Lecture 5: Introduction to Discrete Event Simulation

END 322E System Simulation

Mehmet Ali Ergün, Ph.D.

Introduction

- We will develop a common framework for the modeling of complex systems using discrete-event simulation (DES)
- Discrete-Event simulation is appropriate for systems that changes in system state occur only at discrete points in time
- We will cover basic building blocks all simulation models
- Fundamental concepts we cover in this lecture are not tied to any particular simulation software
- We consider exclusively dynamic, stochastic systems (involving time and containing random elements) which change occur in a discrete manner

Concepts in DES

- System: A collection of entities (e.g., people and machines) that interact together to accomplish one or more goals
- Model: An abstract representation of a system, usually containing structural, logical, or mathematical relationships that describes a system in terms of
    - State
    - Entities and their attributes
    - Sets
    - Processes
    - Events
    - Activities
    - Delays

Concepts in DES

- State: A collection of variables that contain all the information necessary to describe a system at a point in time
- Entity: Any object or component in the system which requires explicit representation in the model (e.g., a server, a customer, a machine)
- Attributes: The properties of a given entity (e.g., the priority of a given customer, the routing of a job through job-shop)
- List: A collection of associated entities, ordered in some logical fashion (e.g., all customers currently waiting in line, FIFO or by priority fashion)
- Event: An instantaneous occurrence that change the system state (e.g., arrival or departure of a customer)

Concepts in DES

- Event Notice: Record of an event to occur at the current or some future time, with any associated data necessary to execute the event (at a minimum this includes event and the event time)
- Future Event List: A list of event notices for future events, ordered by time of occurrence also called future event list
- Activity: A duration of time of specified length (e.g., a service time or interarrival time)
- Delay: A duration of time of unspecified indefinite length, which is not known until it ends (e.g., a customer's waiting time in line depends on departure time of previous customers)
- Clock: A variable representing simulated time

Concepts in DES

- Different simulation packages use different terminology for the same or similar concepts
- An activity typically represents a service time, an interarrival time, or any other processing time, whose duration is characterized by the modeler. It can be specified as
  - Deterministic, e.g., always exactly 5 minutes
  - Statistical, e.g., as a random draw from among 2,5,7 with equal probabilities
  - A function depending on system variables or entities, e.g., loading time for an iron ore ship as a function of the ship's cargo weight and loading rate in tons per hour

Concepts in DES

- Activity can be characterized by from its specifications
  - If the clock is 100 and an inspection that will take 5 minutes is starting then an event notice can be created at 105 as end of an inspection
- In contrast to activity, a Delay's duration is not specified by the modeler ahead of time but rather is determined by system conditions.
- Delay ends when some logical conditions are met and typically is an output of the simulation.
  - A customer's waiting time in line depends on the number and duration of service of the customers ahead in line as well as the availability of servers and equipment
- A delay is also called a "conditional wait", while an activity is called an "unconditional wait"

Event Scheduling Algorithm

- A discrete event simulation proceeds by producing a sequence of system snapshots which represents the evolution of system through time
- A system snapshot at time t, includes
  - The system state at time t,
  - List of all activities (future event list) currently in progress and when such an event will end.
  - The status of all current entities
  - The current values of cumulative statistics
- Not all models will contain every element

Event Scheduling Algorithm

CLOCK	System state	Entities and attributes	Set 1	Set 2	. . .	Future event list (FEL)	Cumulative statistics and counters
t	$(x, y, z, \ldots)$					$(3, t_1)$ – Type 3 event to occur at time t_1 $(1, t_2)$ – Type 1 event to occur at time t_2	
	. . .						

Figure 1 Prototype system snapshot at simulation time t.

Event Scheduling Algorithm

- The mechanism for advancing the simulation time is based on Future Event List (FEL)
- List contains all event notices for events that have been scheduled to occur at a future time
- Events are ordered chronologically at the FEL
- The clock is advanced to the imminent event time at the FEL and the event is executed to change system state
- If a new event notice needs to be created and added to the FEL, the event notices may need to be reordered

Event Scheduling Algorithm

Old system snapshot at time t

CLOCK	System state	...	Future event list
t	(5, 1, 6)		$(3, t_1)$– Type 3 event to occur at time t_1 $(1, t_2)$– Type 1 event to occur at time t_2 $(1, t_3)$– Type 1 event to occur at time t_3 ⋮ $(2, t_n)$– Type 2 event to occur at time t_n

New system snapshot at time t_1

CLOCK	System state	...	Future event list
t_1	(5, 1, 5)		$(1, t_2)$– Type 1 event to occur at time t_2 $(4, t^*)$– Type 4 event to occur at time t^* $(1, t_3)$– Type 1 event to occur at time t_3 ⋮ $(2, t_n)$– Type 2 event to occur at time t_n

Event-scheduling/time-advance algorithm

Step 1. Remove the event notice for the imminent event (event 3, time t_1) from FEL.

Step 2. Advance CLOCK to imminent event time (i.e., advance CLOCK from t to t_1).

Step 3. Execute imminent event: update system state, change entity attributes, and set membership as needed.

Step 4. Generate future events (if necessary) and place their event notices on FEL, ranked by event time. (Example: Event 4 to occur at time t^*, where $t_2 < t^* < t_3$.)

Step 5. Update cumulative statistics and counters.

Grocery Store Example Revisited

- System State:
  - $L_Q(t)$=Number of customers waiting in line at time t
  - $L_S(t)$= Number of customers being served at time t (0 or 1)
- Entities:
  - Customers and servers
- Events
  - Arrival
  - Departure
- Event Notices
  - (A, C_i, t), representing an arrival event to occur at a future time t by customer i
  - (D, C_i, t), representing a departure event to occur at a future time t by customer i
- Activities
  - Interarrival time defined in lecture 4
  - Service time defined in lecture 4
- Delay
  - Customer time spent waiting in line

Grocery Store Example Revisited

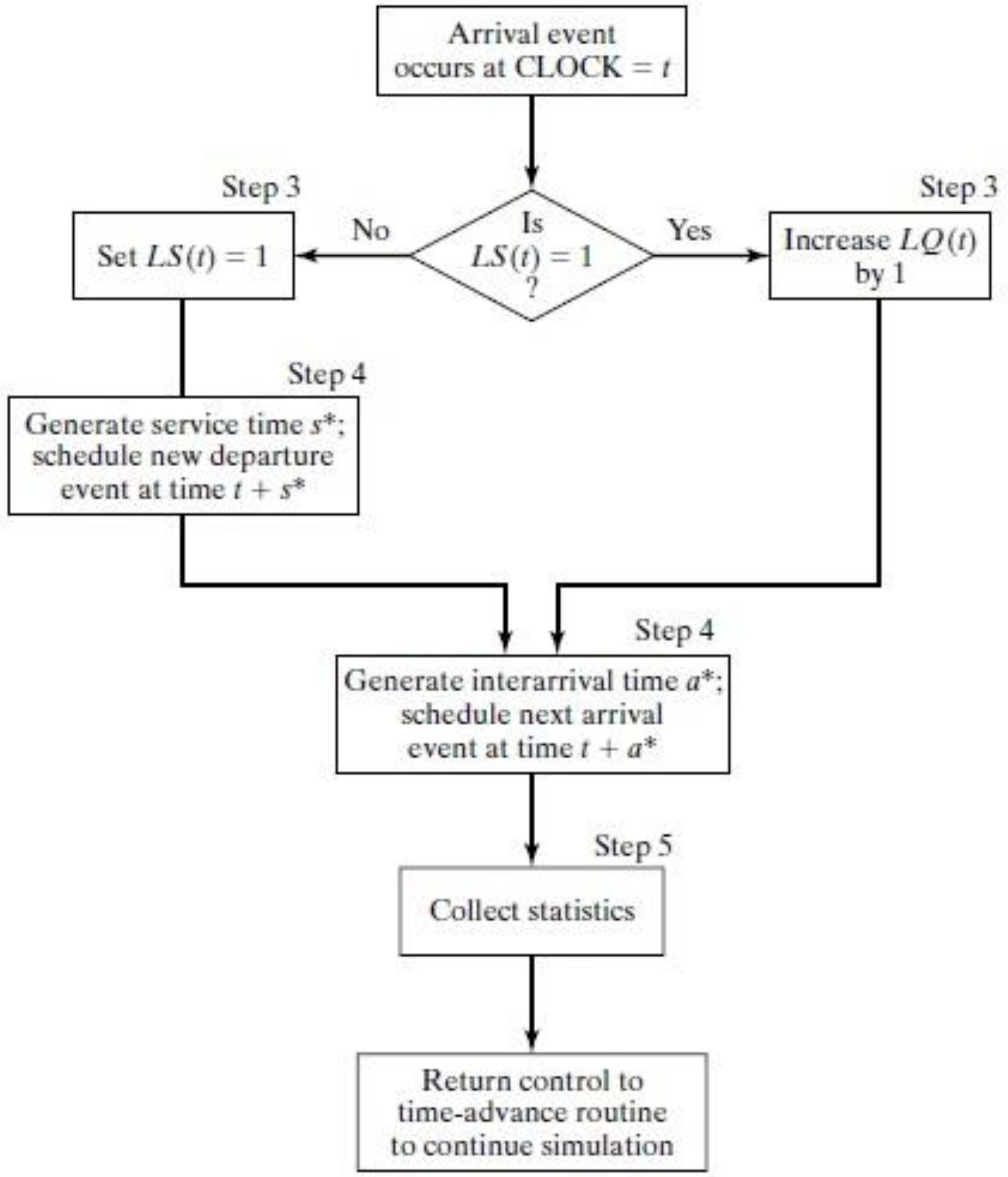


Figure 5 Execution of the arrival event.

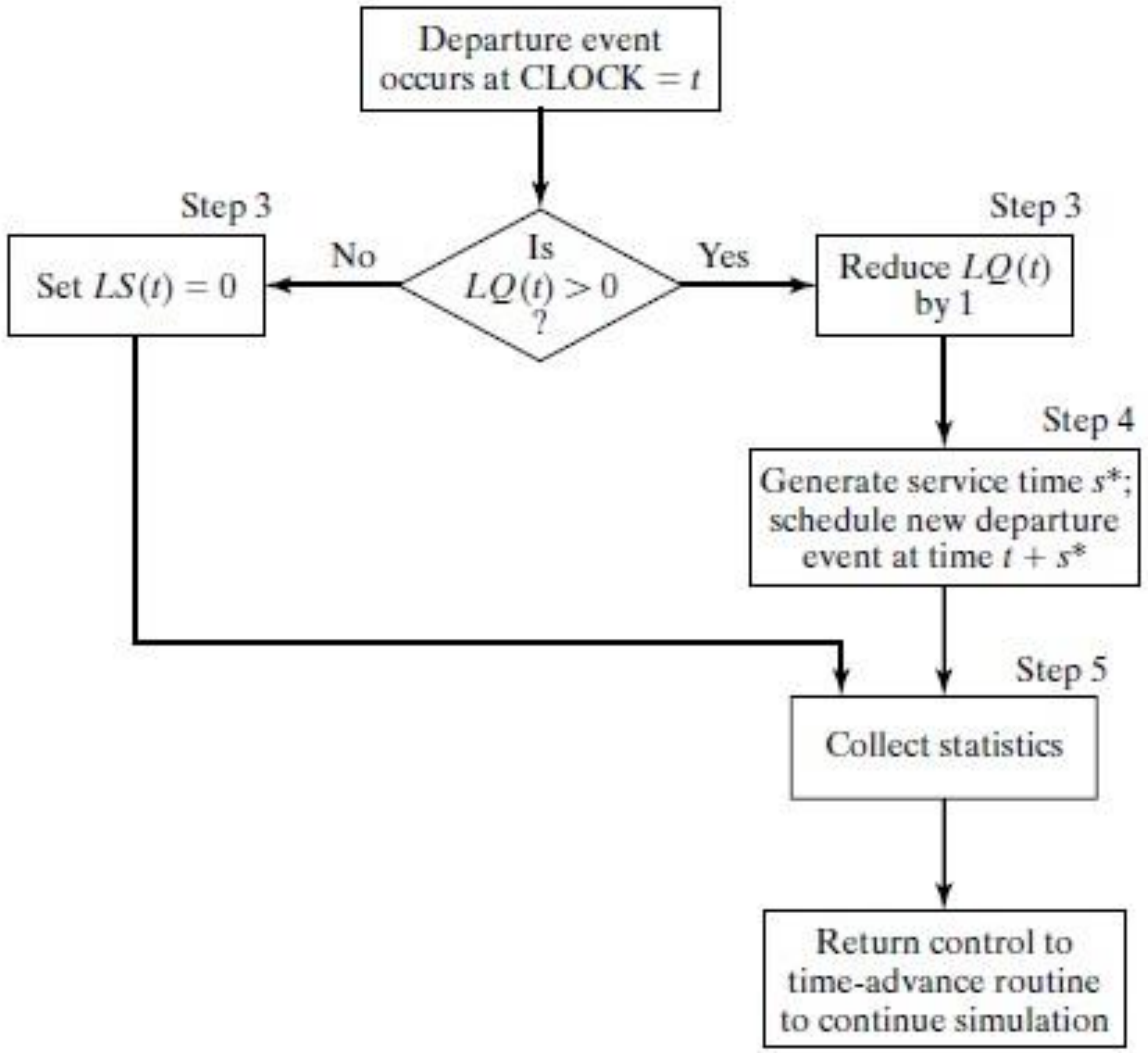


Figure 6 Execution of the departure event.

Grocery Store Example Revisited

- Assume we only want to know maximum queue length and server utilization

System State

Cumulative Statistics

Clock (t)	$L_Q(t)$	$L_S(t)$	Future Event List	Cumulative Server Busy Time	Maximum Queue Length
0	**0**	**1**	**(A, C2, 1) , (D,C1,4)**	**0**	**0**

Grocery Store Example Revisited

- Arrival event executed: system state updated next arrival scheduled and added to event list (FEL always chronologically ordered)

System State

Cumulative Statistics

Clock (t)	$L_Q(t)$	$L_S(t)$	Future Event List	Cumulative Server Busy Time	Maximum Queue Length
0	0	1	(A, C2, 1) , (D,C1,4)	0	0
1	1	1	(A, C3, 2) , (D,C1,4)	1	1

Grocery Store Example Revisited

- Arrival event executed: state updated, new arrival is scheduled and added to the FEL

System State

Cumulative Statistics

Clock (t)	$L_Q(t)$	$L_S(t)$	Future Event List	Cumulative Server Busy Time	Maximum Queue Length
0	0	1	(A, C2, 1) , (D,C1,4)	0	0
1	1	1	(A, C3, 2) , (D,C1,4)	1	1
2	2	1	(D , C1, 4) , (A, C4, 8)	2	2

Grocery Store Example Revisited

- Departure executed: state updated, new service time assigned hence new departure event added to FEL

System State

Cumulative Statistics

Clock (t)	$L_Q(t)$	$L_S(t)$	Future Event List	Cumulative Server Busy Time	Maximum Queue Length
0	0	1	(A, C2, 1) , (D,C1,4)	0	0
1	1	1	(A, C3, 2) , (D,C1,4)	1	1
2	2	1	(D , C1, 4) , (A, C4, 8)	2	2
4	1	1	(D, C2, 6) , (A, C4, 8)	4	2

Grocery Store Example Revisited

- Departure event executed: state updated, new service time assigned and new departure event added to the FEL

System State

Cumulative Statistics

Clock (t)	$L_Q(t)$	$L_S(t)$	Future Event List	Cumulative Server Busy Time	Maximum Queue Length
0	0	1	(A, C2, 1) , (D,C1,4)	0	0
1	1	1	(A, C3, 2) , (D,C1,4)	1	1
2	2	1	(D , C1, 4) , (A, C4, 8)	2	2
4	1	1	(D, C2, 6) , (A, C4, 8)	4	2
6	0	1	(A, C4, 8) , (D, C3, 11)	6	2

Grocery Store Example Revisited

- Arrival event executed: state updated (customer joined the queue) a new arrival is scheduled and added to the FEL

System State

Cumulative Statistics

Clock (t)	$L_Q(t)$	$L_S(t)$	Future Event List	Cumulative Server Busy Time	Maximum Queue Length
0	0	1	(A, C2, 1) , (D,C1,4)	0	0
1	1	1	(A, C3, 2) , (D,C1,4)	1	1
2	2	1	(D , C1, 4) , (A, C4, 8)	2	2
4	1	1	(D, C2, 6) , (A, C4, 8)	4	2
6	0	1	(A, C4, 8) , (D, C3, 11)	6	2
8	1	1	(D, C3, 11) , (A, C5, 11)	8	2

Grocery Store Example Revisited

- Departure and an arrival executed: state updated, new service time assigned hence new departure event added to FEL, an arrival scheduled and added to the FEL

Clock (t)	$L_Q(t)$	$L_S(t)$	Future Event List	Cumulative Server Busy Time	Maximum Queue Length
0	0	1	(A, C2, 1) , (D,C1,4)	0	0
1	1	1	(A, C3, 2) , (D,C1,4)	1	1
2	2	1	(D , C1, 4) , (A, C4, 8)	2	2
4	1	1	(D, C2, 6) , (A, C4, 8)	4	2
6	0	1	(A, C4, 8) , (D, C3, 11)	6	2
8	1	1	(D, C3, 11) , (A, C5, 11)	8	2
11	1	1	(D, C4, 15) , (A, C6, 18)	11	2

Able-Baker Call Center Problem

- Consider a computer technical support where personnel take calls and provide service. Time between calls ranges from 1 to 4 minutes with distribution given below

Inter-arrival Time	Probability	Cumulative Probability	Random Number Interval
1	0.25	0.25	(0 , 0.25]
2	0.40	0.65	(0.25 , 0.65]
3	0.20	0.85	(0.65 , 0.85]
4	0.15	1	(0.85 , 1]

Able-Baker Call Center Problem

- There are two technical support people – Able and Baker. Able is more experienced and can provide service faster than Baker. The distributions of their service times are given as below. Able gets the call if both are idle. One customer in line at time 0. Collect utilization statistics of Able and Baker via simulation

Table 2.12 Service Distribution of Able

Service Time (Minutes)	*Probability*	*Cumulative Probability*	*Random-Digit Assignment*
2	0.30	0.30	01–30
3	0.28	0.58	31–58
4	0.25	0.83	59–83
5	0.17	1.00	84–00

Table 2.13 Service Distribution of Baker

Service Time (Minutes)	*Probability*	*Cumulative Probability*	*Random-Digit Assignment*
3	0.35	0.35	01–35
4	0.25	0.60	36–60
5	0.20	0.80	61–80
6	0.20	1.00	81–00

Able-Baker Call Center Problem

- System State:
  - $L_Q(t)$= Number of customers waiting in line at time t
  - $L_A(t)$= Indicates whether Able is idle (0) or busy (1)
  - $L_B(t)$= Indicates whether Baker is idle (0) or busy (1)
- Entities:
  - Customers and Able and Baker
- Events
  - Arrival, Service completion by Able, Service Completion by Baker
- Event Notices
  - (A, C_i, t), representing an arrival event to occur at a future time t
  - (S_a, C_i, t), representing a service completion by Able at a future time t
  - (S_b, C_i, t), representing a service completion by Baker at a future time t
- Activities
  - Interarrival time, Service time by Able, Service time by Baker
- Delay
  - Customer time spent waiting in line

Able-Baker Call Center Problem

- Next arrival scheduled and added to FEL, service completion of the first call by Able scheduled (using Able's service distribution) and added to FEL

System State					Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , **5**)	0	0

Able-Baker Call Center Problem

- System state updated (call got by Baker, Able still on call, no one in line). Next arrival scheduled (4 mins) and added to FEL, service completion of the second call by Baker scheduled (using Baker's service distribution) and added to FEL (service time generated is 3 mins)

System State					Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , **5**)	0	0

Able-Baker Call Center Problem

- Two calls are completed, system state updated, nobody in line, no events to add to FEL

System State					Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , **5**)	0	0
2	**0**	**1**	**1**	**(S_a,C1, 5), (S_b,C2 ,5), (A,C3,6)**	**2**	**0**

Able-Baker Call Center Problem

- A call arrived Able got it, a service completion event (a service time drawn from Able's distribution (5 minutes)) is added to the FEL, a new arrival event is scheduled and added to FEL (4 minutes)

System State					Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , 5)	0	0
2	**0**	**1**	**1**	(S_a,C1,5), (S_b,C2 ,5), (A,C3,6)	**2**	**0**
5	**0**	**0**	**0**	(A,C3,6)	**5**	**3**

Able-Baker Call Center Problem

- A call arrived Baker got it, a service completion event (a service time drawn from Baker's distribution (6 minutes)) is added to the FEL, a new arrival event is scheduled and added to FEL (2 minutes)

	System State				Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , 5)	0	0
2	**0**	**1**	**1**	(S_a,C1,5), (S_b,C2 ,5), (A,C3,6)	**2**	**0**
5	**0**	**0**	**0**	(A,C3,6)	**5**	**3**
6	**0**	**1**	**0**	(A,C4 , 10), (S_a,C3,11)	**5**	**3**

Able-Baker Call Center Problem

- A call is completed by Able, no one in line system state is updated

	System State				Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , 5)	0	0
2	**0**	**1**	**1**	(S_a,C1,5), (S_b,C2 ,5), (A,C3,6)	**2**	**0**
5	**0**	**0**	**0**	(A,C3,6)	**5**	**3**
6	**0**	**1**	**0**	(A,C4 , 10), (S_a,C3,11)	**5**	**3**
10	**0**	**1**	**1**	(S_a, C3, 11), (A, C5, 12), (S_b, C4, 16)	**9**	**3**

Able-Baker Call Center Problem

- A call has arrived and got by Able. A service completion by Able event added to the FEL (5 min) , a new arrival event is added to the FEL (an interarrival drawn from interarrival distribution 2 min)

	System State				Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , 5)	0	0
2	**0**	**1**	**1**	(S_a,C1,5), (S_b,C2 ,5), (A,C3,6)	**2**	**0**
5	**0**	**0**	**0**	(A,C3,6)	**5**	**3**
6	**0**	**1**	**0**	(A,C4 , 10), (S_a,C3,11)	**5**	**3**
10	**0**	**1**	**1**	(S_a, C3, 11), (A, C5, 12), (S_b, C4, 16)	**9**	**3**
11	**0**	**0**	**1**	(A, C5, 12), (S_b, C4, 16)	**10**	**4**

Able-Baker Call Center Problem

- A call has arrived Able and Baker is busy, hence started waiting in line. New arrival scheduled (and added to the FEL 4 min)

System State					Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , 5)	0	0
2	**0**	**1**	**1**	(S_a,C1,5), (S_b,C2 ,5), (A,C3,6)	**2**	**0**
5	**0**	**0**	**0**	(A,C3,6)	**5**	**3**
6	**0**	**1**	**0**	(A,C4 , 10), (S_a,C3,11)	**5**	**3**
10	**0**	**1**	**1**	(S_a, C3, 11), (A, C5, 12), (S_b, C4, 16)	**9**	**3**
11	**0**	**0**	**1**	(A, C5, 12), (S_b, C4, 16)	**10**	**4**
12	**0**	**1**	**1**	(A, C6, 14), (S_b, C4, 16), (S_a, C5, 17)	**10**	**5**

Able-Baker Call Center Problem

- Utilization of Able= 12/14
- Utilization of Baker=7/14
- Of course needs a way longer simulation to obtain reasonable statistics

	System State				Cumulative Statistics	
Clock (t)	$L_Q(t)$	$L_A(t)$	$L_B(t)$	Future Event List	Able Busy Time	Baker Busy Time
0	0	1	0	(A, C2 , 2), (S_a, C1 , 5)	0	0
2	**0**	**1**	**1**	(S_a,C1,5), (S_b,C2 ,5), (A,C3,6)	**2**	**0**
5	**0**	**0**	**0**	(A,C3,6)	**5**	**3**
6	**0**	**1**	**0**	(A,C4 , 10), (S_a,C3,11)	**5**	**3**
10	**0**	**1**	**1**	(S_a, C3, 11), (A, C5, 12), (S_b, C4, 16)	**9**	**3**
11	**0**	**0**	**1**	(A, C5, 12), (S_b, C4, 16)	**10**	**4**
12	**0**	**1**	**1**	(A, C6, 14), (S_b, C4, 16), (S_a, C5, 17)	**10**	**5**
14	**1**	**1**	**1**	(S_b, C4, 16), (S_a, C5, 17), (A, C7, 18)	**12**	**7**

Lecture 6: Monte Carlo Simulation and Other Simulation Examples

END 322E System Simulation

Mehmet Ali Ergün, Ph.D.

The Manhattan Project

- One of the secret research projects in the USA during World War II
- Contributions to World Heritage:
  - Atomic Bomb
  - Half of the Marvel Characters
- Stansilaw Ulam
  - Polish-American Scientist
  - Loves solitaire
  - Good friends with John von Neumann
  - Ground work for Monte-Carlo Methods

Monte-Carlo Simulation

- Ulam wanted to find a winning strategy
  - Find strategies that works and pinpoint the common aspects
  - Playing over and over again- takes a lifetime
  - Used ENIAC (Electronic Numerical Integrator And Computer) to run multiple scenarios
  - Later, they also use this method in designing the hydrogen bomb
- Monte Carlo: A city in Monaco
  - Famous for its casinos

Monte-Carlo Simulation

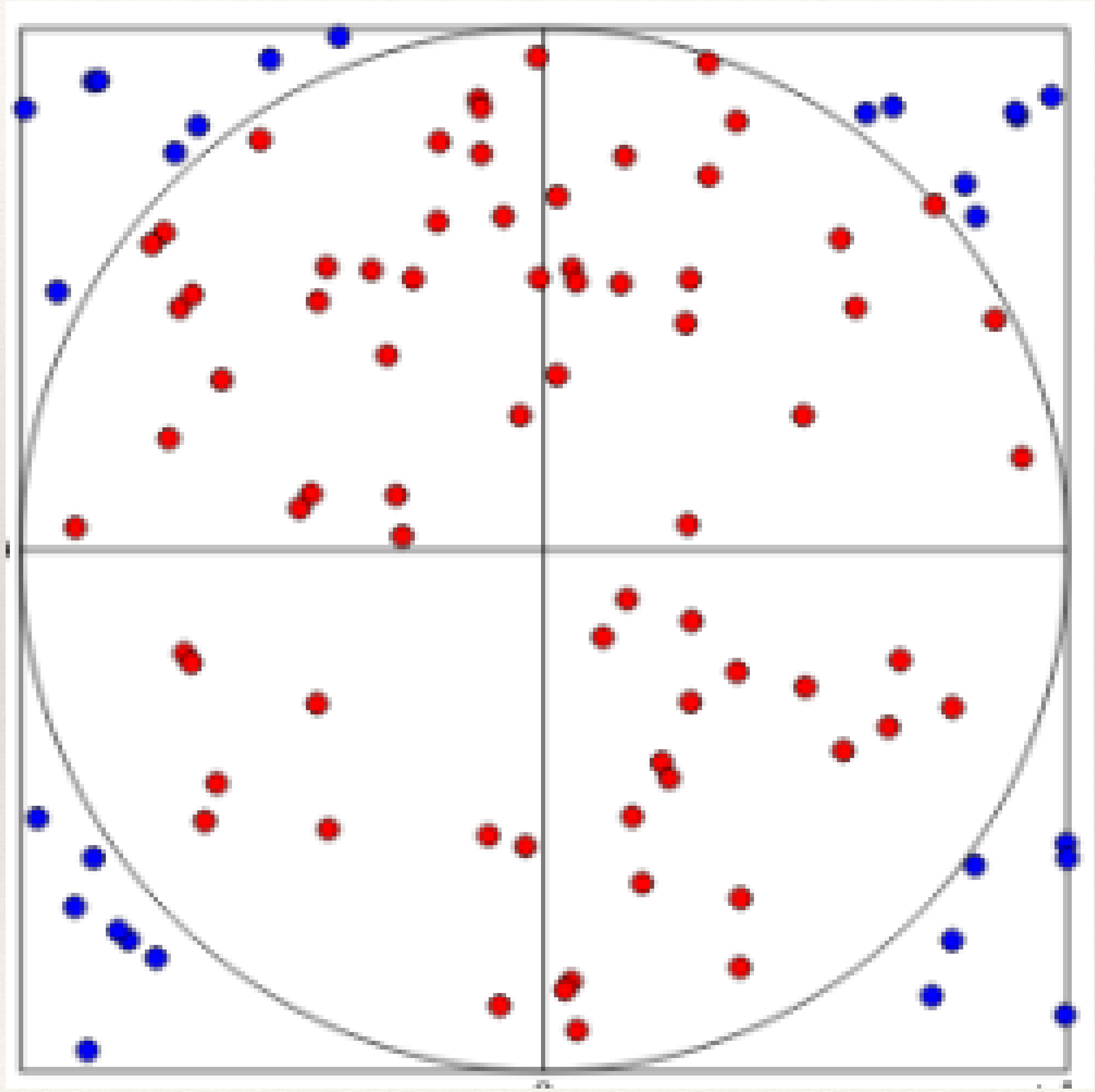
- A method of estimating the value of an uknown quantity using the principles of inferential statistics
- Inferential statistics:
  - Population: set of all possible examples
  - Sample: A proper subset of a population
  - Key factor: A random sample tends to exhibit same properties as the population from which it is drawn

Making some $\boldsymbol{\pi}$

- Consider a unit square (with area one).
- Inscribe in the square a circle with radius 1/2 (with area $\pi/_4$).
- Suppose we toss darts randomly at the square. The probability that a particular dart will land in the inscribed circle is obviously $\pi/_4$ (the ratio of the two areas).
- We can use this fact to estimate π. Toss n such darts at the square and calculate the proportion $\hat{p}_n$ that land in the circle.
- Then an estimate for π is:

$$\hat{\pi}_n = 4\hat{p}_n$$

Making some π



Making some $\boldsymbol{\pi}$

- How would we actually conduct such an experiment?
- To simulate a dart toss, suppose U_1 and U_2 are i.i.d. Unif(0,1). Then (U_1, U_2) represents the random position of the dart on the unit square.
- The dart lands in the circle if:

$$\left(U_1 - \frac{1}{2}\right)^2 + \left(U_2 - \frac{1}{2}\right)^2 \leq \frac{1}{4}$$

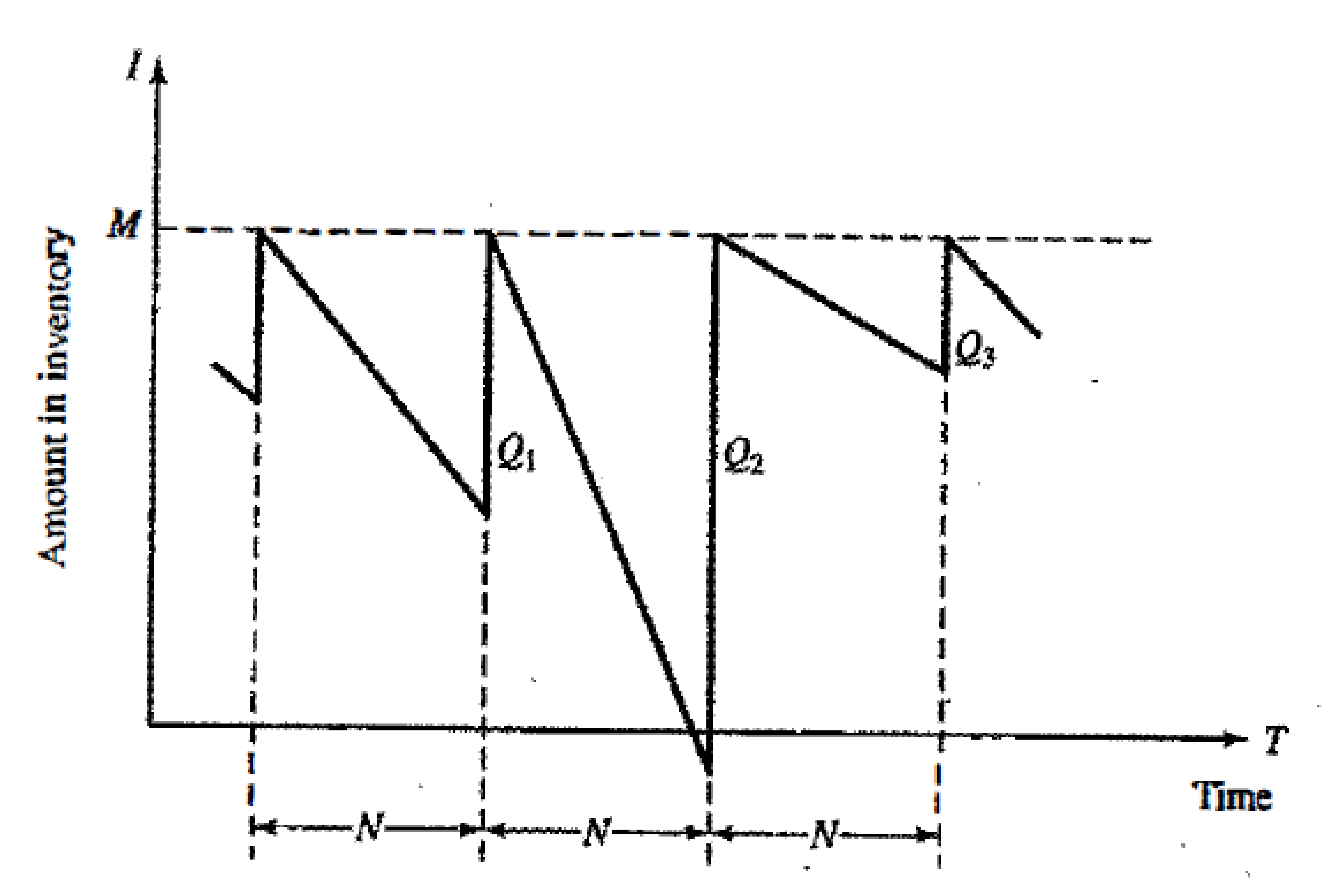
- Generate n such pairs of uniforms and count up how many of them fall in the circle.

Excel Time

- Let's use Monte-Carlo Simulation to estimate the value of the pi

Simulation of (M,N) Inventory System

- Consider a simple inventory example
  - Single product (Refrigerator)
  - Inventory is reviewed at every period N=5 days
  - An order given to bring the inventory up to level M=11.
  - Negative inventory indicates shortages
  - If there are shortages of refrigerators, they are backordered



Simulation of (M,N) Inventory System

- The daily demand for the refrigerators is distributed as follows

Demand	Probability	Cumulative Probability	Random Digit Assignment
0	0.10	0.10	01–10
1	0.25	0.35	11–35
2	0.35	0.70	36–70
3	0.21	0.91	71–91
4	0.09	1.00	92–00

Simulation of (M,N) Inventory System

- There is a random lead time given by the following distribution

Lead Time (Days)	Probability	Cumulative Probability	Random Digit Assignment
1	0.6	0.6	1–6
2	0.3	0.9	7–9
3	0.1	1.0	0

- The replenishment order is given at the end of a day, the refrigerators arrive at the beginning of the day determined by lead time, e.g., zero lead time = Replenishment done at the beginning of next day, one day lead time = Replenishment done at the beginning of the other day

Simulation of (M,N) Inventory System

- The simulation starts with an **inventory of three refrigerators** and a replenishment **order of 8 to be done after 2 days lead time**
- Generate demand for the first day. RN1: 0.26, then 1 refrigerator

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-

Simulation of (M,N) Inventory System

- Day 2
- Generate demand for the day, RN:0.68, then 2 refrigerators

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-

Simulation of (M,N) Inventory System

- Day 3, 8 refrigerators arrived
- Generate demand for the day, RN:0.33, then 1 refrigerator

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-

Simulation of (M,N) Inventory System

- Day 4
- Generate demand for the day, RN:0.39, then 2 refrigerators

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4						

Simulation of (M,N) Inventory System

- Day 5
- Generate demand for the day, RN:0.86, then 3 refrigerators
- 9 refrigerators are ordered to replenish inventory. Generate lead time, RN:0.80, then 2 days

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4	7	2	5	0	-	-
5	5	3	2	0	9	2

Simulation of (M,N) Inventory System

- Day 6
- Generate demand for the day, RN: 0.18, then 1 refrigerator

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4	7	2	5	0	-	-
5	5	3	2	0	9	2
6						

Simulation of (M,N) Inventory System

- Day 7
- Generate demand for the day, RN: 0.64, then 2 refrigerators

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4	7	2	5	0	-	-
5	5	3	2	0	9	2
6	2	1	1	0	-	-
7						

Simulation of (M,N) Inventory System

- Day 8, 9 replenishments arrived, one backorder is fulfilled
- Generate demand for the day, RN: 0.79, then 3 refrigerators

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4	7	2	5	0	-	-
5	5	3	2	0	9	2
6	2	1	1	0	-	-
7	1	2	-1	1	-	-
8	8	3	5	0	-	-

Simulation of (M,N) Inventory System

- Day 9
- Generate demand for the day, RN: 0.55, then 2 refrigerators

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4	7	2	5	0	-	-
5	5	3	2	0	9	2
6	2	1	1	0	-	-
7	1	2	-1	1	-	-
8	8	3	5	0	-	-
9						

Simulation of (M,N) Inventory System

- Day 10: Generate demand for the day, RN:0.74, then 3 refrigerators

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4	7	2	5	0	-	-
5	5	3	2	0	9	2
6	2	1	1	0	-	-
7	1	2	-1	1	-	-
8	8	3	5	0	-	-
9	5	2	3	0	-	-
10	3	3	0	0		

Simulation of (M,N) Inventory System

- Day 10: Generate lead time, RN: 0.70, then 2 days

Day	Beginning Inventory	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Lead Time
1	3	1	2	0	-	-
2	2	2	0	0	-	-
3	8	1	7	0	-	-
4	7	2	5	0	-	-
5	5	3	2	0	9	2
6	2	1	1	0	-	-
7	1	2	-1	1	-	-
8	8	3	5	0	-	-
9	5	2	3	0	-	-
10	3	3	0	0	11	2

End of Simulation

Table 2.21 Simulation Tables for (*M*, *N*) Inventory System

Day	Cycle	Day within Cycle	Beginning Inventory	Random Digits for Demand	Demand	Ending Inventory	Shortage Quantity	Order Quantity	Random Digits for Demand	Lead Time (days)	Days until Order Arrives
1	1	1	3	26	1	2	0	—	—	—	1
2	1	2	2	68	2	0	0	—	—	—	—
3	1	3	8	33	1	7	0	—	—	—	—
4	1	4	7	39	2	5	0	—	—	—	—
5	1	5	5	86	3	2	0	9	8	2	2
6	2	1	2	18	1	1	0	—	—	—	1
7	2	2	1	64	2	0	1	—	—	—	—
8	2	3	9	79	3	5	0	—	—	—	—
9	2	4	5	55	2	3	0	—	—	—	—
10	2	5	3	74	3	0	0	11	7	2	2
11	3	1	0	21	1	0	1	—	—	—	1
12	3	2	0	43	2	0	3	—	—	—	—
13	3	3	11	49	2	6	0	—	—	—	—
14	3	4	6	90	3	3	0	—	—	—	—
15	3	5	3	35	1	2	0	9	2	1	1
16	4	1	2	08	0	2	0	—	—	—	—
17	4	2	11	98	4	7	0	—	—	—	—
18	4	3	7	61	2	5	0	—	—	—	—
19	4	4	5	85	3	2	0	—	—	—	—
20	4	5	2	81	3	0	1	12	3	1	1
21	5	1	0	53	2	0	3	—	—	—	—
22	5	2	12	15	1	8	0	—	—	—	—
23	5	3	8	94	4	4	0	—	—	—	—
24	5	4	4	19	1	3	0	—	—	—	—
25	5	5	3	44	2	1	0	10	1	1	1
Total						68	9				
Average					2.04	2.72	0.36				

End of Simulation

- Average ending inventory is 2.72
- Average shortage quantity is 0.36
- Different policies can be simulated by changing M and N.
- Costs may also be assigned to holding refrigerators in the inventory per day, replenishments and shortages.