generate other PRN with other distributions

(1) Inverse Method:

- Want X ~ F(•) CDF
- Draw u ~ U(0,1)
- $X = F^{-1}(u)$ where F(•) is invertible



Non-Uniform Random Numbers

(2) Convolution – Add them up

ex. Add several Bernoulli RV's to get a Binomial

(3) Composition – Use when CDF can be written as convex combination of others

- Draw U(0,1) to find the piece, then draw from that piece

(4) Acceptance - Rejection

- Useful when no CDF available
- Range of X must be finite and f(x) must have upper and lower bound

★ ***Fortunately*** *- many applications have generators for different distributions readily available as functions*

Stochastic Models

Input Distributions

Discrete Distributions

- **Bernoulli - model situation with 2 choices**
  - **Flipping a (maybe biased) coin**
  - **e.g., success/failure, right/left, 0/1**
- **Binomial - n independent Bernoulli trials**
  - **Number of successes in n trials**
- **Poisson - used to model number of occurrences of an event in some given time period**
  - **Number of arrivals to a bank in an hour**
  - **Number of accidents at an intersection in a year**
- **Geometric - number of trials until the first success**
  - **Number times light is turned on before it fails**

Stochastic Models

Input Distributions

Continuous Distributions

- **Uniform – models process when outcome is equally likely to be between a and b**
  - **Compute as $x = a + (b - a)u$, with $u \sim U(0,1)$**
- **Exponential – models time until some occurrence of some event**
  - **has range of 0 - 8**
  - **one parameter distribution $F(x) = 1 - e^{-lx}$, $x = 0$**

    $E[X] = 1/l$, Arrival rate = l
  - **Memoryless: $P[X > x + x_o \mid X > x_o] = P[X > x]$**
    - **Nice mathematical properties**
    - **Basis of queuing theory**

Input Distributions

- **Normal Distribution - Model error**
  - **- Model anything symmetrically distributed around a central point**
  - **- Hard to generate Normal Random Numbers**
    (No closed form for CDF)
  - **- Remember: -8 < x < 8**
- **Beta Distribution - very flexible, many different shapes**
  - **- Range of X ~** b **: [0,1] Parameters:** a **and** b
  - **- E[X] =** a/(a+b), **Var.[X] =** ab/[(a+b)^2(a+b+1)]
  - **- Can be used to model distribution over range [a,b]**
    Y = (b - a)X + a, E[Y] = (b - a)E[X] + a
    Var[Y] = (b - a)^{2}Var[X]

Input Distributions

- **Beta Distribution (continued)**
  - **- Can be used with limited information**
    - **- Estimate (1) optimistic (2) pessimistic**
      (3) expected outcomes
    - **- Set a = pessimistic, b = optimistic,**

$$E[Y] = \frac{\text{pessimistic} + 4(\text{exp}) + \text{optimistic}}{6}$$

$$\hat{s}_Y = \frac{\text{pessimistic} + \text{optimistic}}{6}$$

- Then solve for a **and** b

Empirical CDF

Have sample of data (e.g., completion times for jobs on machine)

$x_1, x_2, x_3, ..., x_n$

- **Get order statistics:**

  $x_{[1]}, x_{[2]}, x_{[3]}, ..., x_{[n]}$ **such that** $x_{[i]} = x_{[i+1]}$

  - **approximate:** $F(x_{[i]}) = (i-1)/(n-1)$
  - **define** $F(x)$ **for** $x_{[i-1]} < x < x_{[i]}$ **by interpolation**
- **Generate RN's using inversion**
- **Packages handle these**
- **Careful: Ignores the tails of the distribution**

Simulation Modeling (continued)

Def: <u>Entity</u> – An object in the system

ex. Phone system has the following entities

- **Phones**
- **Switchboard**
- **Operators**
- **Users**
- **Calls**

Entities can be:

- **Permanent (phone)**
- **Temporary (call)**

Simulation Modeling (continued)

Def: Attribute – Characteristic of an entity

ex. Entity phone may have the following attributes:

- **Number**
- **Location**
- **Dial type**
- **Status**
- **Color**

Attributes can be:

- **Static – Their values don't change**
- **Dynamic – Their values change during the course of the simulation**
- **Critical (important enough to be included)**
- **Non-critical (can be left out of the model)**

Simulation Modeling (continued)

Def: <u>State of the Entity</u> – the values of the dynamic attributes of the entity.

Def: <u>State of the System</u> – the combined states of all the critical entities in the system.

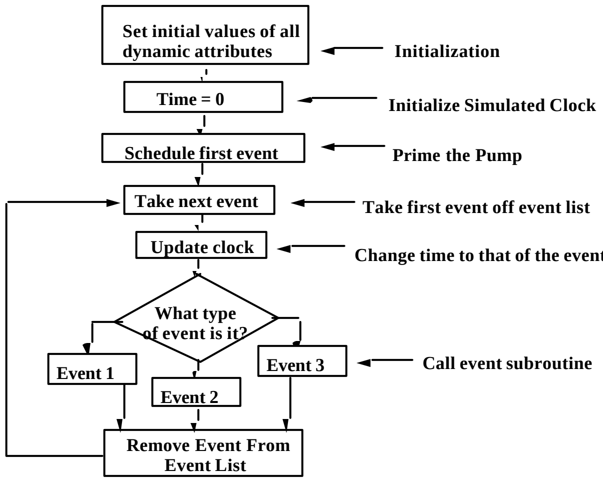
Def: <u>Event</u> – the *INSTANT IN TIME* when the value of one or more dynamic attributes of any entity of the system changes.

Def: <u>Event Step Simulation</u> – a simulation in which time advances when a new event takes place.

- **Other types:**
  - **Time step - clock advances in equal steps**
  - **Continuous time**

Stochastic Models

General Structure



Stochastic Models

Steps in Modeling

★ (1) Identify entities of the system

★ (2) Identify attributes of the entities

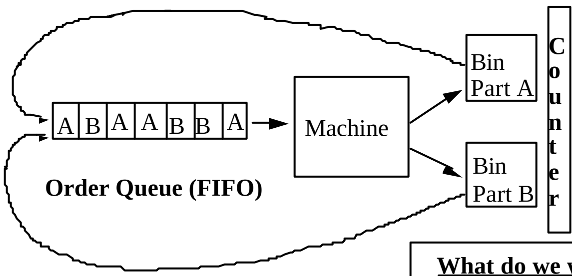
★ (3) Classify the attributes
- – Critical vs Noncritical
- – Static vs Dynamic

★ (4) Identify events by finding the instants that the value of each dynamic attribute might change

(5) Flowchart each identified event

★ ***Must be able to do for this class***

Example – Kanban System



<u>Customer</u>
- **Wants part A 40%**
- **Wants part B 60%**
- **Arrives every 14 mins**
- **Leaves if he can't get what he wants**
- **Wants only 1 part**

<u>Machine</u>
- **Makes part A in 10 minutes**
- **Makes part B in 6 minutes**
- **Takes 5 minutes to switch**

<u>What do we want to measure?</u>
- **Inventory in bins**
- **Customer satisfaction**
- **Utilization of machine**

<u>What do we want to accomplish?</u>
- **Find a policy that will**
  Minimize inventory
  Maximize customer satisfaction

Other Issues

- **Computer languages:**
  - **General purpose languages can be used**
    - Fortran
    - Pascal
    - C
  - **Simulation languages**
    - GPSS
    - Simscript
    - SLAM
- **Randomness**
  - **Random number generation**
  - **Utilities on computer**
  - **Built in to simulation languages**

Other Issues (continued)

- **How do you start? – Initialization bias**
- **How do you stop? – Stopping rules**
  - **- Number of a type of event**
  - **- Time**
- **Statistics**
  - **– How many runs?**
  - **– Experimental design**
  - **– Analysis of output**
- **Validation and Verification (Is the model any good?)**