IERG 3050: Simulation and Statistical Analysis
Week 1

Bolei Zhou
Department of Information Engineering
Outline

• Lecture 1: Course overview and logistics
• Lecture 2: Introduction to simulation and examples of simulation

• Special thanks to Prof. Pascal Vontel, Prof. Minghua Chen, Rosana Chan, Prof. Angele Zhang, Prof. Jianwei Huang for contributing to the slides
Course Logistics

• Instructor: Bolei Zhou
  Office hour: every Tuesday 16:30 – 18:00 in SHB 717
• TA: Yinghao Xu
  Office hour: TBD
• Course Website: https://course.ie.cuhk.edu.hk/~ierg3050/
• Piazza: https://piazza.com/cuhk.edu.hk/fall2019/ierg3050/home
Course Objectives

• Introduce simulation as a network and system design strategy
• Teach students to do
  • simulation and modeling
  • statistical analysis
  • system design
• After the course, you will gain:
  • understanding in queueing theory and statistics
  • Practical experience in simulation
Course Textbook

- Averill M. Law, Simulation Modeling and Analysis, 5th Edition
What is Simulation

• Simulation is the imitation of the operation of a real-world process or system over time

• The act of simulating something first requires a model to be developed; this model represents the key characteristics or behaviors/functions of the selected physical or abstract system
  • The model represents the system itself
  • The simulation represents the operations of the system over time
Why Simulation

• To understand existing (natural or man-made) systems
• To build new systems
• Show the eventual real effects of alternative conditions and courses of action
• Very often, the real system cannot be engaged, because it may not be accessible,
  • it may be dangerous or unacceptable to engage,
  • it is being designed but not yet built,
  • it may simply not exist.
Examples of Simulation: Weather Simulation
Examples of Simulation: Crowd Simulation

Crowds in real world

Self-Driven Particles
Examples of Simulation: Estimating Pi

Area of Circle = $\pi r^2$

Area of Square = $4r^2$

$P($Rain drop landing in the circle$) = \frac{\pi r^2}{4r^2}$

A example of Monte Carlo method!

https://www.youtube.com/watch?v=VJTFIqlQ4TU
Key Issues in Simulation

• **Collecting** valid and relevant source information about the system, including key characteristics and behaviors)

• Be careful about the **setting, approximations, and assumptions** within the simulation

• Justify the **fidelity and validity** of the simulation outcomes (data)

• **Statistics** is the study of the collection, organization, analysis, interpretation and presentation of data.
Relevance to You

Is this course related to your career? Yes!

How:

• **If you will work in IT industry:** system modeling and design is very important in middle to large scale IT projects, which require fundamental knowledge in queueing theory and statistics

• **If you will work in finance or big-data industry:** modeling and simulation are two fundamental skills to establish and test your ideas on algorithms/systems

• **If you will pursue further graduate studies:** simulation is essential for researchers in information engineering (networking, communication, computer vision, reinforcement learning)
Tentative Syllabus

• Part One: Overview
• Part Two: Math Theory
  • Queueing systems
  • Probability and statistics
• Part Three: Simulation Basics
  • Selecting input probability distribution
  • Random number generation
  • Output data analysis
• Part Four: Advanced Simulation Techniques
  • System comparison
  • Data mining and machine learning techniques
Grading Scheme

• Five sets of homework or quizzes: 20% in total
• Project: 30%
• Final exam: 50%

• Homeworks 1,3,5 will be handed in due date

• Two in-class quizzes:
  • Homework 2 and 4 will help you prepare for Quiz 1 and 2, as the problems will be similar
  • 15 min for quiz: 5-min discussion and 10-min problem solving on your own
Course Project (30% of total grade):

• An integral part of the course for applying what you have learned
• Two students in a group (currently 29 enrolled students)
• Each group is required to propose its own unique topic and we will coordinate to ensure different groups have different topics
• Milestones
  • late Sep. 2019: form a 2-person group and submit project proposal (5%).
  • end of Oct. 2019: a check-point report (5%).
  • late Nov. 2019: oral presentation session where individual groups explain their projects to the whole class (5%), 10-min presentation per group.
  • end of semester: final report with detail statistical analysis (15%).
• More detail and example documents will be available soon.
Final Remarks

• Read the well-written textbook
• Ask for help whenever you need
• Work hard and think independently
Modeling Crowd Behaviors using Mixture of Dynamic Agents

Modeling Crowd Behaviors using Mixture of Dynamic Agents

- **Beliefs:** $B = (\mu^a, \Phi^a, \mu^e, \Phi^e)$
  
  $p(x_e) = N(x_e | \mu^e, \Phi^e)$,
  
  $p(x_o) = N(x_o | \mu^e, \Phi^e)$.

- **Dynamics** $D = (A, \Gamma)$
  
  $x_t = Ax_{t-1} + w_t, \quad p(x_t | x_{t-1}) = N(x_t | Ax_{t-1}, \Gamma)$,
  
  $y_t = Cx_t + e_t, \quad p(y_t | x_t) = N(y_t | Cx_t, \Sigma)$,

  linear dynamic system with affine transform.

- **Timings** Poisson process
  
  $p(n; \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}$

**Learning Collective Crowd Behaviors with Dynamic Pedestrian-Agents.**


End of the First Class

• Readings (preview of the next lectures)

Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.8) [5th edition]
Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.9) [4th edition]

Learning Collective Crowd Behaviors with Dynamic Pedestrian-Agents.

Measuring Crowd Collectiveness.
Bolei Zhou, Xiaou Tang, Hepeng Zhang and Xiaogang Wang
IEEE transaction on Pattern Analysis and Machine Intelligence (PAMI), 2014.
## Tutorial Timeslot scheduling

<table>
<thead>
<tr>
<th>Start Time</th>
<th>Thursday</th>
<th>Start Time</th>
<th>Thursday</th>
<th>Start Time</th>
<th>Friday</th>
<th>Start Time</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:00</td>
<td>-6</td>
<td>16:00</td>
<td>-2</td>
<td>13:00</td>
<td>-8</td>
<td>16:00</td>
<td>-6</td>
</tr>
<tr>
<td>13:30</td>
<td>-6</td>
<td>16:30</td>
<td>-4</td>
<td>13:30</td>
<td>-8</td>
<td>16:30</td>
<td>-5</td>
</tr>
<tr>
<td>14:00</td>
<td>-4</td>
<td>17:00</td>
<td>-2</td>
<td>14:00</td>
<td>-7</td>
<td>17:00</td>
<td>-6</td>
</tr>
<tr>
<td>14:30</td>
<td>-3</td>
<td>17:30</td>
<td>-4</td>
<td>14:30</td>
<td>-8</td>
<td>17:30</td>
<td>-5</td>
</tr>
<tr>
<td>15:00</td>
<td>-3</td>
<td>18:00</td>
<td>-5</td>
<td>15:00</td>
<td>-10</td>
<td>18:00</td>
<td>-5</td>
</tr>
<tr>
<td>15:30</td>
<td>-3</td>
<td>18:30</td>
<td>-3</td>
<td>15:30</td>
<td>-8</td>
<td>18:30</td>
<td>-2</td>
</tr>
</tbody>
</table>
Announcement

Two tutorial sessions (same content):
• Thursday: 13:30 – 14:15 at SHB 801
• Friday: 14:30 – 15:15 at SHB 801

TA’s Office Hour:
• Thursday: 14:15 – 15:30 at SHB 801

Lecturer’s Office Hour:
• Tuesday 16:30 – 18:00 at SHB 717

Course codebase: https://github.com/metalbubble/ierg3050simulation
Outline

• Systems, models, and simulation
• Discrete-event simulation
• Steps in a sound simulation study

Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.8) [5th edition]
Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.9) [4th edition]

Acknowledgement to previous lecturers: Prof. Pascal Vontel, Prof. Minhua Chen, Rosana Chan, Prof. Angele Zhang, Prof. Jianwei Huang
Suppose you can redesign the schedules for CUHK school buses with all the existing conditions and resources, how?

Systems, models, and simulation

Ways to study a system
Analytical Solution: Pros and Cons

• Analytical solution: Use mathematical methods to get information on questions of interest
  • E.g. estimating total $$$ spent per month on buying magazines
  • E.g. characterizing average waiting time in a bank by queuing theory

• Advantages:
  • Clean answers in a cost-effective way
  • Can (relatively) easily explore how to optimize the performance

• Disadvantages:
  • Only work with tractable models (not applicable to most real systems)
  • Only explore the specific performance metric (we may be interested in many metrics simultaneously)
Simulation Solution: Pros and Cons

• Advantages
  • Simulation allows great flexibility in modeling complex systems, so simulation models can be highly valid
  • Easy to compare alternatives
  • Control experimental conditions

• Disadvantages
  • Stochastic simulations produce only estimates - with noise
  • Simulation models can be expensive to develop
  • Simulations usually produce large volumes of output - need to summarize, statistically analyze appropriately

• Pitfalls
  • Failure to identify objectives clearly up front
  • Inappropriate level of detail (both ways)
  • Inadequate design and analysis of simulation experiments
The Nature of Simulation

- Most real systems are too complex to study through analytic solution
- Systems described by complex models must be studied via simulation
  - Estimate the model characteristics
  - Run simulation and collect data
  - Analyze and evaluate the model numerically

Example: How to reduce the waiting time?
Applications of Simulation

• Designing and operating transportation systems such as airports, freeways, ports, and subways
• Evaluating designs for service organizations such as call centers, fast-food restaurants, hospitals, and post offices
• Reengineering business processes (inventory management, logistics, supply chain)
• Analyzing financial and economic systems
• Computer games
Systems, Models, and Simulation

• **System**: A collection of entities (people, parts, messages, machines, servers, ...) that act and interact together toward some goal

• In simulations, system and entities depend on objectives of study
  • Might limit the boundaries (physical and logical) of the system
  • Specify the level of detail (e.g., what is an entity?)
  • Usually assume a time element - *dynamic system*
Systems, Models, and Simulation

Examples of simulated systems
Systems, Models, and Simulation

• **State of a system**: Collection of variables and their values necessary to describe the system

• Might depend on desired objectives, output performance measures

• E.g., Website model: Number of client requests, number of server processes, network bandwidth etc.

• E.g., Game model: Weapons own by each characters, energy level etc.
Systems, Models, and Simulation

Types of systems

• **Discrete**
  • State variables change instantaneously at separated points in time
  • Bank model: State changes occur only when a customer arrives or departs

• **Continuous**
  • State variables change continuously as a function of time
  • Airplane flight: State variables like position, velocity change continuously

• Many systems are partly discrete, partly continuous
Systems, Models, and Simulation

• Classification of simulation models
  • Static vs. dynamic
  • Deterministic vs. stochastic (contain randomness)
  • Continuous vs. discrete

• Most operational models are dynamic, stochastic, and discrete
  • Called as discrete-event simulation models

• This course focuses on discrete-event simulations
Outline

• Systems, models, and simulation
• Discrete-event simulation
• Steps in a sound simulation study

Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.8) [5th edition]
Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.9) [4th edition]
Discrete-Event Simulation

- **Discrete-event simulation**: Modelling of a system as it evolves over time by a representation where the state variables change instantaneously at separated points in time
  - More precisely, state can change at only a countable number of points in time
  - These points in time are when **EVENTS** occur
- **Event**: Instantaneous occurrence that may change the state of the system
- Discrete-event simulation can in principle be done by hand, but usually done on computer
Discrete-Event Simulation

Entities

• They are objects that compose a simulation model
  • E.g. customers and servers
• Characterized by data values called *attributes*
• For each entity resident in the model there’s a record

Approaches to modelling

• *Event-scheduling* - coded in general-purpose languages (C, C++, Java, Pascal, Fortran)
• *Process* - focuses on entities and their “experience,” usually requires special-purpose simulation software
  • The simulation is actually executed behind the scenes in the event-scheduling logic
Example: single server queue

- **Objective:** to estimate expected *average delay in queue* (delay in line, not in service)

- **State variables**
  - Status of server (idle, busy) - needed to decide what to do with an arrival
  - Current length of the queue - to know where to store an arrival that must wait in line
  - Time of arrival of each customer now in queue - needed to compute time in queue when service starts
  - Servicing time for each particular customer

- **Events**
  - Arrival of a new customer
  - Service completion (and departure) of a customer
  - Maybe - end-simulation event (a “fake” event) - whether this is an event depends on how simulation terminates (a modeling decision)
Time-Advance Mechanisms

• *Simulation clock*: a variable that keeps the current value of (simulated) time in the model
  • Be careful to consistently use the same *time units*.
  • Usually no relation between simulated time and (real) time needed to run a model on a computer

• **Two approaches for time advance:**
  • *Next-event time advance* (usually used)
  • *Fixed-increment time advance* (seldom used, not taught in detail in this course)

  Simulation clock is advanced in increments of exactly \( \Delta t \) time units. After each update of the clock, a check is made to determine if any events should have occurred during the previous interval of length \( \Delta t \).
Time-Advance Mechanisms

The Next-event Time Advance Approach:

- Determine the times of occurrence of “future” events (keep an initial event list)
- Initialize simulation clock to 0
- Clock advances to the *most imminent* (first) of these future events
  - The state of the system is updated to account for the fact that an event has occurred.
  - Updating the system may involve updating event list.
- Then the simulation clock is advanced to the time of the new most imminent event, the state of the system is updated, and future event times are determined
  - next (most imminent) event, which is executed
  - Event execution may involve updating event list
  ...
- Continue until stopping rule is satisfied (must be explicitly stated)

**Note:** Clock “jumps” from one event time to the next, and doesn’t “exist” for times between successive events, periods of inactivity are ignored
Time-Advance Mechanisms:

- $T_i = \text{time of arrival of } i\text{th customer (}t_0 = 0\text{)}$
- $A_i = t_i - t_{i-1} = \text{interarrival time between } (i-1)\text{st and } i\text{th customers (usually assumed to be a random variable from some probability distribution)}$
- $S_i = \text{service-time requirement of } i\text{th customer (another random variable)}$
- $D_i = \text{delay in queue of } i\text{th customer}$
- $c_i = t_i + D_i + S_i = \text{time when } i\text{th customer completes service and departs}$
- $e_j = \text{time of occurrence of the } j\text{th event (of any type), } j = 1, 2, 3, \ldots$
Components of a Discrete Event Simulation

- **System state**: the collection of variables to describe the system at a particular time
- **Simulation clock**: a variable giving the current value of simulated time
- **Event list**: a list containing the next time when each type of event will occur
- **Statistical counters**: variables used for storing statistical information about system performance
Organization of a Discrete Event Simulation

- **Initialization routine**: a module to initialize the simulation model at time 0
- **Timing routine**: a module that determines the next event from the event list and then advances the simulation clock to the time when that event is to occur
- **Event routines**: to carry out logic for each event type. For each of the event type, there is a module to updates the system state when that particular type of event occurs

[continued on next page...]
Organization of a Discrete Event Simulation

[Continued from previous page]

• **Library routines**: utility routines / library (e.g. CSIM) to generate random variables according to the probability distributions

• **Report generator**: to summarize, report results at end

• **Main program**: ties routines together, executes them in right order. Also checks for termination
Flow Chart of a Discrete Event Simulation

Start

**Call Initialization Routine**
1. Set simulation Clock = 0
2. Initialize system states, statistical counters and event list

**Call Timing Routine**
1. Determine the next event type \( i \)
2. Advance the simulation clock

**Call Event \( i \) Routine**
1. Update system states and statistical counters
2. Generate future events and add to event list

**Library Routines:**
Generate random variates

Is STOPPING rule satisfied?

**Report Generator**
1. Compute estimates of interest
2. Write report

NO

YES

Stop

Acknowledgement: slide contributed by Ka Loi CHAN.
A short break:
Simulation of Collective Motion in Bacteria

https://github.com/metalbubble/ierg3050simulation/tree/master/SDP
Outline

• Systems, models, and simulation
• Discrete-event simulation
• Steps in a sound simulation study

Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.8) [5th edition]
Required reading: Chapter 1 (1.1, 1.2, 1.3, 1.7, 1.9) [4th edition]
Steps in a Sound Simulation Study

**Step 1:** Formulate the problem and plan the study.

**Step 2:** Collect information and data to define a model. Choose the level of model details.
  - Document the model.

**Step 3:** Check if the assumptions document is valid. If no, go back to Step 2.

**Step 4:** Construct a computer program and verify it.
  - Programming languages (required in IERG3050 project).
  - Simulation software.

**Step 5:** Make pilot runs.

[continued on next page . . . ]
Steps in a Sound Simulation Study

[. . . continued from previous page]

Step 6: Validate the programmed model.
- If there is an existing system, compare performance measures of the model with performance measures of the existing system.
- Use sensitivity analysis to determine what model factors have a significant impact on performance measures, and modify the model when necessary.

Step 7: Design experiments.
- Specify length of each simulation run and warm-up period.

Step 8: Make production runs and get the output data.

Step 9: Analysis output data.

Step 10: Document, present, and use results.
- Discuss model building and validation process to promote credibility.
Summary on the advantages, disadvantages, and pitfalls of simulation

Advantages:
• Simulation allows great flexibility in modelling complex systems, so simulation models can be highly valid.
• Easy to compare alternatives.
• Control experimental conditions.

Disadvantages:
• Stochastic simulations produce only estimates — with noise.
• Simulation models can be expensive to develop.
• Simulations usually produce large volumes of output — need to summarize, statistically analyse appropriately.

Pitfalls:
• Failure to identify objectives clearly up front.
• Inappropriate level of detail (both ways).
• Inadequate design and analysis of simulation experiments.
A walk through example of a simulation

A Single-Queue Single-Server System

We will describe it in terms of

• Problem statement
• Walk-through
• Paper-and-pencil simulation
• C program simulation
• Program organization and logic

Required reading: Chapter 1.4
Single-Queue Single-Server System

• A Queueing System consists of one or more servers that provide service to arriving customers
• A customer who arrive to find all servers busy join one in front of the server
• Single-Queue Single-Server system: a queueing system with only one server which has only one queue
• For a customer: $W = \text{total waiting time in the system}$, $D = \text{delay in queue}$, $S = \text{service time}$
Single-Queue Single-Server System

Interarrival times $A_1, A_2, ...$ are assumed to be independent and identically distributed (IID) random variables

- Independent: $\Pr(A_1 \cap \cdots \cap A_n) = \Pr(A_1) \cdots \Pr(A_n)$.
- Identically Distributed: have the same probability distribution

Examples of probability distribution:

- **Uniform**: for complete random variables
- **Normal**: the sum of a large number of (independent) random variables is approximately distributed according to a normal distribution. (as the central limit theorem.)
- **Exponential**: e.g., when the arrival times of the customers to a system are distributed according to a homogeneous Poisson process then the interarrival times are independent and are distributed according to an exponential distribution.

To be covered more in later parts of this course
Problem Statement for Single-Queue Single-Server System

• When a customer arrives
  • If the server is idle: enters service immediately
  • If the server is busy: wait at the end of the queue

• Service discipline: FIFO (first-in-first-out)

• Random variables:
  Interarrival times $A_1, A_2, \ldots$ are IID
  Service times $S_1, S_2, \ldots$ are IID
  The two sets variables are independent of each other
Problem Statement for Single-Queue Single-Server System (cont.)

Initial state: “empty-and-idle”
  • time t=0
  • no customers
  • idle server

Wait for the arrival of the first customer
  • occurs at the first interarrival time $A_1$

Stopping-rule: the n-th customer enters the service
  • A total of n customers have been in the queue.
  • The time at which the simulation ends is a random variable!
Performance Metrics

\(d(n)\): Expected average delay in queue of n-th customer
- excluding service time
- “Expected”: the average delay is a random variable

\(q(n)\): Expected average number of customers in queue (excluding any in service)

\(u(n)\): Expected utilization of the server
- proportion of time busy
- How busy the server is
Performance Metrics (graphs)

**Q(t):** the queue length as a function of time t

**B(t):** the “busy function”, as a function of time t
\[
B(t) \triangleq \begin{cases} 
1 & \text{if the server is busy at time } t \\
0 & \text{if the server is idle at time } t 
\end{cases}
\]

**T(n):** the time required to observe n delays in queue
- time elapsed when the n-th customers completes its delay in queue and start being served
- In our example, simulation will ends at T(6)
Paper-and-Pencil Simulation

Suppose virtual customers are generated with the following interarrival time and service time

\[ A_1 = 0.4, \quad S_1 = 2.0 \]
\[ A_2 = 1.2, \quad S_2 = 0.7 \]
\[ A_3 = 0.5, \quad S_3 = 0.2 \]
\[ A_4 = 1.7, \quad S_4 = 1.1 \]
\[ A_5 = 0.2, \quad S_5 = 3.7 \]
\[ A_6 = 1.6, \quad S_6 = 0.6 \]
\[ A_7 = 0.2, \quad A_8 = 1.4, \quad A_9 = 1.9 \]

In computer simulations, the Ai’s and Si’s would be generated from their corresponding probability distributions
Paper-and-Pencil Simulation

Observe and record the following values:

• Clock

• Event list
  • A: time for next arrival
  • D: time for next departure

• System state
  • Server status
  • Number in the queue
  • Arrival of each customer in queue, time of last event

• Statistical counters
  • Number delayed (=number of customer has ever been delayed, including being in service. Simulation ends when its value = 6)
  • Total delay (= \( \sum_{i}^{n} P_i \) where \( n \) = number delayed)
  • Area under \( Q(t) \)
  • Area under \( B(t) \)
Paper-and-Pencil Simulation

• Class Activity: Run a “Paper-and Pencil” simulation

Interarrival times:  0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times:     2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

Before Starting the Simulation, in each step check which values
• have changed
• are always consistent
• and why?
• also, which values help you to predict the next step?
Paper-and-Pencil Simulation

• Initialization

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

• Event 1: arrival of Customer 1

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...

\[ D_1 = \text{period of time that customer 1 has been in the queue} = 0 \]
Total delay = \( D_1 = 0 \)
Paper-and-Pencil Simulation

• Event 2: arrival of Customer 2

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

• Event 3: arrival of Customer 3

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

• Event 4: departure of Customer 1

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...

\( D_2 = \) period of time that customer 2 has been in the queue = 2.4-1.6
Total delay = \( D_1 + D_2 = 0 + 0.8 = 0.8 \)
Paper-and-Pencil Simulation

• Event 5: departure of Customer 2

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...

\[ D_3 = 3.1 - 2.1 = 1.0 \]
Total delay = \( D_1 + D_2 + D_3 = 0 + 0.8 + 1.0 = 1.8 \)
Paper-and-Pencil Simulation

- Event 6: departure of Customer 3

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

• Event 7: arrival of Customer 4

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...

\[ D_4 = 0 \]

Total delay = \( D_1 + D_2 + D_3 + D_4 = 0 + 0.8 + 1.0 + 0 = 1.8 \)
Paper-and-Pencil Simulation

• Event 8: arrival of Customer 5

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

- Event 9: departure of Customer 4

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...

$D_5 = 0.9$
Total delay $= D_1 + D_2 + D_3 + D_4 + D_5 = 0 + 0.8 + 1.0 + 0 + 0.9 = 2.7$
• Event 10: arrival of Customer 6

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

• Event 11: arrival of Customer 7

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
**Paper-and-Pencil Simulation**

- **Event 12**: arrival of Customer 8

<table>
<thead>
<tr>
<th>Event: $e_{13}$</th>
<th>System state</th>
<th>Statistical Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1 2</td>
<td>8.6 A 9.1 D 9.2</td>
</tr>
<tr>
<td>7 8</td>
<td>5.6 5.8 7.2</td>
<td>Clock 5.7 5.7 6.3</td>
</tr>
<tr>
<td>System</td>
<td>Server status Number in queue Times of last event Time of last event</td>
<td>Number of Delayed Total Delay Area under $Q(t)$ under $B(t)$</td>
</tr>
<tr>
<td>6</td>
<td>2 2</td>
<td>6 5.7 5.7 6.3</td>
</tr>
</tbody>
</table>

**Interarrival times**: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...

**Service times**: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...
Paper-and-Pencil Simulation

- Event 13: departure of Customer 5, Customer 6 end queueing, we have collected 6 piece of data! Quit!

<table>
<thead>
<tr>
<th>Event: $e_{13}$</th>
<th>System state</th>
<th>System representation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>System</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Interarrival times: 0.4, 1.2, 0.5, 1.7, 0.2, 1.6, 0.2, 1.4, 1.9, ...
Service times: 2.0, 0.7, 0.2, 1.1, 3.7, 0.6, ...

$D_6 = 3$
Total delay = $D_1 + D_2 + D_3 + D_4 + D_5 + D_6 = 0 + 0.8 + 1.0 + 0 + 0.9 + 3 = 5.7$
## Paper-and-Pencil Simulation

### Simulation Record

<table>
<thead>
<tr>
<th>Event No.</th>
<th>time</th>
<th>Description</th>
<th>Server status</th>
<th>No. in queue</th>
<th>Times of arrival</th>
<th>Time of last event</th>
<th>No. delayed</th>
<th>Total delay</th>
<th>Area under Q(t)</th>
<th>Area under B(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>Arrival 1</td>
<td>0</td>
<td>0</td>
<td>---</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
<td>Arrival 2</td>
<td>1</td>
<td>1</td>
<td>1.6</td>
<td>1.6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>Arrival 3</td>
<td>1</td>
<td>2</td>
<td>1.6, 2.1</td>
<td>2.1</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
<td>1.7</td>
</tr>
<tr>
<td>4</td>
<td>2.4</td>
<td>Departure 1</td>
<td>1</td>
<td>1</td>
<td>2.1</td>
<td>2.4</td>
<td>2</td>
<td>0.8</td>
<td>0.5</td>
<td>2.0</td>
</tr>
<tr>
<td>5</td>
<td>3.1</td>
<td>Departure 2</td>
<td>1</td>
<td>0</td>
<td>---</td>
<td>3.1</td>
<td>3</td>
<td>1.8</td>
<td>1.1</td>
<td>2.7</td>
</tr>
<tr>
<td>6</td>
<td>3.3</td>
<td>Departure 3</td>
<td>0</td>
<td>0</td>
<td>---</td>
<td>3.3</td>
<td>3</td>
<td>1.8</td>
<td>1.8</td>
<td>2.9</td>
</tr>
<tr>
<td>7</td>
<td>3.8</td>
<td>Arrival 4</td>
<td>1</td>
<td>0</td>
<td>---</td>
<td>3.8</td>
<td>4</td>
<td>1.8</td>
<td>1.8</td>
<td>2.9</td>
</tr>
<tr>
<td>8</td>
<td>4.0</td>
<td>Arrival 5</td>
<td>1</td>
<td>1</td>
<td>4.0</td>
<td>4.0</td>
<td>4</td>
<td>1.8</td>
<td>1.8</td>
<td>3.1</td>
</tr>
<tr>
<td>9</td>
<td>4.9</td>
<td>Departure 4</td>
<td>1</td>
<td>0</td>
<td>---</td>
<td>4.9</td>
<td>5</td>
<td>2.7</td>
<td>2.7</td>
<td>4.0</td>
</tr>
<tr>
<td>10</td>
<td>5.6</td>
<td>Arrival 6</td>
<td>1</td>
<td>1</td>
<td>5.6</td>
<td>5.6</td>
<td>5</td>
<td>2.7</td>
<td>2.7</td>
<td>4.7</td>
</tr>
<tr>
<td>11</td>
<td>5.8</td>
<td>Arrival 7</td>
<td>1</td>
<td>2</td>
<td>5.6, 5.8</td>
<td>5.8</td>
<td>5</td>
<td>2.7</td>
<td>2.9</td>
<td>4.9</td>
</tr>
<tr>
<td>12</td>
<td>7.2</td>
<td>Arrival 8</td>
<td>1</td>
<td>3</td>
<td>5.6, 5.8, 7.2</td>
<td>7.2</td>
<td>5</td>
<td>2.7</td>
<td>5.7</td>
<td>6.3</td>
</tr>
<tr>
<td>13</td>
<td>8.6</td>
<td>Departure 5</td>
<td>1</td>
<td>2</td>
<td>5.8, 7.2</td>
<td>8.6</td>
<td>6</td>
<td>5.7</td>
<td>9.9</td>
<td>7.7</td>
</tr>
</tbody>
</table>
Simulation Result

- Plot of $Q(t)$: queue length
Simulation Result

• Plot of $B(t)$: server business
Average Delay in Queue

A single simulation: an estimated value of average delay in queue

• Average of the $n$ Di’s:

$$\hat{d}(n) = \frac{1}{n} \sum_{i=1}^{n} D_i = \bar{D}(n)$$

• If we run simulations many times, then we will obtain the expected average delay in queue, $d(n)$
Time-Average Queue Length

• Obtain the *expected average queue length* \( q(n) \) by running simulation many times

• Let \( Q(t) \) denote the number of customers in queue at time \( t \)

• Let \( p_i \) be the *expected proportion of the time* that \( Q(t) = i \), we have

\[
q(n) = \sum_{i=0}^{\infty} ip_i
\]

• Thus \( q(n) \) is a *weighted average* of the queue length \( Q(t) \)
Time-Average Queue Length

Single run of simulation: estimated values for pi’s

• Replace the pi’s with the estimated (not expected) proportions
  \[ \hat{p}_i = \frac{\sum_{i=1}^{\infty} i\hat{r}_i}{T(n)} \]

• How to compute \( \hat{p}_i \)
  • \( T_i \): total amount of time the queue is of length i,
  • \( T(n) = T_0 + T_1 + \ldots + T_n \)

\[ q(n) = \sum_{i=1}^{\infty} i\hat{p}_i \]
Time-Average Queue Length

\[ T_0 = (1.6 - 0.0) + (4.0 - 3.1) + (5.6 - 4.9) = 3.2 \]
\[ T_1 = (2.1 - 1.6) + (3.1 - 2.4) + (4.9 - 4.0) + (5.8 - 5.6) = 2.3 \]
\[ T_2 = (2.4 - 2.1) + (7.2 - 5.8) = 1.7 \]
\[ T_3 = 8.6 - 7.2 = 1.4 \]
\[ T_i = 0 \text{ for } i \geq 4 \]

\[ \sum_{i=0}^{\infty} iT_i = (0 \times 3.2) + (1 \times 2.3) + (2 \times 1.7) + (3 \times 1.4) = 9.9 \]

\[ \therefore \hat{q}(6) = \frac{9.9}{8.6} = 1.15 \]

An equivalent expression:

\[ \sum_{i=0}^{\infty} iT_i = \int_0^{T(n)} Q(t) \, dt \]

and \[ \hat{q}(n) = \frac{\int_0^{T(n)} Q(t) \, dt}{T(n)} \]
Utilization

• Expected utilization: how busy the server is
  • the proportion of time that the server is busy
  • A value between 0 and 1, denoted by $u(n)$

• Single simulation: *observed* utilization

\[ \hat{u}(n) = \frac{\int_0^{T(n)} B(t)dt}{T(n)} \]

and $\hat{u}(n) = \frac{(3.3 - 0.4) + (8.6 - 3.8)}{8.6} = \frac{7.7}{8.6} = 0.90$
Paper-and-Pencil Simulation: Summary

Events include:
  • the arrival of a customer
  • the departure of a customer (after a service completion)

State variables to estimate \( d(n) \), \( q(n) \), and \( u(n) \) are
  • The status of the server (0 for idle, 1 for busy)
  • The number of customers in the queue
  • The time of the last (most recent) event
    • Needed to compute the width of the rectangles for the area accumulations in the estimates of \( q(n) \) and \( u(n) \)
Simulation: Program Organization and Logic

• We can write a computer program to simulate the same system

• The simulator should includes routines for
  • The simulation will end when \( n = 1000 \) delays in queue have been completed (rather than \( n = 6 \))
  • The inter-arrival time is modeled as independent random variables from exponential distributions with mean 1 minutes
  • The service time is modeled as independent random variables from exponential distributions with mean 0.5 minutes

• Note: the exponential distribution PDF with mean \( \beta > 0 \)

\[
f(x) = \frac{1}{\beta} e^{-x/\beta} \quad \text{for} \quad x \geq 0
\]
Simulation: Program Organization and Logic

• Computer program should be modularized for better logic and reuse
• Different from the paper-and-pencil simulation in
  • Initialization
  • Timing
  • Report generation, and
  • Generating exponential random variables
  • Events

<table>
<thead>
<tr>
<th>Event description</th>
<th>Event type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival of a customer to the system</td>
<td>1</td>
</tr>
<tr>
<td>Departure of a customer from the system after completing service</td>
<td>2</td>
</tr>
</tbody>
</table>
Main Program Flow

1. Set simulation clock = 0
2. Initialize system state and statistical counters
3. Initialize event list

0. Invoke the initialization routine

1. Invoke the timing routine
2. Invoke event routine i

Repeatedly

Event routine i

1. Update system state
2. Update statistical counters
3. Generate future events and add to event list

Is simulation over?

No

Yes

Report generator

1. Compute estimates of interest
2. Write report

Stop

Timing routine

1. Determine the next event type, say i
2. Advance the simulation clock

Library routines

Generate random variates
Flowchart for arrival routine

Arrival event

Schedule the next arrival event

Is the server busy?

Yes

Add 1 to the number in queue

Yes

Write error message and stop simulation

No

Is the queue full?

Yes

Store time of arrival of this customer

No

Set delay = 0 for this customer and gather statistics

Add 1 to the number of customers delayed

Make the server busy

Schedule a departure event for this customer

Return

Flowchart for departure routine

Departure event

Is the queue empty?

Yes

Make the server idle

No

Eliminate departure event from consideration

Subtract 1 from the number in queue

Compute delay of customer entering service and gather statistics

Add 1 to the number of customers delayed

Schedule a departure event for this customer

Move each customer in queue (if any) up one place

Return
Simulation Sample Output

• An example of the simulation output data

Mean interarrival time    1.000 minute
Mean service time         0.500 minute
Number of customers       1000
Average delay in queue    0.430 minute
Average number in queue   0.418
Server utilization        0.460
Time simulation ended     1027.915 minute
End: Thank you!

• Next Week: Review of Basic Probability and Statistics
• Required reading this week: Chapter 4 of the Textbook
• Updated course outline and slide will be uploaded to Piazza and course page