4th Class

Simulation & Modeling

النمذجة والمحاكاة

أستاذة المادة: أ.م.د. هناء محسن

Simulation & Modeling

Introduction:

System: A set of objects interact and depend on each other.

Components of system

- Entity: An object of interest in the system, for an example machines in factories.
- Attribute: The property of an entity, for an example of the speed and amplitude.
- Activity: A time period of specified length, for an example of welding
- State of the System: A collection of variables that describe the system in any time for an example the case of system goals machinery (working, not working, unemployed)
- Events: A instantaneous occurrence that might change the state of the system:

  There are two types of event:

  1. Endogenous events (self-evolution): Activities and events occurring with the system
  2. Exogenous (external events): Activities and events occurring with the environment

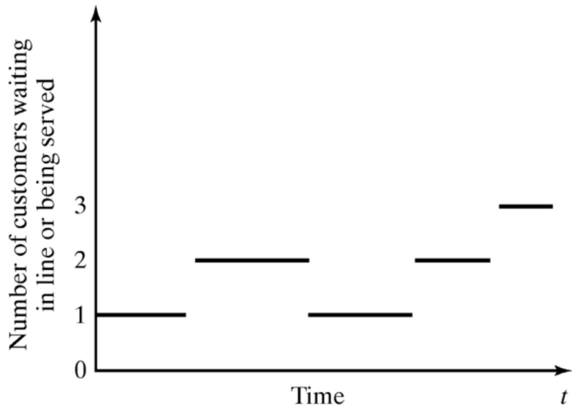
Example of Systems and components:

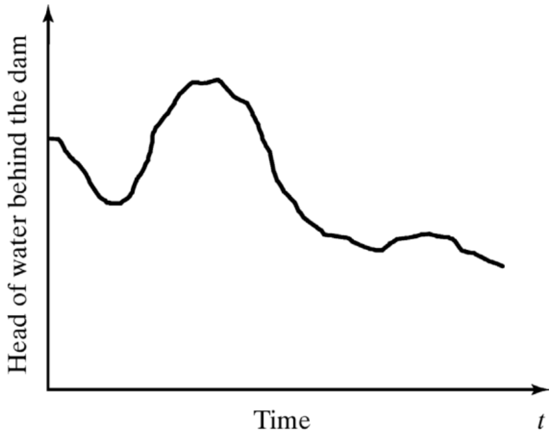
System	Entities	Attributes	Activities	Events	State Variables
Banking	Customers	Checking account balance	Making deposits	Arrival; Departure	Number of busy tellers; number Of customers waiting
Production	Machines	Speed; capacity; breakdown rate	Welding; stamping	Breakdown	Status of machines (busy, idle, or down)
Communications	Messages	Length; destination	Transmitting	Arrival at destination	Number waiting to be transmitted

Continuous and Discrete System:

Systems can be classified through the affected systems in time to continuous or intermittent systems:

1. **Discrete systems** are of changes in the case of variable rate or qualities at intermittent points is constant During a period of time
2. **Continuous systems** is of changes in one or more of the variable rate on an ongoing basis during the period of time





Model of a system:

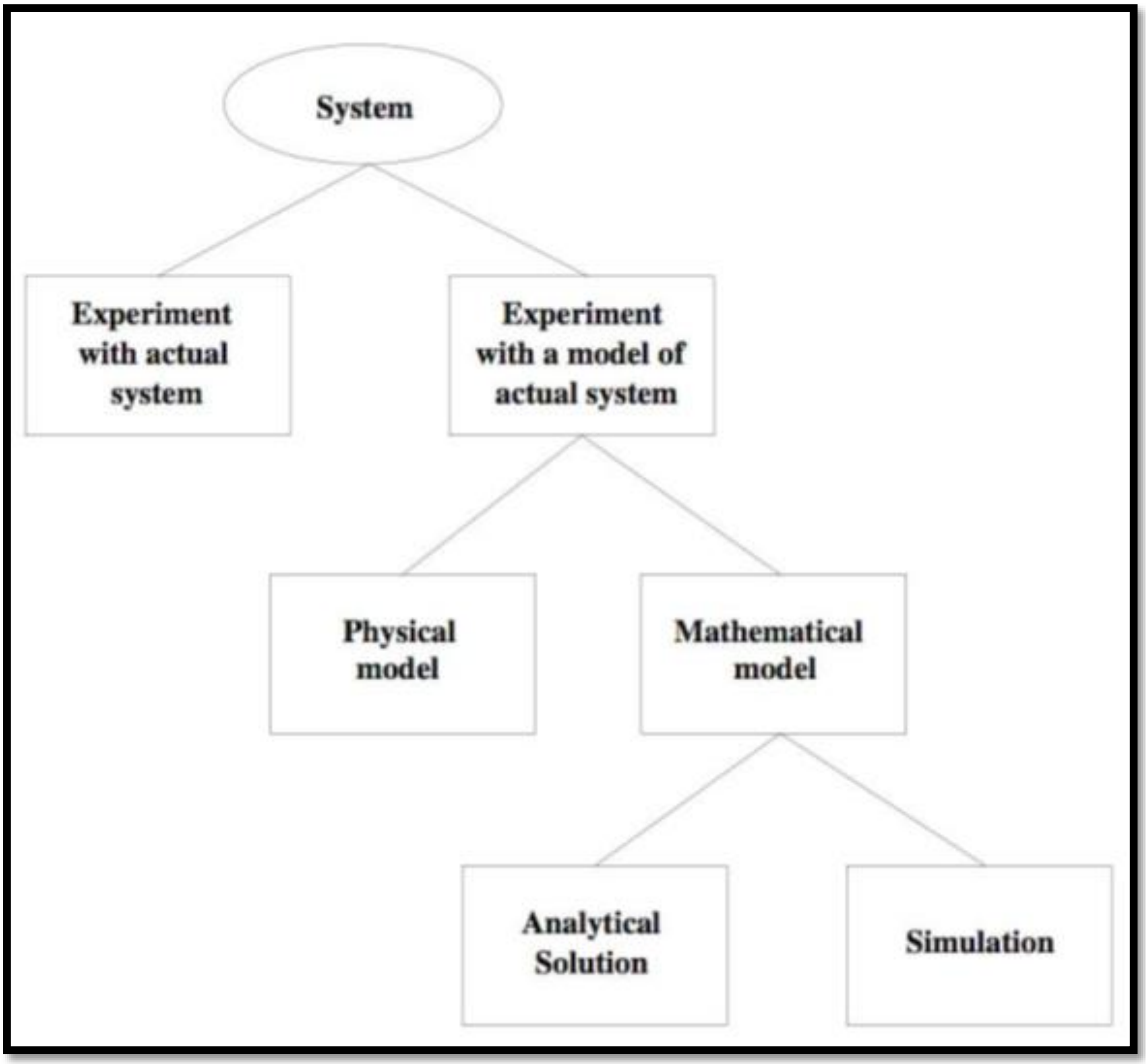
Model: construct a conceptual framework that describes a system

Component of a model: Components of the model are the same as system components which entities recipes and can contain any other components that are important to the study

Types of models

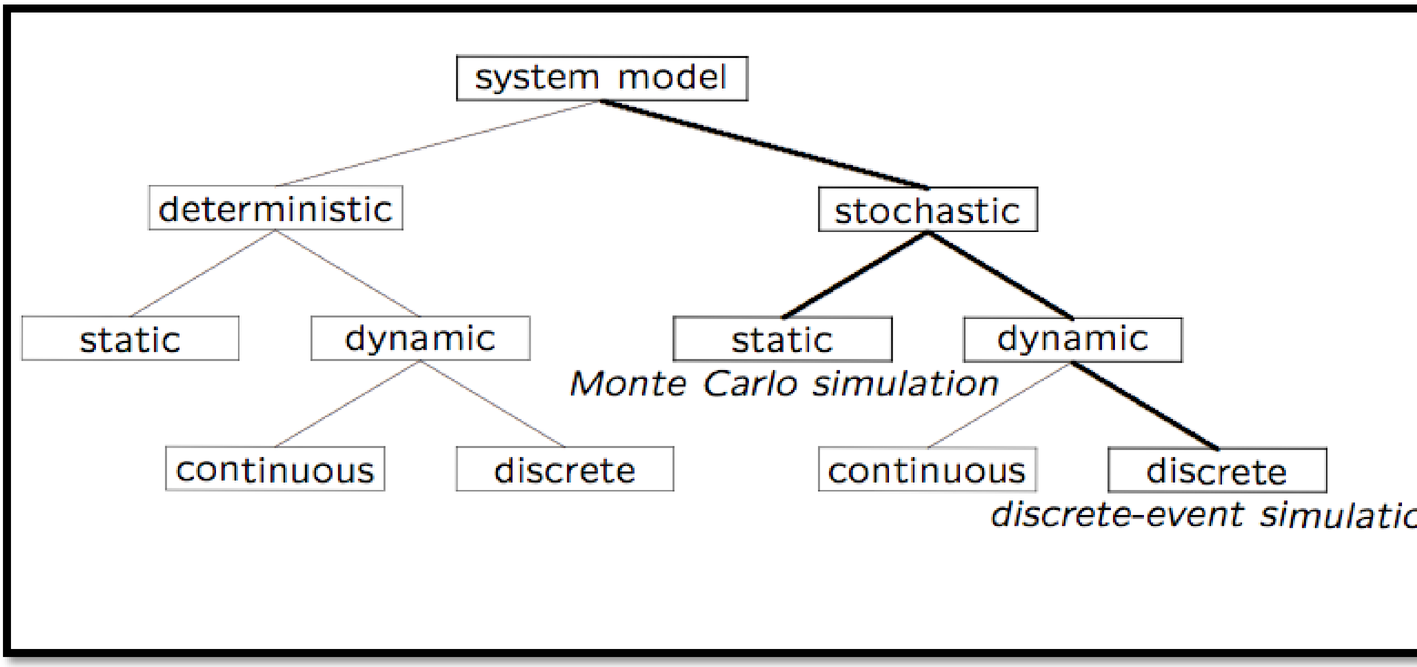
1. **Physical models:** They represent physical systems and the construction process is very expensive
2. **mathematical models:** They used symbols and mathematical equations to represent the system and the

simulation model is a kind



Model Taxonomy:

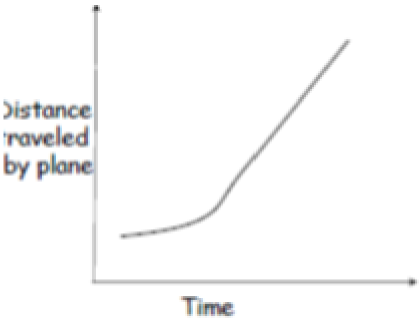
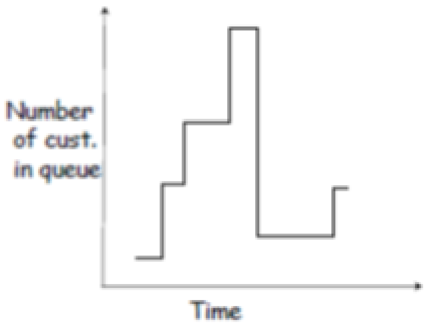
The types of models can be represented graphically as follows



In the previous illustration divided static mathematical models to analytical and numerical and mathematical models, dynamic and analytical and numerical simulation because very little of the mathematical models can be solved analytically using mathematical theories and logical conclusions, and most of them can be solved numerically, using numerical methods and algorithms, all of which can be solved by simulation.

Model Classification

1. Continuous and Discrete Models

 (a) Continuous	 (b) Discrete
Continuous models - State variables change continuously with time	Discrete models – State variables change only at discrete points in time (e.g., when a customer arrives or leaves the system as in the superstore example)

2. Deterministic and Stochastic Models

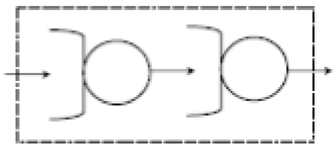
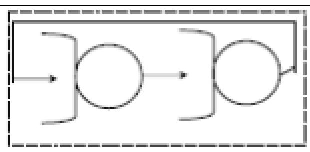
Deterministic models produce deterministic results	Stochastic or probabilistic models are subject to random effects Typically, they have one or more random inputs (e.g., arrival of customers, service time etc.). Outputs from stochastic models are “estimates” of the true characteristics of the system Need to repeat experiments number of times Need to have confidence in the

	results

3. Static and dynamic models

Static models – system state independent of time	Dynamic models - system state change with time

4. Open and closed models

 (a) Open Model	 (b) Closed Model

5. Linear and non-linear models

Linear models – output is a linear function of input parameters	

What is a System Model

A model of a system: is created by making assumptions and approximations of the system (also referred to as facility) being studied.

Assumption and approximations can be logical or mathematical Models have many applications and can answer questions such as:

- Why does my Web performance suffer when my room-mate starts using the WiFi connection?

- What will be the path of a hurricane? Etc.

When to use simulations?

Simulations can be used:

1. To compare design alternatives for a system that doesn't exist.
2. To study the effect of alterations to an existing system. Why not change the system
3. To reinforce/verify analytic solutions.
4. Simulations should not be used:
5. If model assumptions are simple such that mathematical methods can be used to obtain exact answers (analytical solutions)
6. Note that simulations provide estimates of system behavior

Types of Simulation

1. **Monte Carlo simulation**
    - Static simulations with no time component
    - Used for evaluating non-probabilistic expressions (e.g., an integral) using probabilistic methods
    - Wide variety of mathematical problems

2. **Trace-driven simulation**
    - Extensively used in computer systems performance evaluation; e.g., paging algorithms

- Advantages: credibility, easy validation, less randomness, accurate workload
- Disadvantages: complexity, only a snap-shot, representative?, single point of validation

3. **Discrete-event simulation**:
   a simulation using a discrete-state model of the system, E.g., Widely used for studying computer systems

4. **Continuous-event simulation:**
   Uses a continuous-state models, E.g., Widely used in chemical/pharmaceutical studies

Simulation objectives:

1. To study the current system.
2. Analysis of some of the proposed regulations.
3. Planning and design of more sophisticated systems

When Simulation Is Appropriate Tool:

The simulation has the following features:

1. Simulations have the capacity to study everything related to partial systems Complex.
2. Simulation of informatics and administrative changes and environmental study and see the effect of alternatives on the form.

3. Simulation systems in the process contribute to the design provide to develop a system under Discussion.

4. Change the input values and simulation see output to determine the most important variables and how Interaction variables.

5. Use of simulation as a tool to strengthen education and correct analytical solution methods.

6. Using simulation to test a new design or new ways by the application to see what may Happen.

7. emulates different machine that can determine their needs.

8. Simulation model designed for training allows learning from non-cost.

9. Modern systems and factories are too complex for that interaction is only through simulation.

When Simulation Is not Appropriate:

1. When the problem can be solved by common sense.
2. When the problem can be solved analytically.
3. If it is easier to perform direct experiments.
4. If cost exceed savings.
5. If resource or time are not available.
6. If system behavior is too complex. Like human behavior

Simulation advantage and disadvantage:

1. The cost of the analysis models much less than the cost of similar experiments performed on the system Real.

2. Enables models of time pressure.
3. The cost of the error in the attempt less experience when using models than in reality.
4. - Allows use of risk models calculate the specific measures.
5. Mathematical models able to analyze a large number of solutions.
6. - Promote models and strengthen education and training.

Use of models:

1. Facilitate understanding: be the simplest model of the system which is understood more easily if the representation of its, and the relationships between them in a simplified manner.
2. Facilitate contact: Once you understand the existing resolve the problem of the system often need to connect this understanding, to others.
3. To predict the future: mathematical model can predict what can happen in the future but It may not be 100% accurate in this case.
4. Called activity using the simulation model

How to develop a model:

1) Determine the goals and objectives

2) Build a ***conceptual*** model

3) Convert into a ***specification*** model

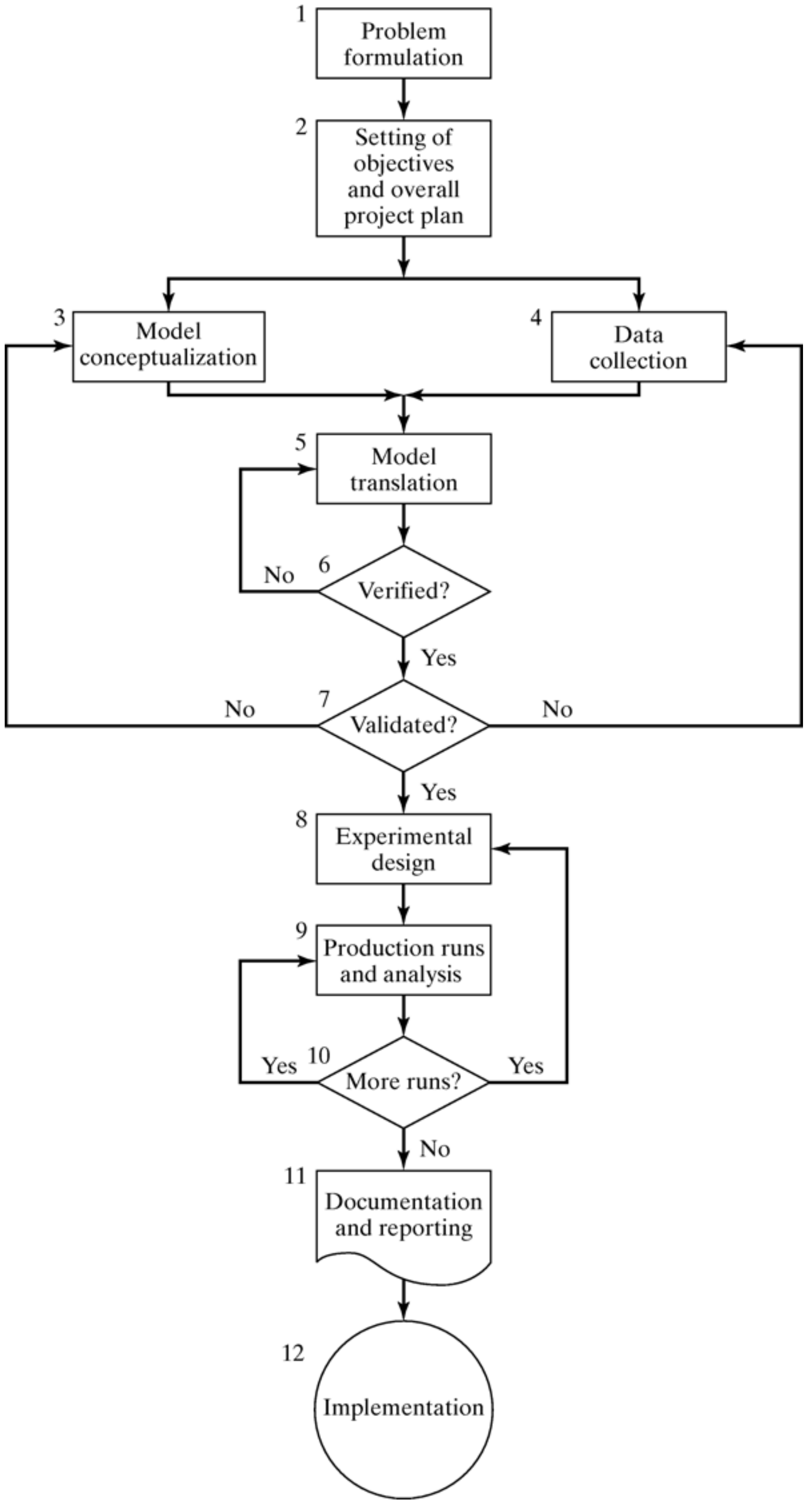
4) Convert into a ***computational*** model

5) Verify

6) Validate

Typically an iterative process

Steps in Simulation Study



Properties of Random Numbers

- Two important statistical properties:
  - Uniformity
  - Independence.
- Random Number, R_i, must be independently drawn from a uniform distribution with pdf:

$$f(x) = \begin{cases} 1, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

$$E(R) = \int_0^1 x\,dx = \left.\frac{x^2}{2}\right|_0^1 = \frac{1}{2}$$

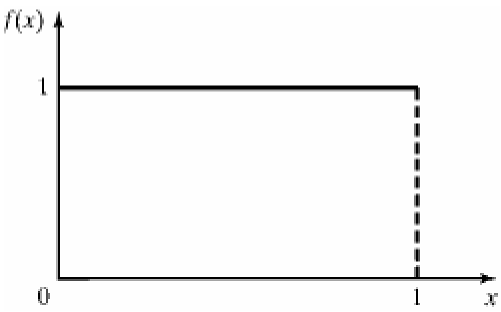


Figure: pdf for random numbers

Generation of Pseudo-Random Numbers

- "Pseudo", because generating numbers using a known method removes the potential for true randomness.
- Goal: To produce a sequence of numbers in [*0*, *1*] that simulates, or imitates, the ideal properties of random numbers (RN).
- Important considerations in RN routines:
  - Fast
  - Portable to different computers
  - Have sufficiently long cycle
  - Replicable
  - Closely approximate the ideal statistical properties of uniformity and independence.

Techniques for Generating Random Numbers

- Linear Congruential Method (LCM).
- Combined Linear Congruential Generators (CLCG).
- Random-Number Streams.

Linear Congruential Method

To produce a sequence of integers, X_1, X_2, ... between 0 and $m-1$ by following a recursive relationship:

$$X_{i+1} = (\underbrace{a}_{\text{The multiplier}} X_i + \underbrace{c}_{\text{The increment}}) \bmod \underbrace{m}_{\text{The modulus}}, \quad i = 0,1,2,...$$

The selection of the values for a, c, m, and X_0 drastically affects the statistical properties and the cycle length.
The random integers are being generated $[0, m-1]$, and to convert the integers to random numbers:

$$R_i = \frac{X_i}{m}, \quad i = 1,2,...$$

---

1. LCM: Example

- Use $X_0 = 27$, $a = 17$, $c = 43$, and $m = 100$.
- The X_i and R_i values are:

$X_1 = (17*27+43) \bmod 100 = 502 \bmod 100 = 2$, $R_1 = 0.02$;
$X_2 = (17*2+43) \bmod 100 = 77$, $R_2 = 0.77$;
$X_3 = (17*77+43) \bmod 100 = 52$, $R_3 = 0.52$;
...

Characteristics of a Good Generator

- Maximum Density
  - Such that he values assumed by R_i, $i = 1, 2, \ldots$, leave no large gaps on $[0, 1]$
  - Problem: Instead of continuous, each R_i is discrete
  - Solution: a very large integer for modulus m
    - Approximation appears to be of little consequence
- Maximum Period
  - To achieve maximum density and avoid cycling.
  - Achieve by: proper choice of a, c, m, and X_0.
- Most digital computers use a binary representation of numbers
  - Speed and efficiency are aided by a modulus, m, to be (or close to) a power of 2.

Combined Linear Congruential Generators

- Reason: Longer period generator is needed because of the increasing complexity of stimulated systems.
- Approach: Combine two or more multiplicative congruential generators.
- Let $X_{i,1}, X_{i,2}, \ldots, X_{i,k}$, be the i^{th} output from k different multiplicative congruential generators.
  - The j^{th} generator:
    - Has prime modulus m_j and multiplier a_j and period is $m_j - 1$
    - Produces integers $X_{i,j}$ is approx ~ Uniform on integers in $[1, m-1]$
    - $W_{i,j} = X_{i,j} - 1$ is approx ~ Uniform on integers in $[1, m-2]$

- Suggested form:

$$X_i = \left(\sum_{j=1}^{k} (-1)^{j-1} X_{i,j} \right) \bmod m_1 - 1 \qquad \text{Hence, } R_i = \begin{cases} \dfrac{X_i}{m_1}, & X_i \succ 0 \\ \dfrac{m_1 - 1}{m_1}, & X_i = 0 \end{cases}$$

The coefficient: Performs the subtraction $X_{i,1-1}$

  - The maximum possible period is:

$$P = \frac{(m_1 - 1)(m_2 - 1)\ldots(m_k - 1)}{2^{k-1}}$$

- Example: For 32-bit computers, L'Ecuyer [1988] suggests combining k = 2 generators with m_1 = 2,147,483,563, a_1 = 40,014, m_2 = 2,147,483,399 and a_2 = 20,692. The algorithm becomes:

  Step 1: Select seeds
  - $X_{1,0}$ in the range [1, 2,147,483,562] for the 1st generator
  - $X_{2,0}$ in the range [1, 2,147,483,398] for the 2nd generator.

  Step 2: For each individual generator,

  $X_{1,j+1}$ = 40,014 $X_{1,j}$ mod 2,147,483,563

  $X_{2,j+1}$ = 40,692 $X_{1,j}$ mod 2,147,483,399.

  Step 3: $X_{j+1} = (X_{1,j+1} - X_{2,j+1})$ mod 2,147,483,562.

  Step 4: Return

$$R_{j+1} = \begin{cases} \dfrac{X_{j+1}}{2{,}147{,}483{,}563}, & X_{j+1} > 0 \\ \dfrac{2{,}147{,}483{,}562}{2{,}147{,}483{,}563}, & X_{j+1} = 0 \end{cases}$$

  Step 5: Set $j = j+1$, go back to step 2.

- Combined generator has period: $(m_1 - 1)(m_2 - 1)/2 \sim 2 \times 10^{18}$

Random-Numbers Streams

- **The seed for a linear congruential random-number generator:**
  - Is the integer value X_0 that initializes the random-number sequence.
  - Any value in the sequence can be used to "seed" the generator.
- **A random-number stream:**
  - Refers to a starting seed taken from the sequence $X_0, X_1, \ldots, X_P$.
  - If the streams are b values apart, then stream i could defined by starting seed: $$S_i = X_{b(i-1)}$$
  - Older generators: $b = 10^5$; Newer generators: $b = 10^{37}$.
- **A single random-number generator with k streams can act like k distinct virtual random-number generators**
- **To compare two or more alternative systems.**
  - Advantageous to dedicate portions of the pseudo-random number sequence to the same purpose in each of the simulated systems.

Tests for Random Numbers

- **Two categories:**
  - **Testing for uniformity:**

    $$H_0: \quad R_i \sim U[0,1]$$
    $$H_1: \quad R_i \nsim U[0,1]$$

    - Failure to reject the null hypothesis, H_0, means that evidence of non-uniformity has not been detected.
  - **Testing for independence:**

    H_0: $R_i \sim$ independently

    H_1: $R_i \nsim$ independently

    - Failure to reject the null hypothesis, H_0, means that evidence of dependence has not been detected.
- **Level of significance α, the probability of rejecting H_0 when it is true:** $\alpha = P(\text{reject } H_0 | H_0 \text{ is true})$

- When to use these tests:
  - If a well-known simulation languages or random-number generators is used, it is probably unnecessary to test
  - If the generator is not explicitly known or documented, e.g., spreadsheet programs, symbolic/numerical calculators, tests should be applied to many sample numbers.
- Types of tests:
  - Theoretical tests: evaluate the choices of m, a, and c without actually generating any numbers
  - Empirical tests: applied to actual sequences of numbers produced. Our emphasis.

Frequency Tests

- Test of uniformity
- Two different methods:
  - Kolmogorov-Smirnov test
  - Chi-square test

Kolmogorov-Smirnov Test

- Compares the continuous cdf, $F(x)$, of the uniform distribution with the empirical cdf, $S_N(x)$, of the N sample observations.
  - We know: $F(x) = x, \quad 0 \le x \le 1$
  - If the sample from the RN generator is $R_1, R_2, \ldots, R_N$, then the empirical cdf, $S_N(x)$ is:

$$S_N(x) = \frac{\text{number of } R_1, R_2, \ldots, R_n \text{ which are} \le x}{N}$$

- Based on the statistic: $D = max| F(x) - S_N(x)|$
  - **Sampling distribution of D is known** (a function of N, tabulated in Table A.8.)
- A more powerful test, recommended.

- Example: Suppose 5 generated numbers are 0.44, 0.81, 0.14, 0.05, 0.93.

Step 1: Arrange $R_{(i)}$ from smallest to largest

Step 2: $D^+ = max \{i/N – R_{(i)}\}$, $D^- = max \{R_{(i)} - (i-1)/N\}$

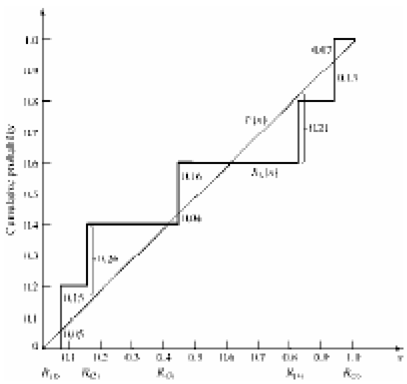
Step 1:	$R_{(i)}$	0.05	0.14	0.44	0.81	0.93
	i/N	0.20	0.40	0.60	0.80	1.00
Step 2:	$i/N – R_{(i)}$	0.15	0.26	0.16	-	0.07
	$R_{(i)} – (i-1)/N$	0.05	-	0.04	0.21	0.13

Step 3: $D = max(D^+, D^-) = 0.26$

Step 4: For $\alpha = 0.05$,

$D_\alpha = 0.565 > D$

Hence, H_0 is not rejected.



17

Chi-square test

- Chi-square test uses the sample statistic:

n is the # of classes

$$X_0^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

E_i is the expected # in the i^{th} class

O_i is the observed # in the i^{th} class

  - **Approximately the chi-square distribution with *n-1* degrees of freedom** (where the critical values are tabulated in Table A.6)
  - **For the uniform distribution, E_i, the expected number in the each class is:**

$$E_i = \frac{N}{n}, \quad \text{where N is the total \# of observation}$$

- Valid only for large samples, e.g. N >= 50

Tests for Autocorrelation

- Testing the autocorrelation between every m numbers (m is a.k.a. the lag), starting with the i^{th} number
  - **The autocorrelation ρ_{im} between numbers: R_i, R_{i+m}, R_{i+2m}, $R_{i+(M+1)m}$**
  - ***M* is the largest integer such that** $i + (M+1)m \le N$
- Hypothesis:

$$H_0: \ \rho_{im} = 0, \quad \text{if numbers are independent}$$
$$H_1: \ \rho_{im} \neq 0, \quad \text{if numbers are dependent}$$

- If the values are uncorrelated:
  - **For large values of M, the distribution of the estimator of ρ_{im}, denoted $\hat{\rho}_{im}$ is approximately normal.**

- Testing the autocorrelation between every m numbers (m is a.k.a. the lag), starting with the i^{th} number
  - The autocorrelation ρ_{im} between numbers: R_i, R_{i+m}, R_{i+2m}, $R_{i+(M+1)m}$
  - M is the largest integer such that $i+(M+1)m \leq N$
- Hypothesis:

$$H_0: \quad \rho_{im} = 0, \quad \text{if numbers are independent}$$
$$H_1: \quad \rho_{im} \neq 0, \quad \text{if numbers are dependent}$$

- If the values are uncorrelated:
  - For large values of M, the distribution of the estimator of ρ_{im}, denoted $\hat{\rho}_{im}$ is approximately normal.

- Test statistics is:

$$Z_0 = \frac{\hat{\rho}_{im}}{\hat{\sigma}_{\hat{\rho}_{im}}}$$

  - Z_0 is distributed normally with mean = *0* and variance = *1*, and:

$$\hat{\rho}_{im} = \frac{1}{M+1}\left[\sum_{k=0}^{M} R_{i+km} R_{i+(k+1)m}\right] - 0.25$$

$$\hat{\sigma}_{\rho_{im}} = \frac{\sqrt{13M+7}}{12(M+1)}$$

- If $\rho_{im} > 0$, the subsequence has positive autocorrelation
  - High random numbers tend to be followed by high ones, and vice versa.
- If $\rho_{im} < 0$, the subsequence has negative autocorrelation
  - Low random numbers tend to be followed by high ones, and vice versa.

<u>Example</u>

- Test whether the 3^{rd}, 8^{th}, 13^{th}, and so on, for the following output on P. 265.
  - Hence, $\alpha = 0.05$, $i = 3$, $m = 5$, $N = 30$, and M = 4

$$\hat{\rho}_{35} = \frac{1}{4+1}\begin{bmatrix}(0.23)(0.28)+(0.25)(0.33)+(0.33)(0.27)\\+(0.28)(0.05)+(0.05)(0.36)\end{bmatrix} - 0.25$$
$$= -0.1945$$
$$\hat{\sigma}_{\rho_{35}} = \frac{\sqrt{13(4)+7}}{12(4+1)} = 0.128$$
$$Z_0 = -\frac{0.1945}{0.1280} = -1.516$$

  - From Table A.3, $z_{0.025} = 1.96$. Hence, the hypothesis is not rejected.

Shortcomings

- The test is not very sensitive for small values of M, particularly when the numbers being tests are on the low side.
- Problem when "fishing" for autocorrelation by performing numerous tests:
  - If $\alpha = 0.05$, there is a probability of 0.05 of rejecting a true hypothesis.
  - If 10 independence sequences are examined,
    - The probability of finding no significant autocorrelation, by chance alone, is $0.95^{10} = 0.60$.
    - Hence, the probability of detecting significant autocorrelation when it does not exist = *40%*

Random-Variate Generation

Random-Variate generation is converting from a random number (R_i) to a Random Variable, X_i ~ *some distribution.*

Prerequisite

All the techniques assume that a source of uniform [0,1] random numbers $R_1, R_2, \ldots$ is readily available, where each R_i has **pdf** :

$$f_R(x)=\begin{cases}1, & 0\le x\le 1\\ 0, & \text{otherwise}\end{cases}$$

and **cdf** :

$$F_R(x)=\begin{cases}0, & x<0\\ x, & 0\le x\le 1\\ 1, & x>1\end{cases}$$

General Techniques to Generating Random Variates

Illustrate some widely-used techniques for generating random variates.

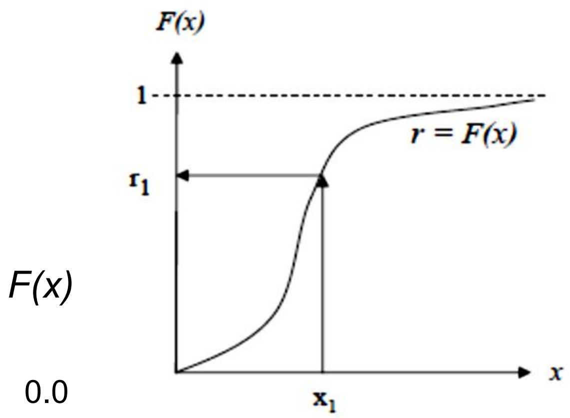
- Inverse-transform technique
- Acceptance-rejection technique
- Special properties

1- Inverse-transform technique

The concept:

- For cumulative distribution function CDF function: r = F(x)
- Generate r from uniform (0,1), a.k.a U(0,1)
- Find x,

$x = F^{-1}(r)$



- The inverse-transform technique can be used in principle for any distribution.
- Most useful when the CDF F(x) has an inverse $F^{-1}(x)$.

Steps in inverse-transform technique:

Step1. Compute the cdf of the desired random variable X:

$$F(x) = 1 - e^{-\lambda x} \qquad x \geq 0$$

Step2. Set F(X) = R on the range of X

Step3. Solve the equation F(x) = R for X in terms of R.

$$1 - e^{-\lambda X} = R$$
$$e^{-\lambda X} = 1 - R$$
$$-\lambda X = \ln(1 - R)$$
$$X = -\frac{1}{\lambda}\ln(1 - R)$$

Step 4. Generate (as needed) uniform random numbers R_1, R_2, R_3, . . and compute the desired random variates.

Exponential Distribution

- Exponential PDF:

$$F(x) = \lambda e^{-\lambda x} \qquad \text{for } x>=0$$

- Exponential CDF:

$$r = F(x)$$
$$= 1 - e^{-\lambda x} \qquad \text{for } x \geq 0$$

To generate $X_1, X_2, X_3 \ldots$

$$X_i = F^{-1}(R_i)$$

$$= -(1/\lambda) \ln(1-R_i)$$

Example1: Table below gives a sequence of random numbers and the computed exponential variates, Xi, given by Equation ($x=-1/\lambda \ln(1-R_i)$) with the value $\lambda = 1$. Figure (a) is a histogram of 200 values, R_1, R2, · · ,R_{200} from the uniform distribution, and Figure (b) is a histogram of the 200 values, X_1 , X_2, ·.· · X_{200} , computed by Equation ($x=-1/\lambda \ln(1-R_i)$). Compare these empirical histograms with the theoretical density functions in Figure (c) and (d). As illustrated here, a histogram is an estimate of the underlying density function.

i	R_i	X_i
1	0.1306	0.1400
2	0.0422	0.0431
3	0.6597	1.078
4	0.7965	1.592
5	0.7696	1.468

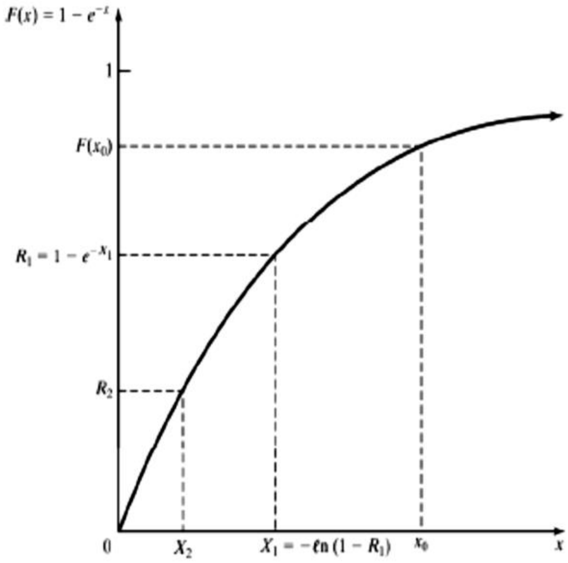
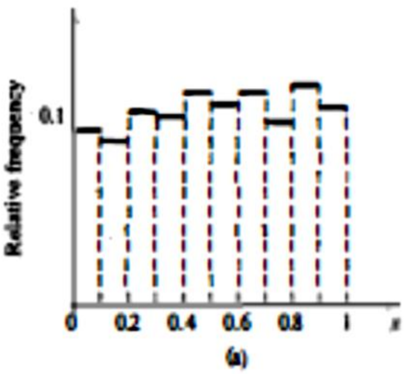
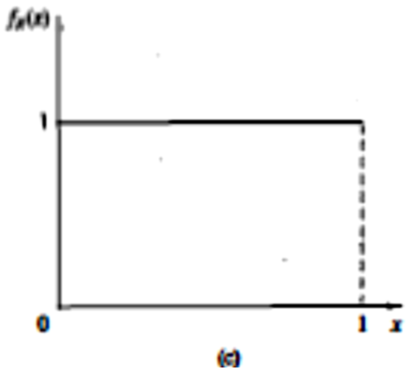
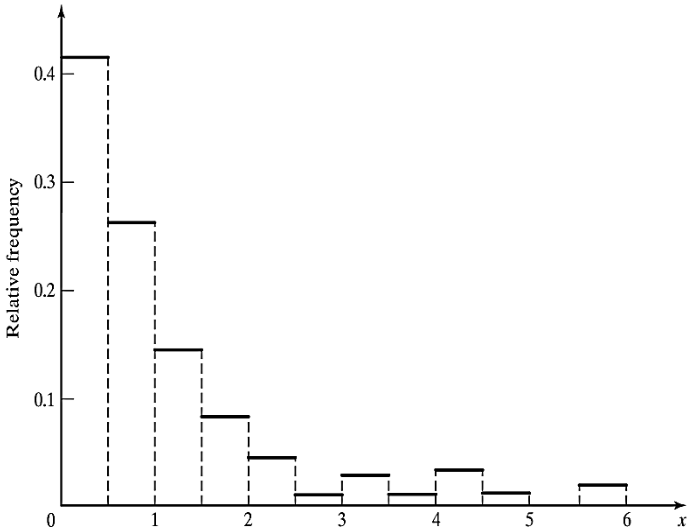


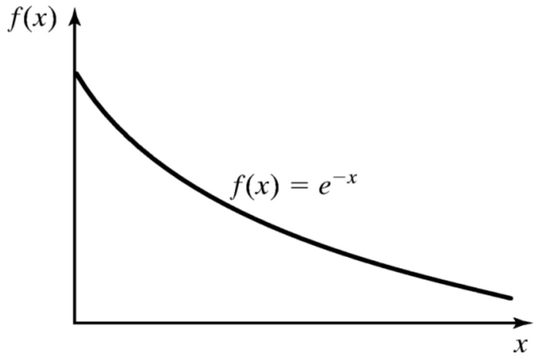
Figure: Inverse-transform technique for *exp(λ = 1)*







(b)



(d)

Example2: The packet sizes (trimodal) probabilities:

Size	probability
64 byte	0.7
128 byte	0.1
512 byte	0.2

The CDF for this distribution is:

$$F(x) = \begin{cases} 0.0 & 0 \leq x < 64 \\ 0.7 & 64 \leq x < 128 \\ 0.8 & 128 \leq x < 512 \\ 1.0 & 512 \leq x \end{cases}$$

The inverse function is:

$$F^{-1}(u) = \begin{cases} 64 & 0 < u \leq 0.7 \\ 128 & 0.7 < u \leq 0.8 \\ 512 & 0.8 < u \leq 1 \end{cases}$$

$$\text{Generate } u \sim U(0,1)$$
$$u \leq 0.7 \Rightarrow Size = 64$$
$$0.7 < u \leq 0.8 \Rightarrow size = 128$$
$$0.8 < u \Rightarrow size = 512$$

Note: CDF is *continuous from the right*

⇒ the value on the right of the discontinuity is used

⇒ The inverse function is continuous from the left

⇒ u=0.7 ⇒ x=64

Other Distributions

Examples of other distributions for which inverse **CDF** works are:

· **Uniform distribution:**

Consider a random variable X that is uniformly distributed on the interval [a, b]. A reasonable guess for generating X is given by:

$$X=a+(b-a)R$$

The cdf is given by:

$$F(x)=\begin{cases} 0, & x<a \\ x-a/b-a, & a\le x\le b \\ 1, & x>b \end{cases}$$

Set $F(X)=(X-a)/(b-a)=R$, Solving for X yields $X=a+(b-a)R$

Therefore the random variate is: $X=a+(b-a)R$

· **Weibull distribution:**

model for time to failure for machines or electronic components.

$$X = a[- \ln(1 - R)]^{1/b}$$

· **Triangular distribution:**

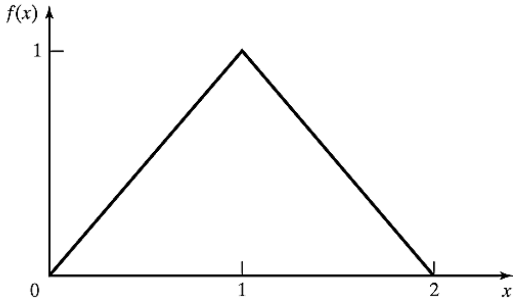
The cdf is given by:

$$F(x)=\begin{cases} 0, & x\le 0 \\ x^2/2, & 0<x\le 1 \\ 1-((2-x)^2/2), & 1<x\le 2 \\ 1, & x>2 \end{cases}$$

For $0\le X\le 1$, $R=X^2/2$ and for $1\le X\le 2$, $R=1-((2-X)^2/2)$

Thus X is generated by:

$$X = \begin{cases} \sqrt{2R}, & 0 \le R \le 1/2 \\ 2 - \sqrt{2(1-R)}, & 1/2 < R \le 1 \end{cases}$$



Empirical Continuous Distributions:

When theoretical distribution is not applicable that provides a good model, then it is necessary to use empirical distribution of data

To collect empirical data:

- One possibility is to resample the observed data itself
  - This is known as *using the empirical distribution*
  - It makes sense if the input process takes a finite number of values
- If the data is drawn from a continuous valued input process, then we can interpolate between the observed data points to fill in the gaps

Given a small sample set (size *n*):

- Arrange the data from smallest to largest $X_1 \le x_2 \le \ldots\ldots\ldots \le x_n$
- Define $x_{(0)} = 0$
- Assign the probability *1/n* to each interval $x_{(i-1)} \le x \le x_{(i)}$
- The resulting empirical cdf has a i^{th} line segment slope as

$$a_i = \frac{x_{(i)} - x_{(i-1)}}{1/n - (i-1)/n} = \frac{x_{(i)} - x_{(i-1)}}{1/n}$$

- The inverse cdf is calculated by

$$\boxed{X = \hat{F}^{-1}(R) = x_{(i-1)} + a_i\left(R - \frac{(i-1)}{n}\right)}$$

where $(i\text{-}1)/n < R \le i/n$; *R is the random number generated*

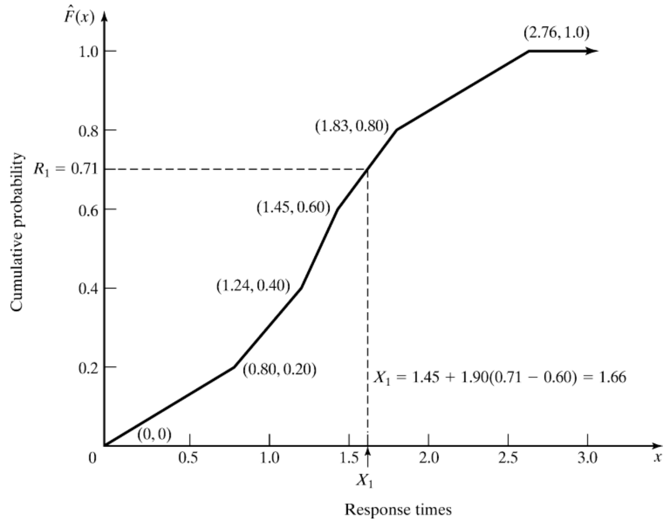
Example: Five observations of fire crew response times (in minutes) to incoming alarms are collected to be used in a simulation investigating possible alternative staffing and crew-scheduling policies.

The data are: 2.76, 1.83, 0.80, 1.45, 1.24

First: Arranging in ascending order: 0.80, 1.24, 1.45, 1.83, 2.76

i	*Interval (minutes)*	*Probability 1/n*	*Cumulative Probability, i/n*	*Slope, a_i*
1	0 ≤ x ≤ 0.80	0.2	0.20	4.00
2	0.80 ≤ x ≤ 1.24	0.2	0.40	2.2
3	1.24 ≤ x ≤ 1.45	0.2	0.60	1.1
4	1.45 ≤ x ≤ 1.83	0.2	0.80	1.9
5	1.83 ≤ x ≤ 2.76	0.2	1.00	4.65

$$R_1 = 0.71$$
$$X_1 = x_{(4-1)} + a_4(R_1 - (4-1)/n)$$
$$= 1.45 + 1.90(0.71 - 0.6)$$
$$= 1.66$$



In general, given c_i is the cumulative probability of first *i* intervals, $x_{(i-1)} \le x \le x_{(i)}$ is the i^{th} interval, then the slope of the i^{th} line segment is:

$$a_i = \frac{x_{(i)} - x_{(i-1)}}{c_i - c_{i-1}}$$

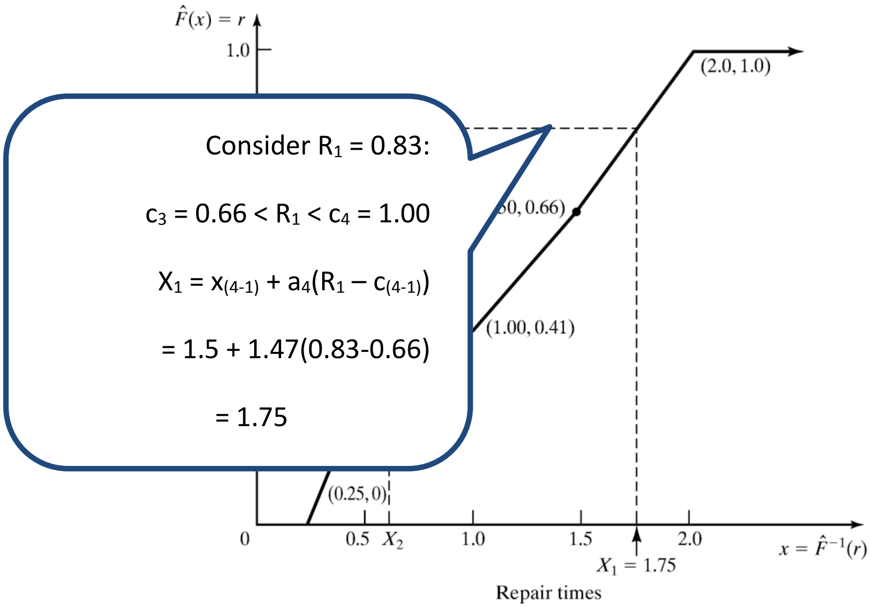
c_i cumulative probability of the first i intervals

- The inverse CDf is given by:

$$\boxed{\begin{array}{l} X = \hat{F}^{-1}(R) = x_{(i-1)} + a_i\left(R - c_{i-1}\right) \\ \text{when } \; c_{i-1} < R \le c_i \end{array}}$$

Example: Suppose the data collected for 100 broken-widget repair times are:

i	*Interval (Hours)*	*Frequency*	*Relative Frequency*	*Cumulative Frequency, c_i*	*Slope, a_i*
1	$0.25 \le x \le 0.5$	31	0.31	0.31	0.81
2	$0.5 \le x \le 1.0$	10	0.10	0.41	5.0
3	$1.0 \le x \le 1.5$	25	0.25	0.66	2.0
4	$1.5 \le x \le 2.0$	34	0.34	1.00	1.47



- Problems with empirical distributions
  - The data in the previous example is restricted in the range $0.25 \le X \le 2.0$
  - The underlying distribution might have a wider range
  - Thus, try to find a theoretical distribution

- Hints for building empirical distribution based on frequency tables
  - It is recommended to use relatively short intervals.
    - Number of bin increase.
  - This will result in a more accurate estimate.

Discrete Distribution

All discrete distributions can be generated via inverse-transform technique, either numerically through a table-lookup procedure, or algebraically using a formula

Examples of application:

1. Empirical.
2. Discrete uniform.
3. Gamma.

Example: (An Empirical Discrete Distribution)Suppose the number of shipments, x, on the loading dock of a company is either 0, 1, or 2

• Data - Probability distribution:

x	*p(x)*	*F(x)*
0	0.50	0.50
1	0.30	0.80
2	0.20	1.00

- The inverse-transform technique as table-lookup procedure :

$$F(x_{i-1}) = r_{i-1} < R \le r_i = F(x_i)$$

Set X1 = xi

Method - Given R, the generation scheme becomes:

$$x = \begin{cases} 0, & R \le 0.5 \\ 1, & 0.5 < R \le 0.8 \\ 2, & 0.8 < R \le 1.0 \end{cases}$$

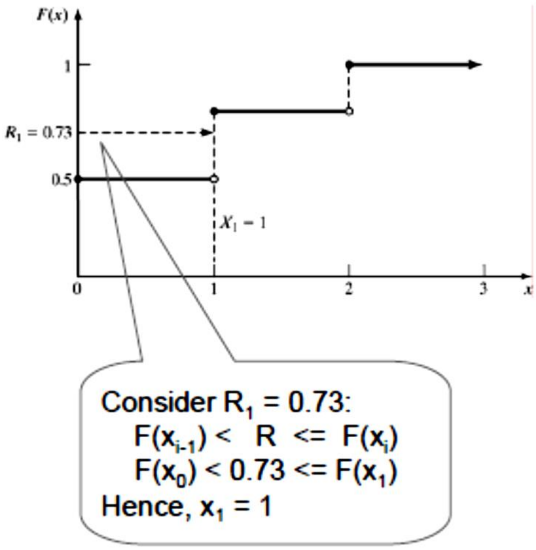


Table for generating the discrete variate X

i	Input r_i	Output x_i
1	0.5	0
2	0.8	1
3	1.0	2

<u>Problems with Inverse-Transform Approach</u>

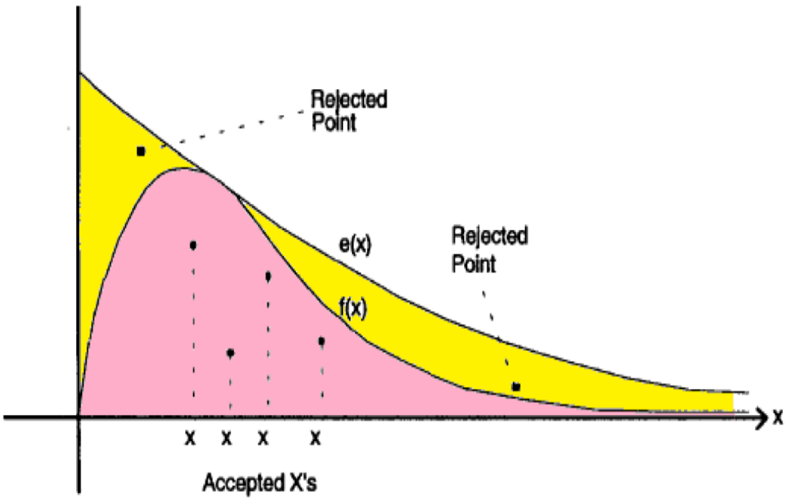
Must invert CDF, which may be difficult (numerical methods)

May not be the fastest or simplest approach for a given distribution

2- Acceptance-Rejection technique

The acceptance-rejection method is most easily explained for a continuous distribution.

Suppose that we wish to sample from the distribution with density f(x) but that it is difficult to do so by the inverse transform method. Suppose now that the following three assumptions hold:

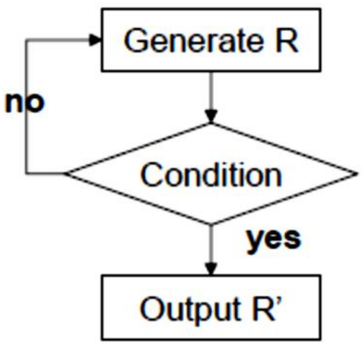


1. There is another function e(x) that dominates f(x) in the sense that e(x) > f'(x) for all x.
2. 2. It is possible to generate points uniformly scattered under the graph of e(x) (above the x-axis). Denote the coordinates of such a typical point by (X, Y).
3. If the graph off (x) is drawn on the same diagram, the point (X, Y) will be above or below it according as Y > f(X) or Y I f(X).

The acceptance-rejection method works by generating such points (X, Y) and returning the X coordinate as the generated X value, but only if the point lies under the graph off (x) [i.e., only if Y I f(X)]. Intuitively it is clear that the density of X is proportional to the height o f f , so that X has the correct density.

- Used when the inverses transform fails or is too complicated.
- It can be applied to a continuous or discrete density functions.
- The objective is to generate a variate X having a CDF F(x) and PDF f(x).
- Illustration: To generate random variates, X ~ U(1/4, 1):

Procedures:
Step 1: Generate R ~ U[0,1]
Step 2a: If R >= ¼, accept X=R.
Step 2b: If R < ¼, reject R, return to: Step 1



- R does not have the desired distribution, but R conditioned (R') on the event {R ≥ ¼} does.
- Efficiency: Depends heavily on the ability to minimize the number of rejections.

Poisson Distribution

A Poisson random variable, N, with mean α > 0 has pmf

$$p(n) = P(N = n) = \frac{e^{-\alpha}\alpha^n}{n!}, \quad n = 0,1,\ldots$$

Recall that inter-arrival times *A1, A2,* ... of successive customers are exponentially distributed with rate α (i.e., α is the mean arrivals per unit time) Then,

$$A_1 + A_2 + \ldots + A_x \le 1 < A_1 + A_2 + \ldots + A_x + A_{x+1}$$

Relation N=n states that there were exactly n arrivals during one unit of time Second relation states that the n^{th} arrival occurred before time 1 while the *(n+1)* arrival occurred after time 1. Clearly these two are equivalent.

Procedure of generating a Poisson random variate N is as follows:

Step 1 set n = 0, P =1

Step 2 generate a random number Rn+1
And replace P by P * Rn+1

Step 3 if P < $e^{-\lambda}$, then accept, otherwise, reject the current n,
increase n by 1 and return to step 2

How many random numbers are required to generate one Poisson variate *N?*

- Answer: If *N=n,* then *n+1* random numbers are required. So the average # is: E(N+1)=α+1.

Example: Generate **three** Poisson variates with mean α =0.2

• exp(-0.2) = 0.8187

- Variate 1

• Step 1: Set n=0, P=1
• Step 2: R1 = 0.4357, P = 1 x 0.4357
• Step 3: Since P = 0.4357 < exp(- 0.2), accept N = 0

- Variate 2

• Step 1: Set n=0, P=1
• Step 2: R1 = 0.4146, P = 1 x 0.4146
• Step 3: Since P = 0.4146 < exp(-0.2), accept N = 0

- **Variate 3**

• Step 1: Set n=0, P=1

• Step 2: R1 = 0.8353, P = 1 x 0.8353

• Step 3: Since P= 0.8353 > exp(-0.2), reject n=0 and return to Step 2 with n=1

• Step 2: R2 = 0.9952, P = 0.8353 x 0.9952 = 0.8313

• Step 3: Since P= 0.8313 > exp(-0.2), reject n=1 and return to Step 2 with n=2

• Step 2: R3 = 0.8004, P = 0.8313 x 0.8004 = 0.6654

• Step 3: Since P = 0.6654 < exp(-0.2), accept N = 2

The calculations for the generation of these Poisson random variates is summarized as:

n	R_{n+1}	P	Accept/Reject	Result
0	0.4357	0.4357	Accept	N=0
0	0.4146	0.4146	Accept	N=0
0	0.8353	0.8353	Reject	
1	0.9952	0.8313	Reject	
2	0.8004	0.6654	Accept	N=2

Example: by using Acceptance-Rejection Technique generate 4 value of a Poisson random variate with mean (λ=2) and ($e^{-0.2}$ = 0.8187) use the random number:

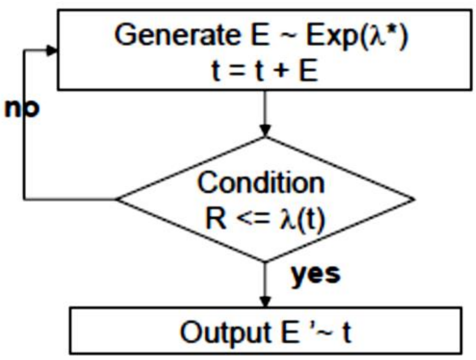
0.25	0.01	0.93	0.70	0.66	0.74	0.79	0.47	0.68	0.18	0.88	0.07	0.99	0.51	0.04	0.01
0.43	0.60	0.59	0.55	0.64	0.10	0.61	0.22	0.85	0.42	0.01	0.98	0.05	0.20	0.11	0.23
0.68	0.41	0.96	0.48	0.11	0.59	0.11	0.10								

$$step\ 1: set\ n = 0, P = 1$$
$$step\ 2: R_1 = 0.25, \quad P \to P \times R_1, \quad P = 1 \times 0.25 = 0.25$$
$$step\ 3: Sice\ P = 0.25 < e^{-\lambda} = 0.8187, accept\ N = 0.$$
$$step\ 1: set\ n = 0, P = 1$$
$$step\ 2: R_1 = 0.01, \quad P \to P \times R_1, \quad P = 1 \times 0.01 = 0.01$$
$$step\ 3: Sice\ P = 0.01 < e^{-\lambda} = 0.8187, accept\ N = 0.$$
$$step\ 1: set\ n = 0, P = 1$$
$$step\ 2: R_1 = 0.93, \quad P \to P \times R_1, \quad P = 1 \times 0.93 = 0.93$$
$$step\ 3: Sice\ P = 0.93 \geq e^{-\lambda} = 0.8187, reject\ n = 0, and\ return\ to\ step\ 2\ with\ n = 1.$$
$$step\ 2: R_2 = 0.70, \quad P \to P \times R_2, \quad P = 0.93 \times 0.70 = 0.65$$
$$step\ 3: Sice\ P = 0.65 < e^{-\lambda} = 0.8187, accept\ N = 1.$$
$$step\ 1: set\ n = 0, P = 1$$
$$step\ 2: R_1 = 0.66, \quad P \to P \times R_1, \quad P = 1 \times 0.66 = 0.66$$
$$step\ 3: Sice\ P = 0.66 < e^{-\lambda} = 0.8187, accept\ N = 0.$$

If 4 value need is: [0,0,1,0]

Non-stationary Poisson Process (NSPP)

- Non-stationary Poisson Process (NSPP): a Possion arrival process with an arrival rate that varies with time
- Idea behind thinning:
  - Generate a stationary Poisson arrival process at the fastest rate, $\lambda^* = \max \lambda(t)$
  - But "accept" only a portion of arrivals, thinning out just enough to get the desired time-varying rate



Generic algorithm that generates T_i as the time of i^{th} arrival

- Step1: Let $\lambda^* = \max \lambda(t)$ be the maximum arrival rate function and set $t=0$ and $i=1$
- Step 2: Generate E from the exponential distribution with rate λ^* and let $t=t+E$ (this is the arrival time of the stationary Poisson process)
- Step3: Generate random number R from the $U(0,1)$ distribution. If $R \leq \lambda(t)/\lambda^*$, then $T_i = t$ and $i=i+1$

Example: Generate a random variate for a NSPP

Data: Arrival Rates

t (min)	*Mean Time Between Arrivals (min)*	*Arrival Rate $\lambda(t)$ (#/min)*
0	15	1/15
60	12	1/12
120	7	1/7
180	5	1/5
240	8	1/8
300	10	1/10
360	15	1/15
420	20	1/20
480	20	1/20

Procedures:

Step 1. $\lambda^* = \max \lambda(t) = 1/5$, $t = 0$ and $i = 1$. 3

Step 2. For random number $R = 0.2130$,

$$E = -5\ln(0.213) = 13.13$$

3- Special Properties

Variate generate techniques that are based on features of particular family of probability distributions, rather than being general-purpose techniques like the inverse-transform or acceptance-rejection techniques.

For example:

- Direct Transformation for *normal* and *lognormal* distributions
- Convolution
- Beta distribution (from gamma distribution)

Direct Transformation

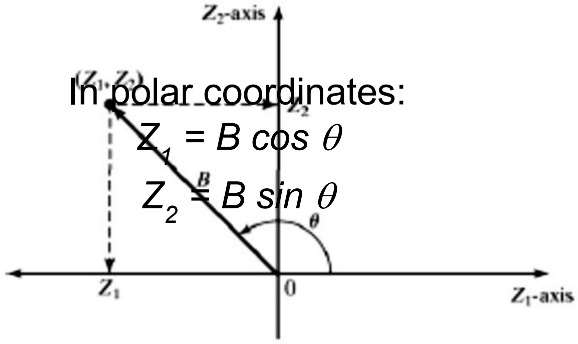
Approach for *Normal(0,1)*:

- Consider two standard normal random variables, Z_1 and Z_2, plotted as a point in the plane:

In polar coordinates:

$Z_1 = B \cos \theta$

$Z_2 = B \sin \theta$



- $B^2 = Z^2{}_1 + Z^2{}_2$ ~ chi-square distribution with *2* degrees of freedom that is equivalent to *Exp(l = 2)*. Hence, $B=(-2\text{Ln } R)^{1/2}$
- The radius *B* and angle θ are mutually independent.

$$\boxed{\begin{aligned} Z_1 &= (-2\ln R)^{1/2}\cos(2\pi R_2) \\ Z_2 &= (-2\ln R)^{1/2}\sin(2\pi R_2) \end{aligned}}$$

Approach for *Normal* (μ,σ^2):

- Generate $Z_i \sim N(0,1)$

$$\boxed{X_i = \mu + \sigma Z_i}$$

Approach for lognormal(μ,σ^2):

- Generate $X \sim N(\mu,\sigma^2)$

$$\boxed{Y_i = e^{X_i}}$$

Example:

Given that $R_1 = 0.1758$ and $R_2 = 0.1489$.

Two standard normal random variates are generated as follows:

$Z_1 = [-2\ln(0.1758)]^{1/2} \cos(2p\ 0.1489) = 1.11$

$Z_2 = [-2\ln(0.1758)]^{1/2} \sin(2p\ \ 0.1489) = 1.50$

To transform the two standard normal variates into normal variates with mean m = 10 and variance $\sigma^2 = 4$,compute

$$X_1 = 10 + 2(1.11) = 12.22$$

$$X_2 = 10 + 2(1.50) = 13.00$$

Convolution

Applicable to situation where the random variable of interest can be expressed as a sum of other random variables that are IID (independent identical distributed)

$$X=Y1+Y2+Y3.... +Yn$$

Idea: Generate Y1…. Yn and add these up to calculate X

Beta distribution (from gamma distribution)

There are many relationships among probability distributions that can be exploited for random-variate Generation.

Suppose that X_1 has a gamma distribution with shape parameter $ß_1$ and scale parameter $ø_1 = 1/ ß_1$, while X_2 has a giunma distribution with shape parameter $ß_2$ and scale parameter $ø_2= 1/ ß_2$, and that these two random variables are independent Then:

$$Y=x_1/(x_1+x_2)$$

has a beta distribution with parameters $ß_1$ and $ß_2$ on the interval (0, I). If, instead, we want Y to be defined on the interval (a, b), then set

$$Y=a+\left[(b-a) \quad (x_1+x_2)\right]$$

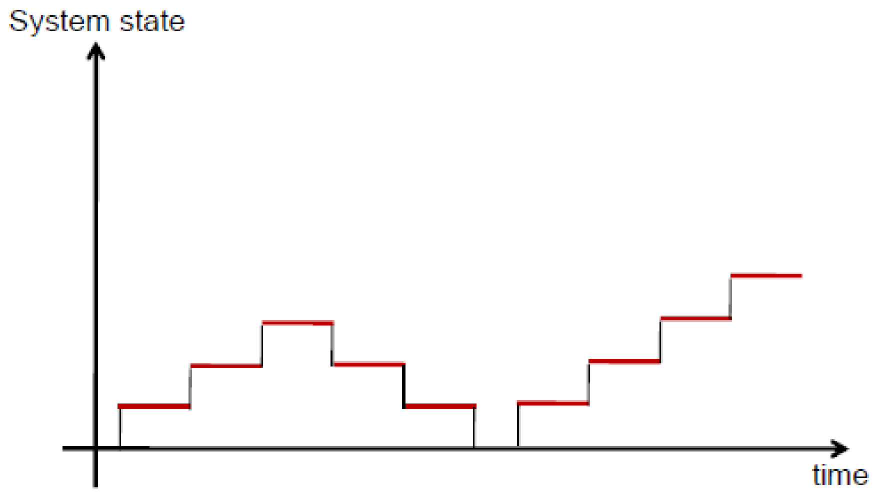
Thus, using the acceptance-rejection technique for gamma variates defined in the previous section, we can generate beta variates, with two gamma variates required for each b

Simulation: The process of designing a mathematical or logical model of a system and then conducting computer-based experiments with the model to describe, explain, and predict the behavior of the system.

Framework for modeling systems by discrete-event simulation

• A system is modeled in terms of its state at each point in time

• This is appropriate for systems where changes occur only at discrete points in time



Concepts in Discrete-Event Simulation

System: a collection of entities that interact together over time to accomplish one or more goals, (e.g., bank, network, production system.)

Model: an abstract representation of a system, usually containing structural, logical, or mathematical relationships that describe the system.

System state: a collection of variables that contain all the information necessary to describe the system at any time.

Entity: an object in the system that requires explicit representation in the model (e.g., people, machines, server, customer).

Attributes: the properties of an entity (e.g. length of a packet, capacity of machine, the priority of a customer).

List: a collection of associated entities ordered in some logical fashion such as customers in a waiting line ordered by some rule, e.g. FIFO, LIFO, priority)

Event: an instantaneous occurrence that changes the state of a system such as (an arrival of anew customer)

Event notice: a record of an event to occur at the current or some future time, along with any associated data necessary to execute the event the record includes the event type and the event time..

Event list: a list of event notices for future events, ordered by time of occurrence; known as the Future Event List (FEL)

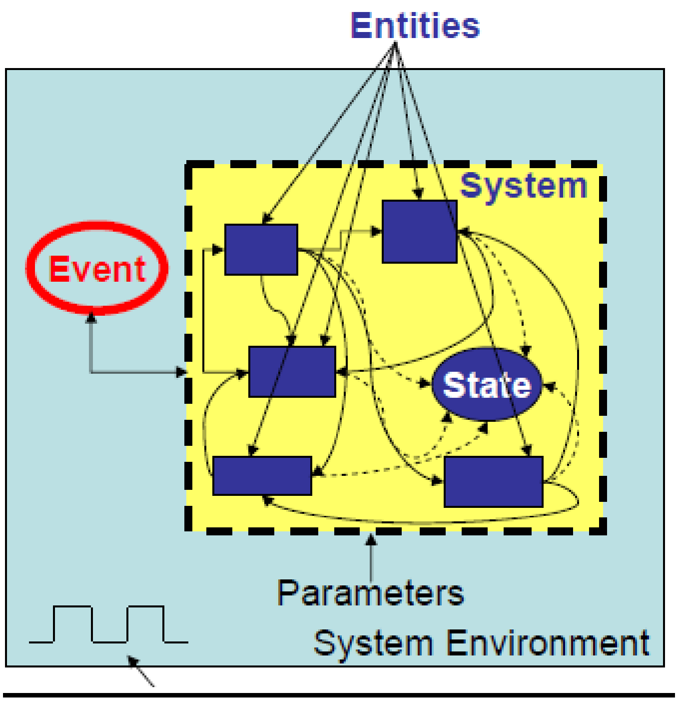
Activity (un conditional wait): duration of time of specified length, e.g. service time, or arrival time, which is known when it begins. The duration of an activity can be specified as:

• Deterministic – exactly 5 time units,

• Statistical – Random draw from {2, 5, 7}

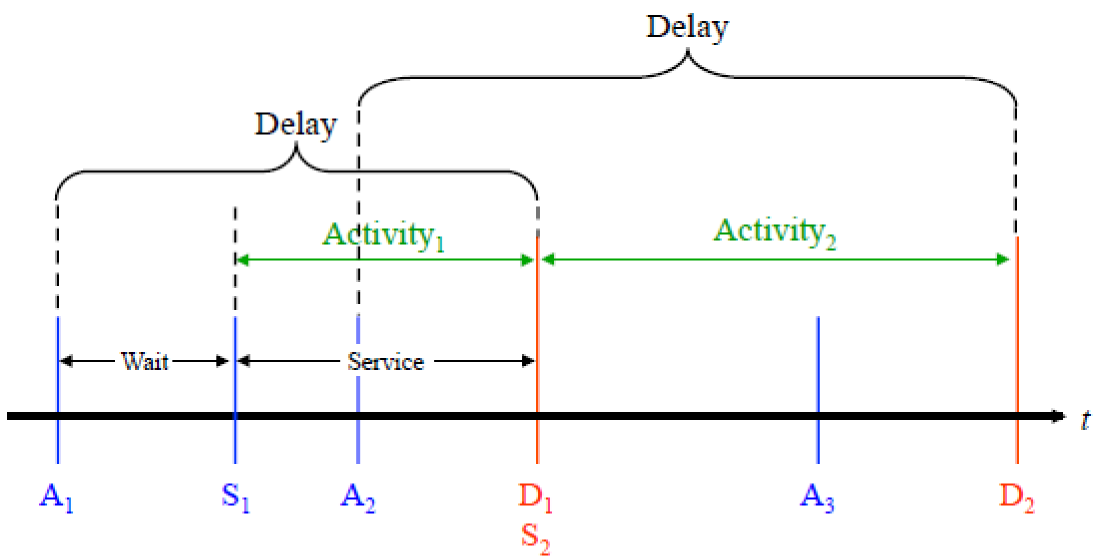
Delay (conditional wait): a duration of time of unspecified indefinite length, which is not known until it ends e.g. (customer's delay in waiting line depends on the number and service times of other customers).

Clock: a variable representing the simulated time.

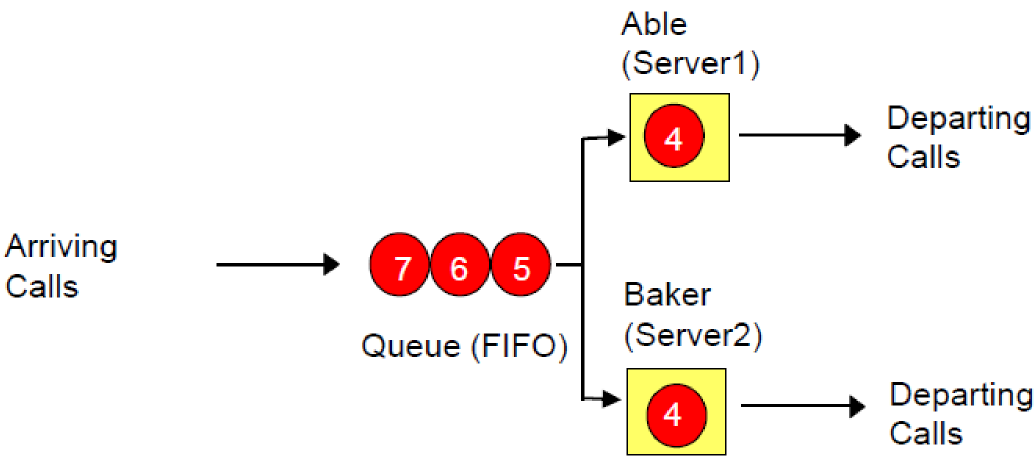


Clock

Activity vs. Delay



Able Baker (Call Center) Example: Two parallel servers



Problem Statement:

A call center has two technical support people: Able and Baker.
Able is better than Baker!
a) She has lower servicing times
b) She gets the first customer if both are idle.
What is given in the problem statement?
a) Inter-arrival time distribution for the customers
b) Service time distribution for Able
c) Service time distribution for Baker

System state

- LQ(t): the number of callers waiting to serve
- LA(t):0 or 1 indicate Able is idle or busy
- LB(t):0 or 1 indicate Baker is idle or busy

Entities

- Caller

Events

- Arrival event
- service completion by Able or Baker

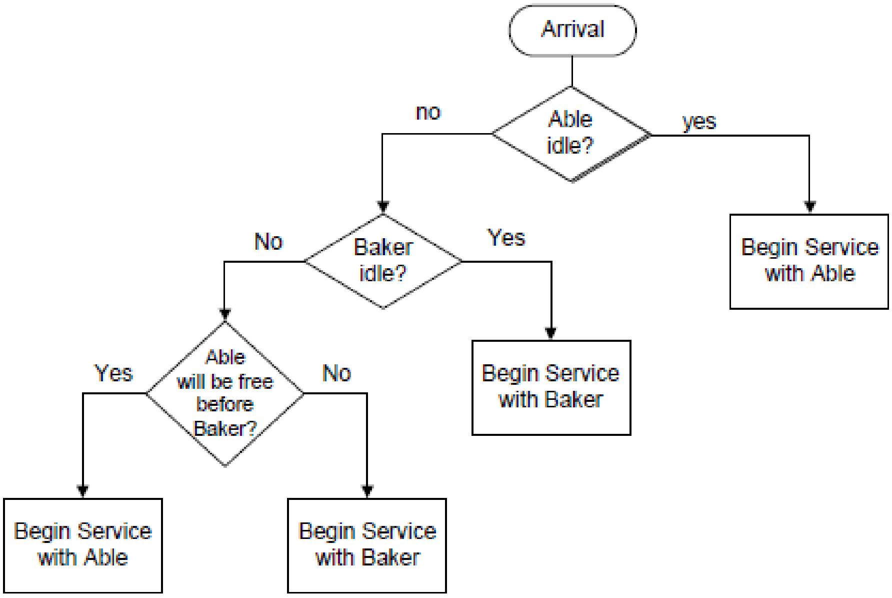
Activities

- Service time by Able/Baker and Inter-arrival time

Delay

A caller wait in queue until Able or Baker becomes free.

Able Baker Example Diagram



The two bank tellers example: revisited

- System state:
  - Q(t) – number of customers waiting in the queue at time t
  - S1(t) – state of teller 1: busy/idle
  - S2(t) – state of teller 2: busy/idle
- Entities
  - Teller 1
  - Teller 2
  - Arrivals generator (it is actually external to the system)
- Activities
  - Teller 1 service time
  - Teller 2 service time
  - Inter-arrival time

the duration of the activity not affected by the occurrence of other events; it is based on the modeler's specifications: unconditional wait

Determined by the system conditions (not specified by the modeler ahead of time): conditional wait

- Delay
  - Customer waiting time until service started or completed

7

Event Scheduling

Clock	System state	Entities and Attributes	Set1	Set2	FEL	Cumulative statistics and counters
t	(x,y,z,…)		Queue membership		(3,t1) (1,t2)	
t1	(x-1,y,z,…)				(1,t2)	
.....						

For our example:

state variable x (e.g. customers waiting)

y (e.g. teller 1 busy)

z = (e.g. teller 2 busy)

Type 1 event: arrival
Type 2 event: departure teller1
Type 3 event: departure teller 2

Future event list: {(3,t1),(1,t2)}:

type 3 event will occur at t=t1 (e.g. teller 2 service completion at t1)

type 1 event will occur at t=t2 (e.g. customer arrival at t2)

Event-scheduling/Time-advance algorithm

Event-scheduling/Time-advance algorithm

- Step 1: Remove the event notice for the imminent event from FEL
  - event $(3, t_1)$ in the example
- Step 2: Advance Clock to imminent event time
  - Set clock = t_1
- Step 3: Execute imminent event
  - update system state
  - change entity attributes
  - set membership as needed
- Step 4: Generate future events and place their event notices on FEL
  - Event $(4, t^*)$
- Step 5: Update statistics and counters

Old system snapshot at time t

Clock	State	…	Future event list
t	(5,1,6)		$(3,t_1)$ – Type 3 event to occur at t_1
			$(1,t_2)$ – Type 1 event to occur at t_2
			$(1,t_3)$ – Type 1 event to occur at t_3
			…
			$(2,t_n)$ – Type 2 event to occur at t_n

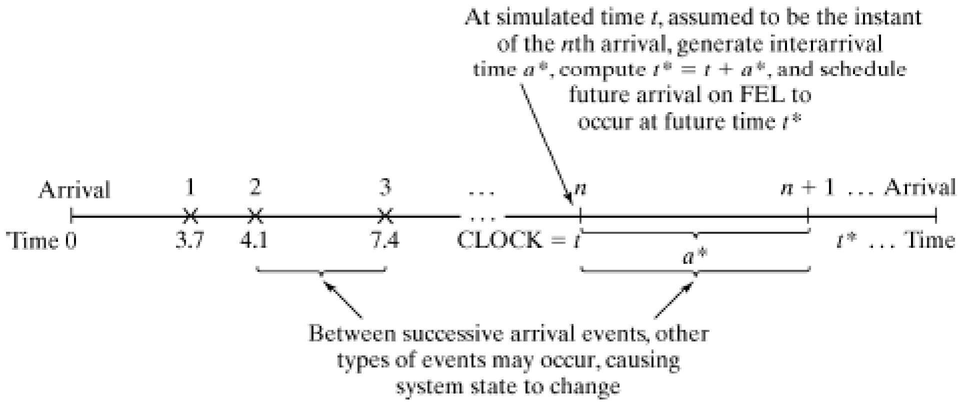
New system snapshot at time t_1

Clock	State	…	Future event list
t_1	(5,1,5)		$(1,t_2)$ – Type 1 event to occur at t_2
			$(4,t^*)$ – Type 4 event to occur at t^*
			$(1,t_3)$ – Type 1 event to occur at t_3
			…
			$(2,t_n)$ – Type 2 event to occur at t_n

Generation arrival of a customer

- At t=0 first arrival is generated and scheduled

- When the clock is advanced to the time of the first arrival, a second arrival is generated.
- Generate an inter arrival time a^*
- Calculate $t^* = t + a^*$.
- Place event notice at t^* on the FEL.



Stopping time of simulation

Every simulation must have a stopping event, called T_E, which defines how long the simulation will run. There are generally two ways to stop a simulation:

1. At time 0, schedule a stop simulation event at a specified Future time T_E. Thus, before simulating, it is known that the Simulation will run over the time interval $[0, T_E]$.

2. Run length T_E is determined by the simulation itself. T_E is the time of occurrence of some specified Event *E. Examples:* T_E is the time of the 100th service_completion at a certain service center. T_E is the time of Breakdown of a complex system

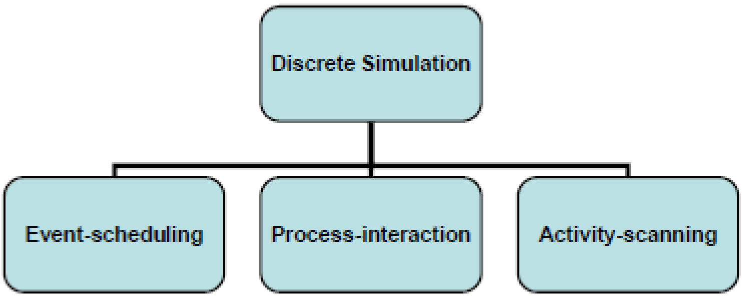
For example: banks opens at 8:30 a.m. (time 0)with no customers presents and 8 of 11 teller working (initial conditions) & closes at 4:30

p.m.

Time T_E = [480 minutes]

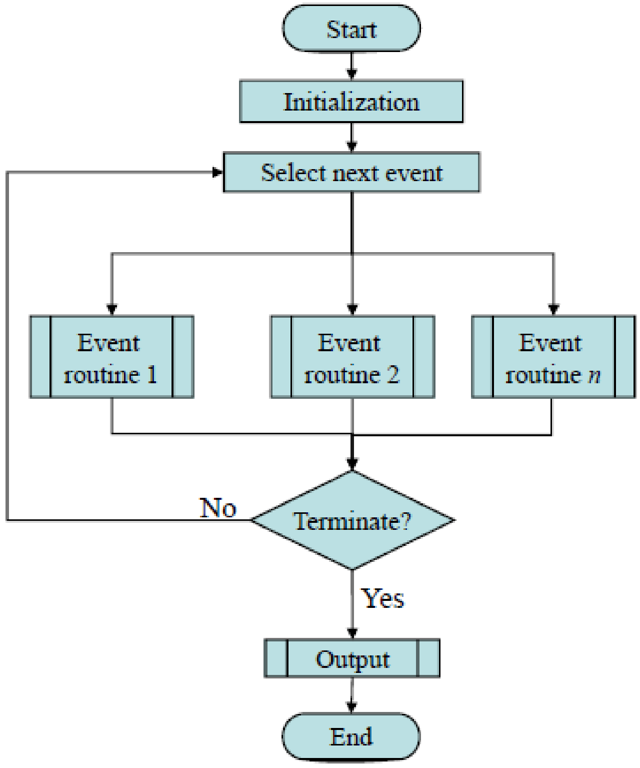
World View

- A world view is an orientation for the model developer
- Manual Simulation typically support some world views
- Here, only world views for discrete simulations.



Event-scheduling

- Focus on events
- Identify the entities and their attributes
- Identify the attributes of the system
- Define what causes a change in system state
- Write a routine to execute for each event
- Variable time advance



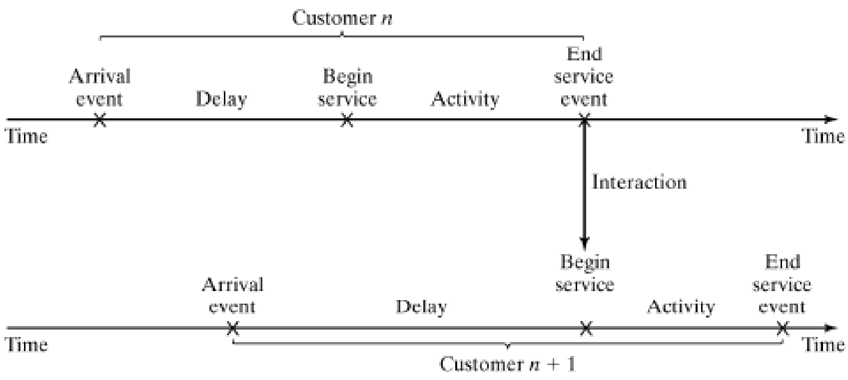
Process-Interaction

• A process is the lifecycle of one entity, which consists of various events and activities (like process in o.s).

Processes interact, e.g., one process has to wait in a queue because the resource it needs is busy with another process

• A process is a time-sequenced list of events, activities and delays, including demands for resource, that define the life cycle of one entity as it moves through a system

• Variable time advance



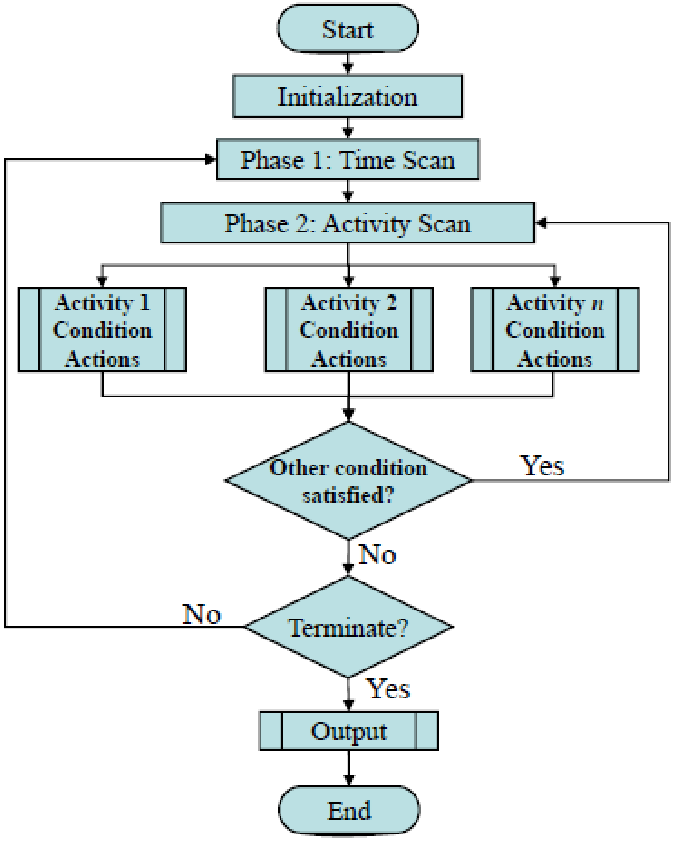
Activity-scanning

- Modeler concentrates on activities of a model and those conditions that allow an activity to begin Initialization.
- At each clock advance, the conditions for each activity are checked, and, if the conditions are true, then the corresponding activity begins
- Fix time advance
- It is suitable for small system
- It is very fast

Disadvantage: The repeated scanning to discover whether an activity can begin results in slow runtime.

Improvement: Three-phase approach

Combination of event scheduling with activity scanning



In the three-phase approach, events are considered to be activity duration-zero time units. With this definition, activities are divided into two categories called B and C.

- **B activities**: activities bound to occur; all primary events and Unconditional activities

- **C activities**: activities or events that are conditional upon certain Conditions being true.

The simulation proceeds with repeated execution of the 3 phases until it is completed:

Phase A Remove the imminent event from the FEL and advance the

Clock to its event time. Remove from the FEL any other events that have the same event time.

Phase B Execute all B-type events that were removed from the FEL.

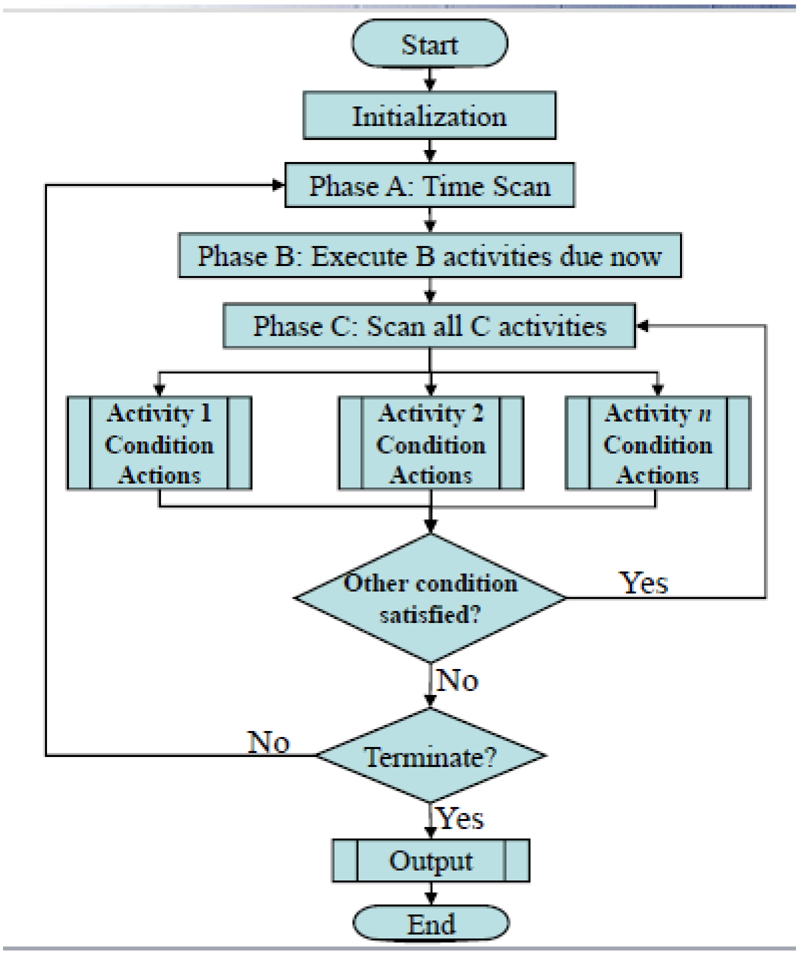
(this could free a number of resources or otherwise change system state.)

Phase C Scan the conditions that trigger each C-type activity and activate any whose conditions are met. Rescan until no additional

C-type activities can begin and no event occurs.

The three-phase approach improves the execution efficiency of

the activity scanning method



Event Scheduling Example (Grocery Center)

System state at time t is given by [LQ(t), LS(t)]

• LQ(t) = Number of customers in the waiting line at t

• LS(t) = Number of customers being served at t (0 or 1)

 Entities

• Server and customers are not explicitly modeled

 Events

• Arrival (A)

• Departure (D)

• Stopping event (E)

□ Event notices

• (A, t) arrival event at future time t

• (D, t) departure event at future time t

• (E, t) simulation stop at future time t

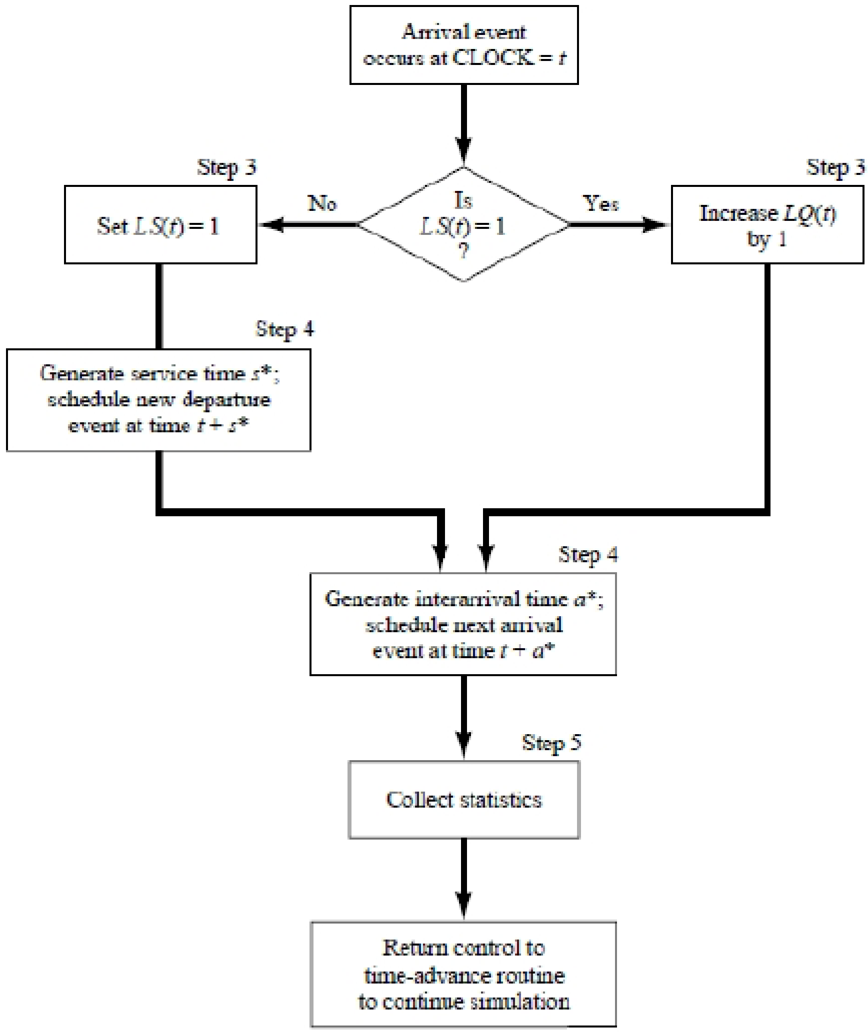
□ Activities

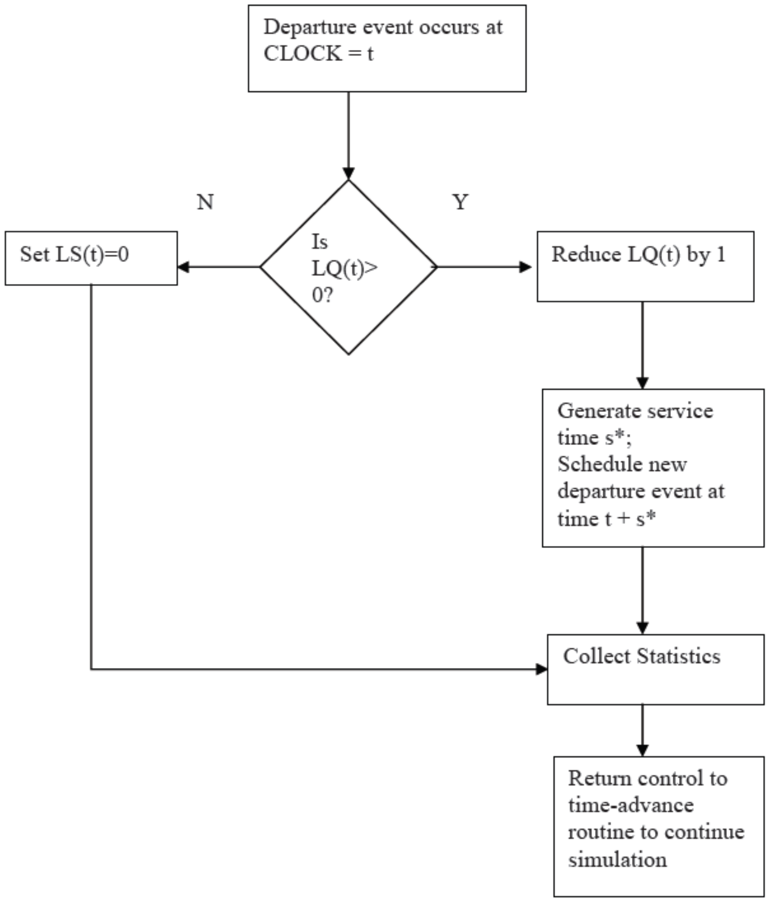
• Inter arrival time.

• Service time

□ Delay

• Customer time spent in waiting line


Arrival event occurs at CLOCK = t
Step 3
Set LS(t) = 1
No
Is LS(t) = 1 ?
Yes
Step 3
Increase LQ(t) by 1
Step 4
Generate service time s*; schedule new departure event at time t + s*
Step 4
Generate interarrival time a*; schedule next arrival event at time t + a*
Step 5
Collect statistics
Return control to time-advance routine to continue simulation



Simulation table for checkout counter.

Clock	LQ(t)	LS(t)	FEL	Comment	B	MQ
0	0	1	(D,4) (A,8) (E,60)	First A occurs (a* = 8) schedule next A (s* = 4) schedule next D	0	0
4	0	0	(A,8) (E,60)	First D occurs;(D,4)	4	0
8	0	1	(D,9) (A,14) (E,60)	Second A occurs;(A,8) (a* = 6) schedule next A (s* = 1) schedule next D	4	0
9	0	0	(A,14) (E,60)	Second D occurs;(D,9)	5	0

Example 2 The DumpTruck Problem

・Six dump trucks are used to haul coal from the entrance of a small mine to the railroad

• Each truck is loaded by one of two loaders

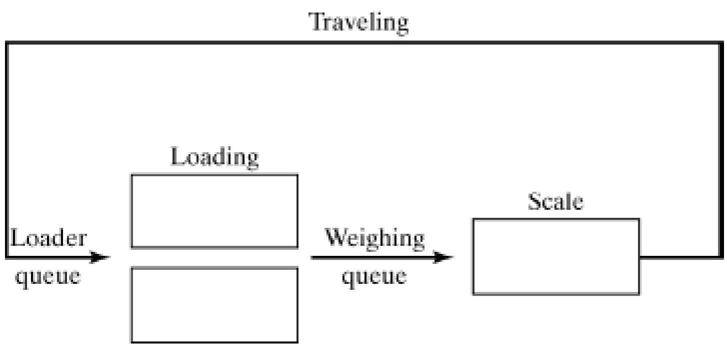
• After loading, the truck immediately moves to the scale, to be weighed

• Loader and Scale have a first-come-first-serve (FCFS) queue

• The travel time from loader to scale is negligible

• After being weighed, a truck begins a travel time, after wards unloads

The coal and returns to the loader queue.



- **System state [LQ(t), L(t), WQ(t), W(t)]**
  - LQ(t) = number of trucks in the loader queue $\in\{0,1,2,...\}$
  - L(t) = number of trucks being loaded $\in\{0,1,2\}$
  - WQ(t) = number of trucks in weigh queue $\in\{0,1,2,...\}$
  - W(t) = number of trucks being weighed $\in\{0,1\}$
- **Event notices**
  - (ALQ, t, DTi) dump truck i arrives at loader queue (ALQ) at time t
  - (EL, t, DTi) dump truck i ends loading (EL) at time t
  - (EW, t, DTi) dump truck i ends weighing (EW) at time t
- **Entities**
  - The six dump trucks DT1, DT2, ..., DT6
- **Lists**
  - Loader queue – Trucks waiting to begin loading, FCFS
  - Weigh queue – Truck waiting to be weighed, FCFS
- **Activities**
  - Loading – Loading time
  - Weighing – Weighing time
  - Travel – Travel time
- **Delays**
  - Delay at loader queue
  - Delay at scale

- **Initialization**
  - It is assumed that five trucks are at the loader and one is at the scale at time 0
- **Activity times**
  - Loading time: 10, 5, 5, 10, 15, 10, 10
  - Weighing time: 12, 12, 12, 16, 12, 16
  - Travel time: 60, 100, 40, 40 80
- **Statistics**
  - BL – Total busy time of both loaders
  - BS – Total busy time of the scale

Both loaders are busy!

	System State				Lists			Statistics	
Clock	LQ(t)	L(t)	WQ(t)	W(t)	Loader Queue	Weigh Queue	Future Event List	BL	BS
0	3	2	0	1	DT4, DT5, DT6		(EL,5,DT3) (EL,10,DT2) (EW,12,DT1)	0	0
5	2	2	1	1	DT5, DT6	DT3	(EL,10,DT2) (EL,5+5,DT4) (EW,12,DT1)	10	5
10	1	2	2	1	DT6	DT3, DT2	(EL,10,DT4) (EW,12,DT1) (EL,10+10,DT5)	20	10
10	0	2	3	1		DT3, DT2, DT4	(EW,12,DT1) (EL,20,DT5) (EL,10+15,DT6)	20	10
12	0	2	2	1		DT2, DT4	(EL,20,DT5) (EW,12+12,DT3) (EL,25,DT6) (ALQ,12+60,DT1)	24	12
20	0	1	3	1		DT2, DT4, DT5	(EW,24,DT3) (EL,25,DT6) (ALQ,72,DT1)	40	20
24	0	1	2	1		DT4, DT5	(EL,25,DT6) (EW,24+12,DT2) (ALQ,72,DT1) (ALQ,24+100,DT3)	44	24

Computing Mean Response Time

Suppose the system analyst desires to estimate the mean response time and mean proportion of customers who spend 4 or more minutes in the system the above mentioned model has to be modified.

o **Entities** (Ci, t), representing customer Ci who arrived at time t.

o **Event notices** *(A*, *t*, Ci), the arrival of customer Ci at future time t

(D, f, Cj), the departure of customer Cj at future time t.

o **Set** "CHECKOUT LINE," the set of all customers currently at the checkout Counter (being served or waiting to be served), ordered by time of arrival

Three new statistics are collected: **S, the sum of customer's response times for all customers who have departed by the current time; F, the total number of customers who spend 4 or more minutes at the checkout counter; ND the total number of departures up to the current simulation time.**

- These three cumulative statistics are updated whenever the departure event occurs.
- The simulation table is given below

System state					Cumulative statistics		
Clock	LQ(t)	LS(t)	Checkout line	FEL	S	N_D	F
0	0	1	(C 1,0)	(D,4,C1) (A,8,C2) (E,60)	0	0	0
4	0	0		(A,8,C2) (E,60)	4	1	1
8	0	0	(C2,8)	(D,9,C2) (A,14,C3) (E,60)	4	1	1
9	0	0		(A,14,C3) (E,60)	5	2	1

List processing:

The management of a list is called list processing, deals with methods for handling lists of entities and the future event list. The major list processing operations performed on a FEL are Removal of the imminent event, addition of a new event to the List, and occasionally removal of some event (called cancellation of an event).

As the imminent event is usually at the top of the list, its removal

is as efficient as possible. Addition of a new event (and Cancellation of an old event) requires a search of the list. The removal and addition of events from the FEL is illustrated in the example of event scheduling

When event 4 (say, an arrival event) with event time t* is generated at step 4, one possible way to determine its correct position on the FEL is to conduct a top-downs search:

If $t* < t2$, place event 4 at the top of the FEL.

If t2 < $t*$ < t3, place event 4 second on the list.

If $t3$, < $t*$ < t4, place event 4 third on the list.

If tn < t*, event 4 last on the list.

Simulation

A simulation is a tool to evaluate the performance of a system, existing or proposed, under different configurations of interest and over long periods of real time.

Simulation using a table introducing simulation by manually simulating on a table or using computer

Simulation steps using Simulation Table

1. Determine the characteristics of each of the inputs to the simulation (probability distributions).
2. Construct a simulation table (repetition 1).
3. For each repetition i, generate a value for the inputs, and evaluate function, calculating a value of response yi.

Repetition	Inputs					Response
i	x_{i1}	x_{i2}	…	x_{ij}	…	x_{ip}	y_i
1							
2							
. . .							
n							

Static meta data

Dynamic data during simulation run

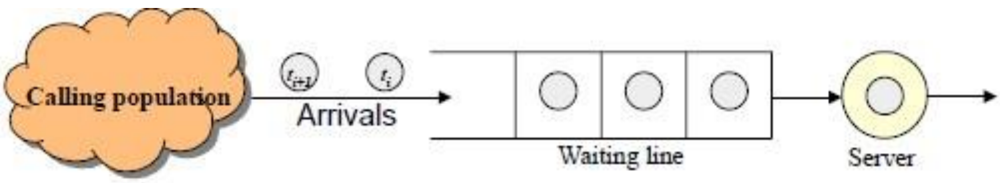
Simulation of Queuing Systems

A queuing system is described by:-

1. Calling population
2. Arrival rate
3. Service mechanism
4. System capacity
4. Queuing discipline

Where there was:-

- Single server -queue
    - Calling population is infinite and Arrival rate does not change
  - Units are served according FIFO
    - Arrivals are defined by the distribution of the time between arrivals (inter-arrival time).
    - Service times are according to a distribution
    - Arrival rate must be less than service rate (stable system) Otherwise waiting line will grow unbounded (unstable system).



Queuing system state:-

It is described by state of server or state of units

- Number of units in the system (in queue or being served)

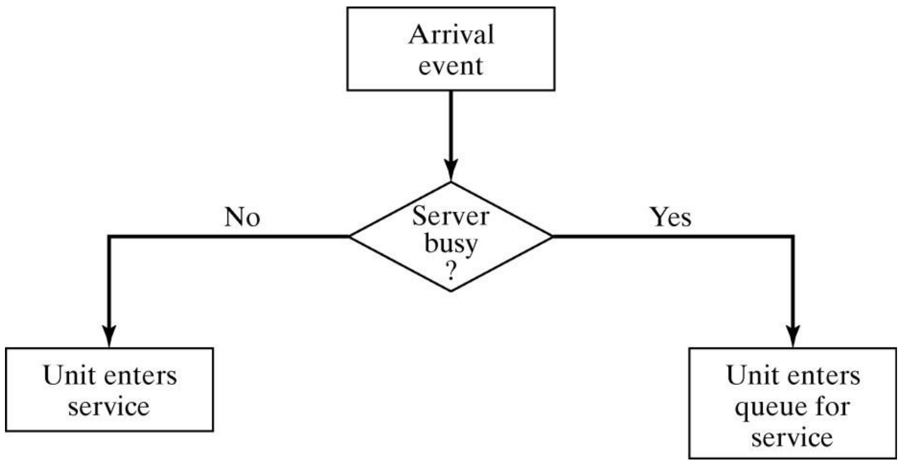
- Status of server (idle,

- busy) Clock of

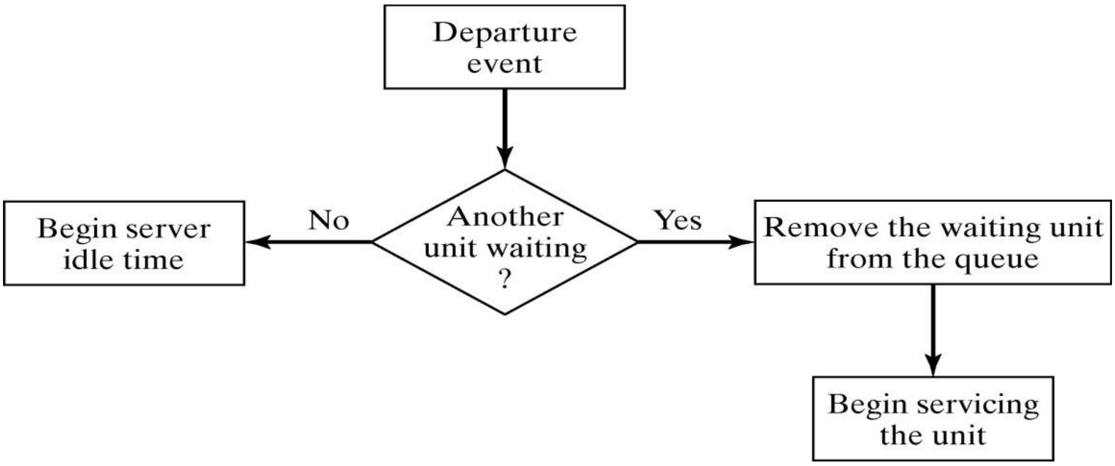
  simulation.

Event: - An instantaneous occurrence that might change the state of the system breakdown that either to be:

1- Arrival of a unit:- If server idle unit gets service, otherwise unit enters queue.

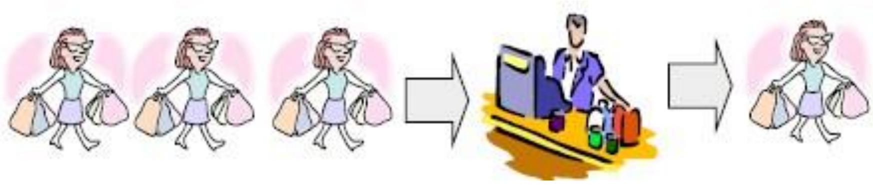


2- Departure of a unit:- If queue is not empty begin servicing next unit, otherwise server will be idle.



		Queue status	
		Not empty	Empty
Server status	Busy	Enter queue	Enter queue
	Idle	Impossible	Enter service

example 1: A Grocery



- One checkout counter
- Arrival time between customers are 1 to 8 minutes (equal probability)
- Service time vary from 1 to 6 (service time table)
- We are going to analysis for 20 customers.

For this system simulation we need to generate table for time between arrivals distribution:-

Time between Arrivals (Minutes)	Probability	Cumulative Probability	Random Numbers Assignment
1	0.125	0.125	0.0000-0.1250
2	0.125	0.250	0.1251-0.2500
3	0.125	0.375	0.2501-0.3750
4	0.125	0.500	0.3751-0.5000
5	0.125	0.625	0.5001-0.6250
6	0.125	0.750	0.6251-0.7500
7	0.125	0.875	0.7501-0.8750
8	0.125	1.000	0.8751-1.0000

And do the same for service time distribution from 1 to 6 minutes :-

Service Time (Minutes)	Probability	Cumulative Probability	Random Numbers Assignment
1	0.10	0.10	0.000-0.100
2	0.20	0.30	0.101-0.300
3	0.30	0.60	0.301-0.600
4	0.25	0.85	0.601-0.850
5	0.10	0.95	0.851-0.950
6	0.05	1.00	0.951-1.000

Now we are need to simulate this system for arrive and serve 20 customer ,

To do simulation manually we are need to generate time between arrival and service time by using last previous tables , and from this appear table of time between arrivals for 20 customer :-

Customer	Random Number Generated	Time between Arrivals (Minutes)
1	0.8879	8
2	0.4065	4
3	0.0799	1
4	0.8029	7
5	0.9915	8
6	0.0381	1
7	0.7456	6
8	0.5014	5
9	0.1786	2
10	0.2481	2
11	0.4027	4
12	0.2708	3
13	0.9065	8
14	0.6057	5
15	0.7184	6
16	0.4033	4
17	0.8510	7
18	0.3966	4
19	0.6224	5
20	0.7386	6

With same manner generate table of service time for 20 customer:-

Customer	Random Number Generated	Sevice Times (Minutes)
1	0.869	5
2	0.878	5
3	0.623	4
4	0.251	2
5	0.074	1
6	0.952	5
7	0.440	3
8	0.496	3
9	0.878	5
10	0.665	4
11	0.954	5
12	0.627	4
13	0.087	1
14	0.628	4
15	0.354	3
16	0.366	3
17	0.763	4
18	0.598	3
19	0.902	5
20	0.302	2

Now generate table for track the customer into the system (simulation table) by calculate the following:-

- End of the service= start of service + service time
- Time in queue= start of service – arrival time
- Time in system =service time + time in queue
- Idle time of server =start of service – end of service

Customer Number	Interarrival Time	Arrival Time	Service Time	Start of service	End of service	Time in queue	Time in system	Idle time of server
1	8	8	5	8	13	0	5	0
2	4	12	5	13	18	1	6	0
3	1	13	4	18	22	5	9	0
4	7	20	2	22	24	2	4	0
5	8	28	1	28	29	0	1	4
6	1	29	5	29	34	0	5	0
7	6	35	3	35	38	0	3	1
8	5	40	3	40	43	0	3	2
9	2	42	5	43	48	1	6	0
10	2	44	4	48	52	4	8	0
11	4	48	5	52	57	4	9	0
12	3	52	4	57	61	5	9	0
13	8	60	1	61	62	1	2	0
14	5	65	4	65	69	0	4	3
15	6	71	3	71	74	0	3	2
16	4	75	3	75	78	0	3	1
17	7	82	4	82	86	0	4	4
18	4	86	3	86	89	0	3	0
19	5	91	5	91	96	0	5	2
20	6	97	2	97	99	0	2	1
Totals	96		71			23	94	20

And from the previous table find the performance measure:-

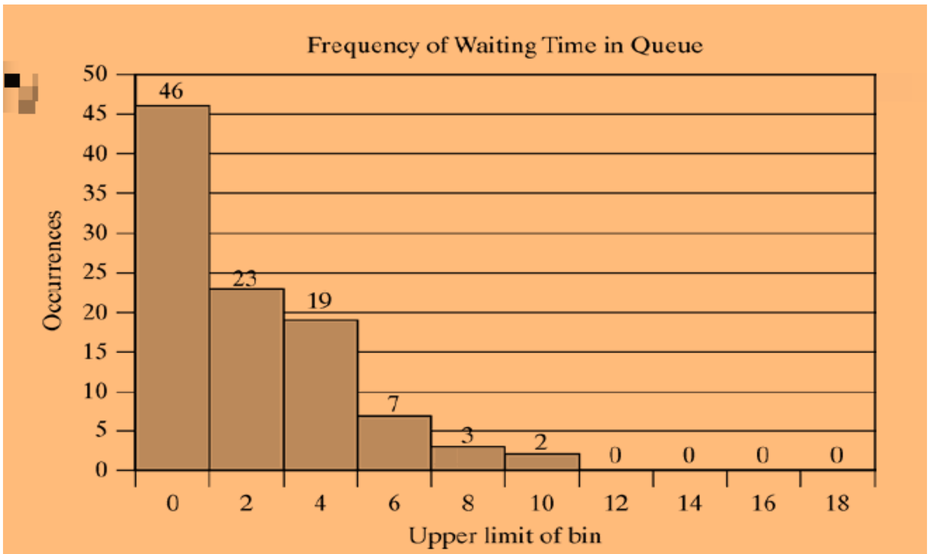
- Average waiting time =total waiting time / total customer

  23/20 = 1.15 minutes
- Probability that a customer has to wait = Number of customer who wait /Number of customers

  8/20= 0.4
- Average service time =total Service time /total customer

  71/20= 3.55 minutes
- Average time between arrivals= total time between arrivals /total customer 96/20 = 4.8 minutes
- Average time a customer spends in system = total time a customer spends in system / total customer

  94/20 = 4.7 minutes



Example 2:- A fast-food restaurant:-

The purpose from this example is to show how to simulate system when there was two or more server, consider there was tow servers that take and deliver the order for cars arrive , and the cars have time between arrivals distribution like :-

Time between Arrivals (Minutes)	Probability	Cumulative Probability	Random Numbers Assignment
1	0.25	0.25	0.000-0.250
2	0.40	0.65	0.251-0.650
3	0.20	0.85	0.651-0.850
4	0.15	1.00	0.851-1.000

Two technical support people (2 server) exists, **Ahmed** more experienced, provides service faster ,and **Baker** newbie, provides service slower, so the distribution of services time between Ahmed and Baker will be :

The service time distribution for Ahmed:

Service Time (Minutes)	Probability	Cumulative Probability	Random Numbers Assignment
2	0.30	0.30	0.000-0.300
3	0.28	0.58	0.301-0.580
4	0.25	0.83	0.581-0.830
5	0.17	1.00	0.831-1.000

And the service time distribution table for Baker will be:

Service Time (Minutes)	Probability	Cumulative Probability	Random Numbers Assignment
3	0.35	0.35	0.000-0.350
4	0.25	0.60	0.351-0.600
5	0.20	0.80	0.601-0.800
6	0.20	1.00	0.801-1.000

Now we need to simulate this system to find it is performance like this tables:

					Ahmad			Bakr			
Cus Tomer No.	Random Number for Arrivals	Time between Arrival	Clock Time of Arrival	Random Number for Service	Time Sevice Start	Service Time	Time Service Ends	Time Sevice Start	Sevice Time	Time Service Ends	Time in Queue
1	0.99284	4	4	0.398244	4	3	7	-	-	-	0
2	0.463490	2	6	0.485853	-	-	-	6	4	10	0
3	0.654933	3	9	0.018479	9	2	11	-	-	-	0
4	0.008010	1	10	0.373430	-	-	-	10	4	14	0
5	0.017539	1	11	0.380226	11	3	14	-	-	-	0
6	0.027118	1	12	0.071695	14	2	16	-	-	-	2
7	0.294307	2	14	0.794733	-	-	-	14	5	19	0
8	0.703278	3	17	0.048615	17	2	19	-	-	-	0
9	0.305171	2	19	0.744830	19	4	23	-	-	-	0
10	0.029153	1	20	0.082780	-	-	-	20	3	23	0
11	0.294875	2	22	0.913267	23	5	28	-	-	-	1
12	0.846545	3	25	0.625406	-	-	-	25	5	30	0
13	0.991276	4	29	0.987133	29	5	34	-	-	-	0
14	0.684252	3	32	0.641578	-	-	-	32	5	37	0
15	0.642370	2	34	0.416842	34	3	37	-	-	-	0
16	0.369203	2	36	0.916370	37	5	42	-	-	-	1
17	0.222240	1	37	0.712437	-	-	-	37	5	42	0
18	0.437991	2	39	0.770969	42	4	46	-	-	-	3
19	0.119146	1	40	0.061159	-	-	-	42	3	45	2
20	0.662990	3	43	0.934648	-	-	-	45	6	51	2
21	0.288916	2	45	0.925251	46	5	51	-	-	-	1
22	0.903758	4	49	0.355554	51	3	54	-	-	-	2
23	0.948593	4	53	0.682907	-	-	-	53	5	58	0
24	0.375286	2	55	0.379748	55	3	58	-	-	-	0
25	0.273955	2	57	0.273077	58	2	60	-	-	-	1
26	0.664870	3	60	0.358811	60	3	63	-	-	-	0
27	0.125086	1	61	0.831475	-	-	-	61	6	67	0
28	0.804005	3	64	0.736537	64	4	68	-	-	-	0
29	0.431573	2	66	0.755743	-	-	-	67	5	72	1
30	0.785686	3	69	0.389873	69	3	72	-	-	-	0
Totals		69				61			56		16

Simulation duration for 72 minutes:-

- Average serve for Ahmed = 61/72=0.847
- Average serve for Baker = 56/72=0.777
- The customer services by Ahmed =18 from 30 which mean 60%
- Average services time for Ahmed= 61/18 =3.93 minutes
- The customer services by Baker =12 from 30 which mean 40%
- Average services time for Baker= 56/12 =4.97 minutes
- Average waiting time for all= 16/30 = 0.53 minutes
- Average time between arrivals =96/30 =2.3 minutes
- Average time spend in system =72/30 =2.4 minutes

Example 3: Salesman Problem newspapers:-

This is normal problem inventory system where there was sells and buy of newspaper, salesman bought one newspaper by 150 dinar and sells it by 200 dinar and the rest of newspaper sold to factory Qratis by 50 dinar for each one, the salesman bought from distributor package consist of 10 newspapers (which he can buy 20, 30 or 40 news for each time). The order on the newspaper depends on news type in that day, may be good news day, fair news day or poor news day, with respective probability 0.20, 0.45, 0.35 .

Distribute of order on newspaper for each day according to news type will be:

	Demand Probability Distribution		
Demand	Good	Fair	Poor
40	0.03	0.10	0.44
50	0.05	0.18	0.22
60	0.15	0.40	0.16
70	0.20	0.20	0.12
80	0.35	0.08	0.06
90	0.15	0.04	0.00
100	0.07	0.00	0.00

Then to determine the optimal number for newspapers that salesman bought it we simulate system for 20 days and determine the daily profit from this relation:-

Profit= revenue from sales – cost of newspapers - lost profit from excess demand + salvage from scrap papers

- Revenue from sales= 200* number of newspapers sold
- Cost of newspapers= 150* number of newspapers purchased
- Lost profit from excess demand= 200* (number of newspapers required- number of newspaper purchased),

if no of required > no of purchased.

Salvage from sale= 50* (number of newspapers purchased – number of newspapers required), if no of purchased > no of required.

To solve this problem we make system simulation for 20 day to determine the number of newspaper purchased where the maximum profit.

The following tables for distribute type of news day:-

Type of Newsday	Probability	Cumulative Probability	Random Numbers Assignment
Good	0.35	0.35	0.000-0.350
Fair	0.45	0.80	0.351-0.800
Poor	0.20	1.00	0.801-1.000

And the following table to distribute order on newspapers according to type of day news

	Cumulative Probability			Random Numbers Assignment		
Demand	Good	Fair	Poor	Good	Fair	Poor
40	0.03	0.10	0.44	0.000-0.030	0.000-0.100	0.000-0.440
50	0.08	0.28	0.66	0.031-0.080	0.101-0.280	0.441-0.660
60	0.23	0.68	0.82	0.081-0.230	0.281-0.680	0.661-0.820
70	0.43	0.88	0.94	0.231-0.430	0.681-0.880	0.821-0.940
80	0.78	0.96	1.00	0.431-0.780	0.881-0.960	0.941-1.000
90	0.93	1.00	1.00	0.781-0.930	0.961-1.000	
100	1.00	1.00	1.00	0.931-1.000		

And make table to simulate 70 newspapers daily for 20 day and calculate the daily profit:-

Day	Random Numbers for Type of Newsday	Type of Newsday	Random Numbers for Demand	Demand	Revenue from Sales	Lost Profits from Excess Demand of	Salvage from Sale Scrap	Daily Profit
1	0.668258	Fair	0.516101	60	12000	-	500	2000
2	0.059141	Good	0.421215	70	14000	-	-	3500
3	0.844465	Poor	0.122752	40	8000	-	1500	-1000
4	0.575663	Fair	0.417585	60	12000	-	500	2000
5	0.777212	Fair	0.873137	70	14000	-	-	3500
6	0.721669	Fair	0.028883	40	8000	-	1500	-1000
7	0.940940	Poor	0.442408	40	8000	-	1500	-1000
8	0.739749	Fair	0.154498	50	10000	-	1000	500
9	0.322162	Good	0.789384	80	16000	2000	-	3500
10	0.933698	Poor	0.948650	70	14000	-	-	3500
11	0.682856	Fair	0.462126	60	12000	-	500	2000
12	0.948467	Poor	0.427917	40	8000	-	1500	-1000
13	0.321750	Good	0.436062	70	14000	-	-	3500
14	0.576449	Fair	0.852654	70	14000	-	-	3500
15	0.469982	Fair	0.664128	60	12000	-	500	2000
16	0.325208	Good	0.551011	80	16000	2000	-	3500
17	0.119710	Good	0.149210	60	12000	-	500	2000
18	0.526681	Fair	0.992919	90	18000	4000	-	3500
19	0.355738	Fair	0.339435	60	12000	-	500	2000
20	0.734686	Fair	0.231963	50	10000	-	1000	500
				244000	8000	11000	37000	

From table found that from simulate 20 days the average of daily profit= 1850 dinar based on 70 newspapers daily.