Lecture Notes 1
Introduction to Simulation

Prof. Marvin Nakayama

- Course webpage has reading assignments and HW.
  - Be sure to check daily for announcements, updates, etc.
  - HW 1 is posted and due in two weeks.
- Mechanics of the class ... instructor, assignments, etc.
- What is simulation? Why simulation?
- Simple examples
- Simulation applications
- Steps in the simulation process

Course Logistics

- **Instructor**: Marvin K. Nakayama
  - Office: NJIT GITC 4312
  - E-mail: marvin@njit.edu
    - Include "CS 661" in subject line
    - E-mail preferred over phone
  - Phone: 973-596-3398
  - Office Hours: Wednesday, Thursday, 1:00 – 2:00, or by appt.
- Announcements posted on moodle, `http://moodle.njit.edu/`
  - Need NJIT UCID to access moodle.
  - If you don't know your UCID (e.g., Rutgers student), send me email with your full name.

My Background

- Education
  - B.A. in Math/Computer Science, UC San Diego
  - M.S. in Operations Research, Stanford University
  - Ph.D. in Operations Research, Stanford University
- Employment
  - IBM Thomas J. Watson Research Center
  - Rutgers Faculty of Management
  - Columbia University Business School
  - Columbia University Engineering School
  - New Jersey Institute of Technology

Course Materials

Webpage: http://web.njit.edu/∼marvin/cs661

- *username*: `cs661` (make sure you type in lower case)
- *password*: `random`
- All course materials are posted on the course webpage.
  - Some adapted from materials developed by Paul Glasserman (Columbia), Peter Glynn (Stanford), Peter Haas (Stanford), Garud Iyengar (Columbia), Stefanos Zenios (Stanford).
- Do not use course materials from previous semesters.
  - If anyone is caught using previous semesters' materials, I will report this to the dean of students.

Lecture Notes:

- Will be posted in pdf on the course web page at least 24 hours before class.
- You must bring printouts of lecture notes to class. **I only handed out the first few pages of Lecture Notes 2.**
- Also, other course materials will be posted and you need to print them out.

Recommended Texts (no required text):

- S. M. Ross, *Simulation*, 5th Edition, Academic Press, 2012, ISBN: 0124158250.
- P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, 2004, ISBN: 0387004513.

Additional References:

- S. Asmussen and P. W. Glynn, *Stochastic Simulation: Algorithms and Analysis*, Springer, 2007.
- J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 5th Ed., Prentice-Hall, 2009.
- P. Bratley, B. L. Fox, and L. E. Schage, *A Guide to Simulation*, 2nd Edition, Springer-Verlag, 1987.
- G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, 1996.
- W. D. Kelton, R. P. Sadowski, and N. B. Swets, *Simulation with Arena*, 5th Edition, McGraw-Hill, 2010.
- A. M. Law, *Simulation Modeling and Analysis*, 5th Edition, McGraw-Hill, 2015.
- B. R. Ripley, *Stochastic Simulation*, Wiley, 2006.

Prerequisites and Academic Integrity

- Calculus-Based Probability and Statistics
  - Course uses lots of calculus-based probability and statistics.
  - Will review this over the next few lectures.
  - **Self-assessment quiz:** Let X be a random variable having density function $f(x) = \lambda e^{-\lambda x}$ for $x \geq 0$, and $f(x) = 0$ for $x < 0$. Compute its CDF, expectation, and variance (not the sample mean and sample variance).
- Computing requirements
  - Know one high-level programming language, e.g., C or Java
  - Understand pseudo-code and be able to write code from it
- Must obey NJIT University Code for Academic Integrity.
  - Any violations will be reported to dean of students.

Grading

Class Participation/Attendance	10%
HW Assignments	20%
Midterm	35%
Final Exam	35%

- Class attendance is mandatory. If you cannot attend class, you must contact me beforehand.
- HW will not be graded carefully.
- Each HW must be stapled, or you will lose half the points.
- HW due at beginning of class. No late HW accepted.
- Tentative midterm date given in syllabus (on webpage).
- Syllabus goes into more detail. **Be sure to read it!**

What is simulation?

Simulation is the process of building a **mathematical** or **logical model** of a system or a decision problem, and **experimenting** with the model to obtain estimates of system **performance measures**.

Important terms in the definition above:

Model: A logical/mathematical abstraction of the real system that captures the essence of the problem. Includes the description of

- input variables: parameters that define the system
- output variables: parameters that define the performance measure
- mathematical/logical relationship between input and output

Experiment:

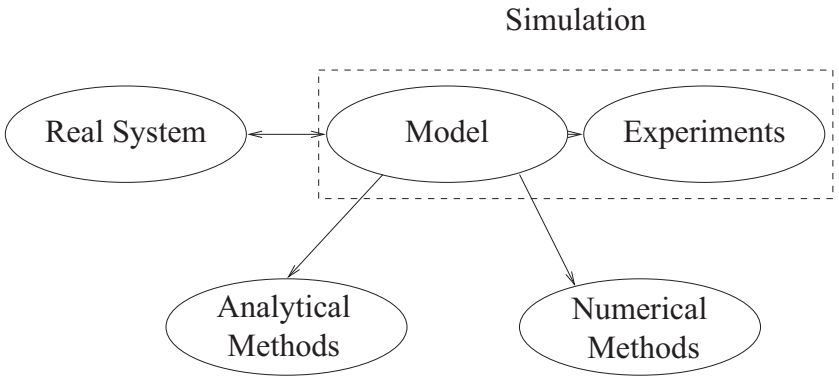
- Generate several instances of input variables and output variables
- From these, estimate the performance measures.

Performance measures:

- Properties of the system that have relevance to the problem.

Other methods of analysis

Modeling is not limited to *simulation* ...



Modeling is an essential interface between the real system and

- Analytical method ... dynamic programming, probabilistic analysis, etc.
- Numerical methods ... linear programming, integer programming, non-linear programming, etc.

Often not a one-shot process

- build model
- run experiments
- go back and update model
- may take many iterations

Statistician George E. P. Box:
"Essentially, all models are wrong, but some are useful."

Dave's Candy Shop

- Boxes of special chocolates ... Valentine's Day Chocolate Massacre
- Cost $7.50 a box.
- *Before* Feb 14th, sell for $12 a box.
- *After* Feb 14th, boxes discounted 50% and are all sold off.
- Historically, demand before Feb 14th has been between 40 to 90 boxes.
- **Goal:** Maximize average profit

Model

1. **Input variables**
   (a) Quantity ordered Q: *decision variable* ... assumed to be 40, 50, 60, 70, 80 or 90 boxes.
   (b) Random Demand D: *uncontrolled variable* ... assumed uniformly distributed on $[40, 90]$ ... approximation
2. **Output variables**: Profit P
3. **Input/Output relation**:

$$P = \begin{cases} 12D - 7.5Q + 6(Q - D), & \text{if } D \leq Q, \\ 12Q - 7.5Q, & \text{if } D > Q \end{cases}$$

Note that P is random since it depends on random demand D.

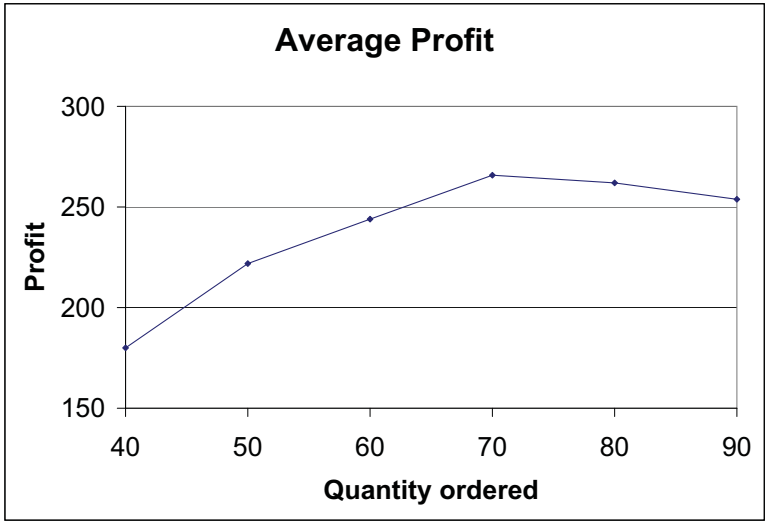
Performance measure: Expected (average) profit $\mathcal{E}[P]$.

Experiments:

1. In steps, set $Q = 40, 50, \ldots, 90$.
2. For each value of Q, generate N replications of the demand $D_1, \ldots D_N$ uniformly distributed in $[40, 90]$. How?
3. Calculate corresponding profit $P_1, \ldots, P_N$ by formula above.
4. Approximate expected profit $\mathcal{E}[P]$ by sample average:

$$\mathcal{E}[P] \approx \frac{1}{N} \sum_{i=1}^{N} P_i$$

   - Why is sample average a good approximation ?
   - How close is the approximation to the real value ?
   - How large should N be to give a good estimate ?



- Could we have done better by using some other means? Yes.
- Recall profit

$$P = \begin{cases} 12D - 7.5Q + 6(Q - D), & \text{if } D \leq Q, \\ 12Q - 7.5Q, & \text{if } D > Q \end{cases}$$

  - Demand is uniformly distributed $[40, 90]$, i.e., profit has density function $f(x) = 1/50$ for $x \in [40, 90]$
- Using simple probability, we get the expected profit $\mathcal{E}[P]$ as a function of the quantity ordered Q as follows:

$$\begin{aligned} \mathcal{E}[P] &= \int_{40}^{Q} \big[12x - 7.5Q + 6(Q - x)\big]\frac{1}{50}\,dx + \int_{Q}^{90} \frac{4.5Q}{50}\,dx \\ &= \big[-3Q^2 + 465Q - 4800\big]\frac{1}{50}. \end{aligned}$$

- A quadratic function, so use calculus to find optimal Q:

$$Q^* = 78 \text{ or } 77$$

- Simulation expensive
  - Have to run experiment for each value of Q
- But what if
  - the demand distribution was complicated ?
  - the relation between P and Q were more complicated ?

Simulation still works just the same !

Why simulation ?

- The *simulator* recreates *elements of reality* that are of interest
- Simulation allows one to modify various system parameters and study the effects
  - experiment with "reality" without the risks/limits that reality entails
- Extremely flexible system analysis tool
  - complex systems that are not amenable to numerical and analytical analysis often can be simulated easily
- Special simulation techniques allow one to "compress" time
  - allow one to "observe" very rare events and save time/money

When not to use simulation ?

- Simulations are slow and never accurate
  - so do not use them when either analytical/numerical methods can give you the answer.
  - Was a bad idea to use simulation for analyzing Dave's candy shop problem

Dirty Dan's Car wash

- Cars arrive at an average rate of 15 cars per hour, i.e., one every 4 minutes on average.
- It takes on average 3 mins to wash a car ... everything looks great !
- But work seems to pile up and customers have to wait !
- Dan wants to estimate
  - the average waiting time
  - fraction of time the wash is busy

  before he improves his facility ...

Model

1. **Input variables**:

   (a) $\{T_i : i \geq 1\}$, where T_i is arrival time of ith customer
   - Arrival rate $\lambda = \frac{1}{4}$.
   - Define *inter-arrival* times A_i, $i \geq 1$:

$$A_0 = 0,$$
$$T_0 = 0,$$
$$A_i = T_i - T_{i-1}, \quad i \geq 1.$$

   - Assume inter-arrival times are independent exponential random variables with mean $\frac{1}{\lambda} = 4$.
   - i.e., Poisson process with rate λ.

   (b) $\{S_i : i \geq 1\}$, where S_i is service time of ith customer
   - S_i, $i \geq 1$, assumed to be independent with mean $\frac{1}{\mu} = 3$.

2. **Output variables**:

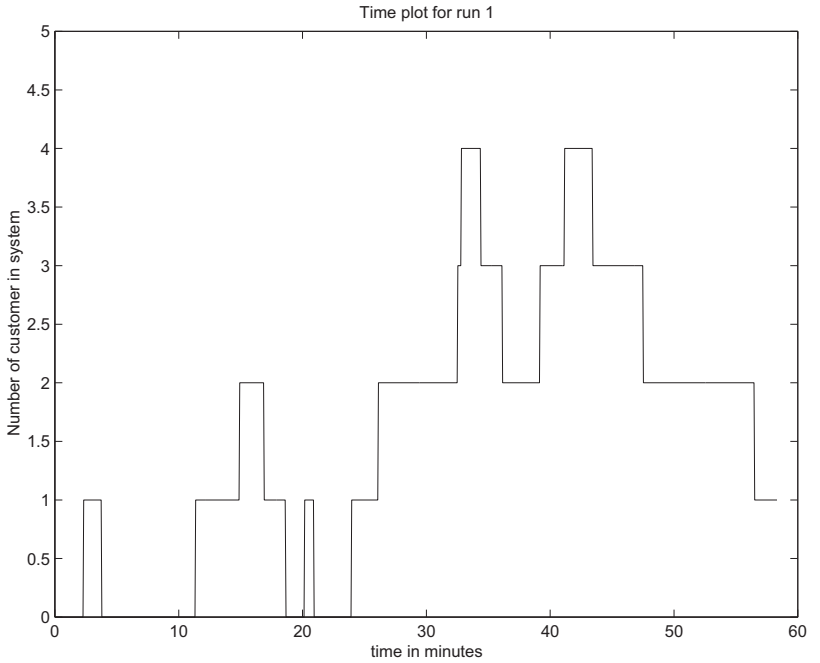
   (a) Waiting times $\{W_i : i \geq 1\}$, of customers.
   - W_i is waiting time of ith customer.

   (b) Idle times $\{I_i : i \geq 1\}$.
   - I_i is amount of time car wash is idle before customer i.

3. **Input/Output relation**: Need to first decide on the service scheme ... assume that the car wash is first-come-first-served

   Relate
   - Idle times $\{I_i : i \geq 0\}$ and waiting times $\{W_i : i \geq 1\}$
   - to inter-arrival times $\{A_i : i \geq 1\}$ and service times $\{S_i : i \geq 1\}$.

   Performance measures: $\mathcal{E}[W]$ and $\mathcal{E}[I]$.



Waiting Times:

- Car i
  - arrives at time T_i
  - waits W_i minutes
  - requires S_i minutes of service
  - departs at time

$$D_i = T_i + W_i + S_i.$$

Waiting Times:

- Recall: Car i leaves at time $D_i = T_i + W_i + S_i$.
- Car $(i+1)$ arrives at time T_{i+1}
  - If $T_{i+1} \geq D_i$, then car $(i+1)$ does not wait, so

$$W_{i+1} = 0 \quad \text{when} \quad D_i - T_{i+1} \leq 0$$

  - If $T_{i+1} < D_i$, then car $(i+1)$ has a wait of

$$W_{i+1} = D_i - T_{i+1} \quad \text{when} \quad D_i - T_{i+1} > 0$$

  - Combining the two cases gives

$$\begin{aligned} W_{i+1} &= \max(D_i - T_{i+1}, 0) \\ &= \max(W_i + S_i + T_i - T_{i+1}, 0) \\ &= \max(W_i + S_i - A_{i+1}, 0) \end{aligned}$$

since $A_{i+1} = T_{i+1} - T_i$.

Idle Times:

- Car i departs at time $D_i = T_i + W_i + S_i$
- Car $(i+1)$ arrives at time T_{i+1}
  - If $T_{i+1} \leq D_i$, then no idle time, so

$$I_{i+1} = 0 \quad \text{when} \quad T_{i+1} - D_i \leq 0$$

  - If $T_{i+1} > D_i$, then idle time is

$$I_{i+1} = T_{i+1} - D_i \quad \text{when} \quad T_{i+1} - D_i > 0$$

  - Combine the two cases as

$$\begin{aligned} I_{i+1} &= \max(T_{i+1} - D_i, 0) \\ &= \max(T_{i+1} - T_i - W_i - S_i, 0) \\ &= \max(A_{i+1} - W_i - S_i, 0) \end{aligned}$$

since $A_{i+1} = T_{i+1} - T_i$.

Experiments

1. Generate N inter-arrival times $A_1, \ldots, A_N$.
   - Set $T_0 = 0$ and $T_i = A_i + T_{i-1}$.
2. Generate N service times $S_1, \ldots, S_N$.
3. Calculate
   - sequence of waiting times $\{W_i : i \geq 1\}$ and
   - sequence of idle times $\{I_i : i \geq 1\}$.
4. Approximate

$$\begin{aligned} \mathcal{E}[W] &\approx \frac{1}{N} \sum_{i=1}^{N} W_i, \\ \mathcal{E}[I] &\approx \frac{1}{N} \sum_{i=1}^{N} I_i. \end{aligned}$$

Remarks:

- How many customers N should we simulate ?
- Since the queue starts out empty, W_i are initially biased towards zero.
  - How does one correct for this ?

Summary

Model is mathematical abstraction that captures enough of system to give good estimates of performance measure.

Issues in modeling ...

- Input variables:
  - Should we use historical data or generate the variables via some probability distribution?
  - How do we determine an appropriate distribution?
- Input/Output relations: Relatively easy if model is well constructed.

Models can be of different types:

– Static vs. Dynamic: Dave's Candy Shop vs. Dan's Car Wash
– Deterministic vs. Stochastic (variables)
– Continuous vs. Discrete (variables)

Experimenting involves manipulating the input to the model to *estimate* the performance measure.

Issues in experimenting ...

- Input variables: How does one generate the input from a particular distribution ?
- Performance measures:
  - How close is the estimate to the real thing?
  - Is there a way of ensuring the estimate will be close if one simulates "long" enough?
  - Can one do better than simply generating independent random variables?

Basic paradigms of simulation

- Common problem in simulation is to estimate mean

$$\mu \;=\; \mathcal{E}[X] \;=\; \int x\, f(x)\, dx$$

  where X is random variable with probability density function f.
- General approach
  1. Generate sample of n independent and identically distributed observations $X_1, X_2, \ldots, X_n$ using density f.
  2. Estimate of μ is

$$\widehat{\mu}_n \;=\; \frac{1}{n} \sum_{i=1}^{n} X_i$$

- Questions:
  - How close is estimate $\widehat{\mu}_n$ to μ?
  - How many observations n should we collect?

Basic paradigms of simulation

- **Monte Carlo Simulation**: Integration by simulation ...

  Calculate $\alpha \equiv \int_S g(x)\, h(x)\, dx, \quad h(x) \geq 0.$

- Assume $\int_S h(x)\, dx < \infty$, so we can normalize $h(x)$ by defining

$$f(x) \;=\; \frac{h(x)}{\int_S h(u)\, du}$$

- Now $f(x)$ is a density since $f(x) \geq 0$ for all x and

$$\int_S f(y)\, dy \;=\; \frac{1}{\int_S h(u)\, du} \int_S h(y)\, dy \;=\; 1$$

so

$$\begin{aligned} \int_S g(x)h(x)\, dx \;&=\; \left(\int_S h(u)\, du\right) \int_S g(x) f(x)\, dx \\ &=\; \left(\int_S h(u)\, du\right) \mathcal{E}_f[g(X)] \end{aligned}$$

where X is a random variable with density f

- Recall

$$\alpha \;\equiv\; \int_S g(x)h(x)\, dx \;=\; \left(\int_S h(u)\, du\right) \mathcal{E}_f[g(X)]$$

- Thus, we can estimate α as follows:

  1. Generate $X_1, \ldots, X_N$ according to density f
  2. Approximate α by

$$\int_S g(x)\, h(x)\, dx \;\approx\; \left(\int_S h(u)\, du\right) \left[\frac{1}{N} \sum_{i=1}^{N} g(X_i)\right]$$

- Applications:
  - Revenue management
  - Forecasting and stochastic programming
  - Finance: Option pricing, security pricing, etc.
  - Computer graphics: rendering 3-D objects

System Simulation

- Answers questions about systems that have
  - complex dynamics
  - complicated input/output relations
  - situations where analytical solutions do not exist

- Discrete-event system simulation ...
  - models are discrete, dynamic, stochastic
  - "state" of system changes instantaneously at countable number of (time) points

Applications

- OR/MS: queueing, inventory systems, job shop scheduling, production lines, supply chain management, logistics etc.
- EE/CS:
  Communication: design of networks, error rates for codes and communication systems.
- AI: Training Bayesian Belief networks, computational biology.
- Economics: General equilibrium models, cooperative games, international conflict, decision sciences, E-commerce.
- Social sciences: Agent-based simulation of decisions in a social organization, learning in a Skinnerian setting
- General dynamical systems: Petri-nets, autonomous vehicles, etc.

Simulation is a very flexible system analysis tool

But ...

- not the most efficient ...
- intricately tied to the performance measure
  - simulation program has to be redone if it changes

Steps in Simulation

Step 1: **Conceptual Model of the real system**

- Understand and define the problem
- Identify the goals and the output performance measures
- Determine input vars, output vars and input/output relation

Cardinal rule:

- start with the simplest model
- then embellish and enrich them as needed.

Hard to teach ... can be learned only by experience

Will try to give a feel for it by going over more examples in more detail ...

Step 2: **Build the Simulator (experiment phase)**

- Collect necessary data and determine appropriate distributions of random inputs
- Write code to generate inputs from chosen distributions.
- Write code to generate the output using the input/output relation

This will be covered in detail ...

Step 3: **Verification and Validation**

- *Verification*: Ensure that the simulator does what it is supposed to do (debugging!)
- *Validation*: Ensure that the model is a *reasonable* representation of reality.

  (a) "Face validity": ask human experts

  (b) Historical data validity: check performance on historical data.

Will talk about it but not really cover it in much detail ...
more a part of constructing a real system

Step 4: **Design and conduct experiments**

Step 5: **Analyze the output**

- Compute estimates of the performance measures
- Calculate error bounds on the performance measure:
  - Transient measures: method of replications ...
  - Steady-state measures: replicate/delete method, batch means method, regenerative method
- Decide on when to stop simulating ... run-length control

Will cover all of this at varying level of details

Step 6: **Improving efficiency of simulator**

- Memory and run time
  - Coding tricks that we will not get into ...
- Statistical efficiency
  - Improving the "rate" at which the estimate of the performance measure converges to the actual value
  - has to do with reducing variance, so-called *variance reduction*.

Statistical efficiency will be covered in the last part of the course.

Commercial Software

- Common simulation steps built-in ... just put them together
- New ones have nifty GUI's ... user models and simulator does rest
- Some popular general purpose simulation packages:
  1. GPSS (*G*eneral *P*urpose *S*imulation *S*oftware): oldest and very flexible ... hard to use
  2. ARENA: A simple GUI-based simulator (download from course website).
  3. Crystal Ball, @Risk, and PopTools: Simple Excel add-ins

Many other simulation packages available; e.g., see
`http://www.orms-today.org/surveys/Simulation/Simulation.html`

In this class we will

- focus on understanding how simulations generically work
- also demonstrate how to implement the ideas using ARENA and Excel with add-in for Monte Carlo.