Monte Carlo Simulation

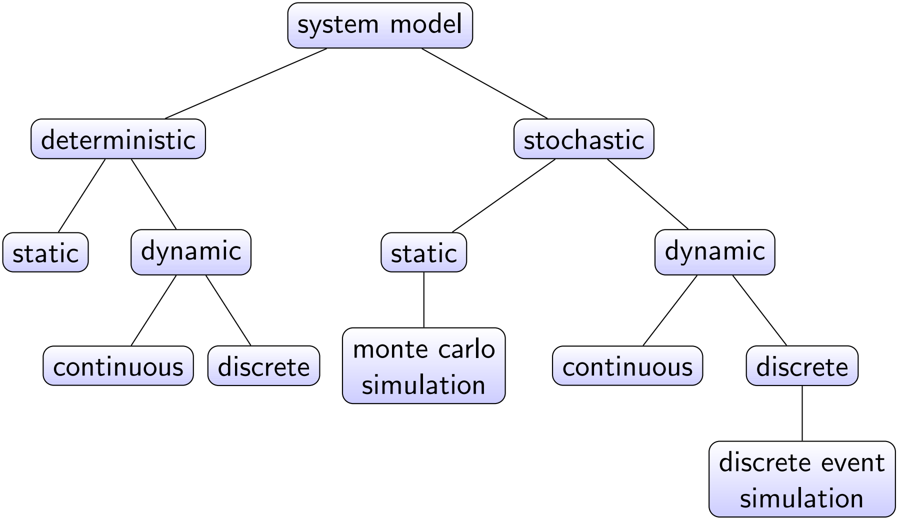
Lawrence M. Leemis and Stephen K. Park, Discrete-Event Simul A First Course, Prentice Hall, 2006

Hui Chen

Department of Engineering & Computer Science
Virginia State University
Petersburg, Virginia

April 17, 2017

Characterization of Models



Monte Carlo Simulation: Objective

- Estimate one of more probabilities by using an experimental technique
  - Its validity is based on the *frequency* theory of probability

Probability

- Empirical (Experimental) Probability.
  Repeat a random experiment n times and count the number of occurrences n_a of an event $\mathcal{A}$
  - The relative frequency of occurrence of event $\mathcal{A}$ is n_a/n
  - The frequency theory of probability asserts that the relative frequency converges as $n \to \infty$

    $$Pr(\mathcal{A}) = \lim_{n\to\infty} \frac{n_a}{n}$$

  - Related to the strong and weak *laws of large numbers*.
  - Monte Carlo simulation uses the *frequency* theory of probability in a natural way.
- Axiomatic Probability.
  A formal, set-theoretic approach
  - Mathematically construct the sample space and calculate the number of events $\mathcal{A}$
  - The *axiomatic* theory of probability.

Monte Carlo Simulation & Empirical and Axiomatic Probabilities

- The axiomatic theory of probability and the frequency theory of probability are complementary.
  - Doing one brings insight to the other. The best solution to any probability problem is a mathematical solution established via the axiomatic method and verified experimentally via an independent Monte Carlo simulation.
  - Hard problems. Many probability problems are too hard to solve mathematically. Monte Carlo simulation may be the only viable approach.

Example 2.3.1

- Roll two dice and observe the up faces. What is the probability of observing 2, or 3, or 4, or ....
- Axiomatic approach:
  - Possible outcomes:

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)
(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)

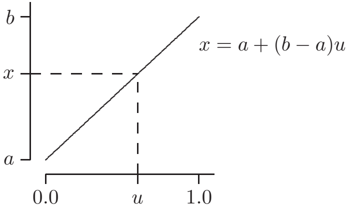
  - If the two up faces are summed, an integer-valued random variable, say X, is defined with possible values 2 through 12 inclusive

sum, x:	2	3	4	5	6	7	8	9	10	11	12
$Pr(X = x)$:	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

- $Pr(X = 7)$ could be estimated by replicating the *experiment* many times and calculating the relative frequency of occurrence of 7's

Random Variates

- A Random Variate is an algorithmically generated realization of a random variable
- $u = Random()$ generates a $Uniform(0, 1)$ random variate
- How can we generate a $Uniform(a, b)$ variate?



Generating a Uniform Random Variate

```
double Uniform(double a, double b)        /* use a < b */ {
   return (a + (b - a) * Random());
}
```

Equilikely Random Variates

- *Uniform*$(0, 1)$ random variates can also be used to generate an *Equilikely*(a, b) random variate

$$
\begin{aligned}
0 < u < 1 &\iff 0 < (b - a + 1)u < b - a + 1 \\
&\iff 0 \le \lfloor (b - a + 1)u \rfloor \le b - a \\
&\iff a \le a + \lfloor b - a + 1)u \rfloor \le b \\
&\iff a \le x \le b
\end{aligned}
$$

- Specifically, $x = a + \lfloor (b - a + 1)u \rfloor$

Generating an Equilikely Random Variate

```
long Equilikely(long a, long b)        /* use a < b */ {
    return (a + (long)((b - a + 1) * Random()));
}
```

Examples

- **Example 2.3.3** To generate a random variate x that simulates rolling two fair dice and summing the resulting up faces, use

$$x = \mathit{Equilikely}(1,6) + \mathit{Equilikely}(1,6);$$

Note that this is *note* equivalent to

$$x = \mathit{Equilikely}(2,12);$$

- **Example 2.3.4** To select an element x at random from the array a[0], a[1], . . . , a[n-1] use

$$i = \mathit{Equilikely}(0, n-1); x = a[i];$$

Galileo's Dice

- If three fair dice are rolled, which sum is more likely, a 9 or a 10?
  - There are $6^3 = 216$ possible outcomes

$$Pr(X = 9) = \frac{25}{216} \cong 0.116 \qquad \textit{and} \qquad Pr(X = 10) = \frac{27}{216} = 0.125$$

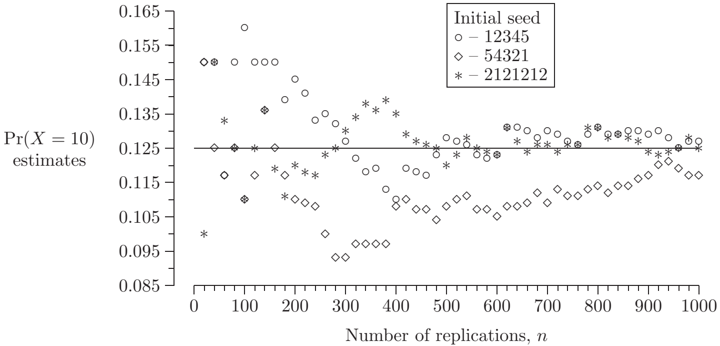
- Program *galileo* calculates the probability of each possible sum between 3 and 18
- The drawback of Monte Carlo simulation is that it only produces an estimate
  - Larger n does not guarantee a more accurate estimate. However, as n becomes larger, the uncertainty and accuracy of the probility estimates will tend to improve.

Exercise L13-1: Varitions of Galileo's Dice

- Run the Galileo's Dice program (in Blackboard) following the following guideline:
  - Choose three different seeds
  - Use the number of replications as 20, 40, 100, 200, 400, 1000, 10000, and 100000
  - Show the result in a graph similar to next slide

Example 2.3.6

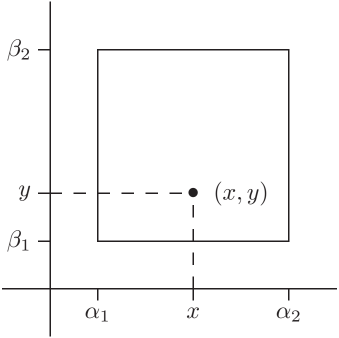
- Frequency probability estimates converge slowly and somewhat erratically



- You should always run a Monte Carlo simulation with multiple initial seeds

Geometric Applications: Rectangle

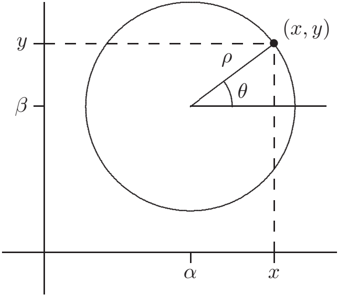
- Generate a point at random inside a rectangle with opposite corners at (α_1, β_1) and (α_2, β_2)



$$x = Uniform(\alpha_1, \alpha_2); \qquad y = Uniform(\beta_1, \beta_2);$$

Geometric Applications: Circle

- Generate a point (x, y) at random on the circumference of a circle with radius ρ and center (α, β)



$$\theta = Uniform(-\pi, \pi); \quad x = \alpha + \rho * cos(\theta); \quad y = \beta + \rho * sin(\theta);$$

Example 2.3.8

- Generate a point (x, y) at random interior to the circle of radius ρ centered at (α, β)

$$\theta = Uniform(-\pi, \pi); \quad r = Uniform(0, \rho);$$
$$x = \alpha + \rho * cos(\theta); \quad y = \beta + r * sin(\theta);$$

Correct?

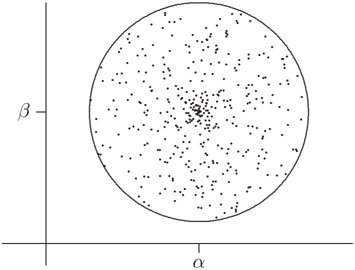
Example 2.3.8

- Generate a point (x, y) at random interior to the circle of radius ρ centered at (α, β)

$$\theta = Uniform(-\pi, \pi); \quad r = Uniform(0, \rho);$$

$$x = \alpha + \rho * cos(\theta); \quad y = \beta + r * sin(\theta);$$

Correct? INCORRECT!

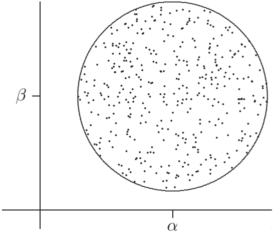


Acceptance/Rejection

- Generate a point at random within a circumscribed square and then either accept or reject the point

Generate a Random Point Interior to a Circle

do {
$x = Uniform(-\rho, \rho);$
$y = Uniform(-\rho, \rho);$ } while $(x * x + y * y >= \rho * \rho);$
$x = \alpha + x; \quad y = \beta + y;$
return $(x, y);$

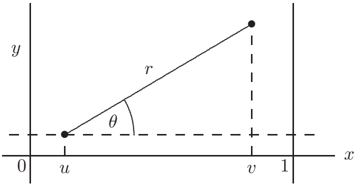


Exercise L13-2 Geometric Application

- Objective: visually examine correctness of a simulation
- Write a program that randomly generate 1000 points within a rectangle using the method in slide 13 and graph the result
- Write a program that reproduces the incorrect (slide 14) and correct (slide 16 generation of points interior to a circle as shown previous slides.
- Write a program to estimate π by extending the above programs.

Buffon's Needle Problem

- Suppose that an infinite family of infinitely long vertical lines are spaced one unit apart in the (x, y) plane. If a needle of length $r > 0$ is dropped at random onto the plane, what is the probability that it will land crossing at least one line?



  - u is the x-coordinate of the left-hand endpoint
  - v is the x-coordinate of the right-hand endpoint,

$$v = u + rcos\theta$$

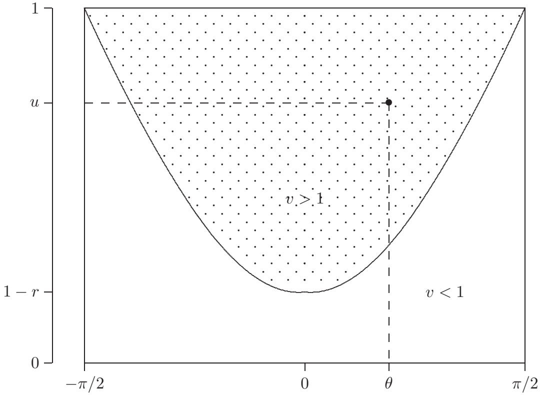
  - The needle crosses at least one line if and only if $v > 1$

Program buffon

- Program buffon is a Monte Carlo simulation
- Inspection of the program buffon illustrates how to solve the problem axiomatically

Axiomatic Approach to Buffon's Needle

- "Dropped at random" is interpreted (modeled) to mean that u and θ are independent *Uniform*$(0,1)$ and *Uniform*$(-\pi/2,\pi/2)$ random variables



Axiomatic Approach to Buffon's Needle

- The shaded region has a curved boundary defined by the equation $u = 1 - rcos\theta$
- if $0 < r \leq 1$, the area of the shaded region is

$$\pi - \int_{-\pi/2}^{\pi/2} (1 - rcos\theta) d\theta = r \int_{-\pi/2}^{\pi/2} cos\theta d\theta = \ldots = 2r$$

- Therefore, because the area of the rectangle is π the probability that the needle will cross at least one line is $2r/\pi$

Exercise L13-3: Buffon's Needle

- Objective: Compare simulation and axiomatic results (does your simulation program need a test case?)
- Calculate the probability that it will land crossing at least one line for the Buffon's needle problem using the axiomatic result 21.
- Revise the program buffon to output the estimated probability with at least 6 digits after the decimal point.
- Run the revised program buffon for 100, 1000, 10000, 100000, 1000000 replications with 3 different seeds for each number of replications
- Choose appropriate graphs to graph the following,
    - The results from the simulations
    - The axiomatic result
    - The different between the simulations and the axiomatic result (i.e., error)

Axiomatic and Experimental Approaches

- *Axiomatic* and *experimental* approaches are complementary
- Slight changes in assumptions can sink an axiomatic solution
- An axiomatic solution is intractable in some other cases
- Monte Carlo simulation can be used as an alterative in either case
- Four more examples of Monte Carlo simulation
  - Metrics and determinants
  - Craps
  - Hatchek girl
  - Stochastic activity network

Example 1: Matrix and Determinants

- *Matrix:* set of real or complex numbers in a rectangular array
- for matrix A, a_{ij} is the element in row i, column j

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

  where A is $m \times n$, i.e., m rows and n columns
- Interesting quantities: eigenvalue, trace, rank, and determinant

Determinants

- The *determinant* of a 2×2 matrix A is

$$|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

- The determinant of a 3×3 matrix A is

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Random Matrices

- *Random matrix*: matrix whose elements are random variables
- Consider a 3×3 matrix whose elements are random with positive diagonal, negative off-diagonal elements
- Question: What is the probability the determinant is positive?

$$\begin{vmatrix} +u_{11} & -u_{12} & -u_{13} \\ -u_{21} & +u_{22} & -u_{23} \\ -u_{31} & -u_{32} & +u_{33} \end{vmatrix} > 0$$

- Axiomatic solution is not easily calculated

Specification Model

- Let event $\mathcal{A}$ be that the determinant is positive
- Generate N 3×3 matrices with random elements
- Compute the determinant for each matrix
- Let $n_a =$ number of matrices with determinant > 0
- Probability of interest: $Pr(\mathcal{A}) \cong N_a/N$

Computational Model: Program det

det

```
for (i = 0; i < N; i++) {
    for (j = 1; j <= 3; j++) {
        for (k = 1; k <= 3; k++) {
            a[j][k] = Random();
            if (j != k)
            a[j][k] = -a[j][k];
        }
    }
    temp1 = a[2][2] * a[3][3] - a[3][2] * a[2][3];
    temp2 = a[2][1] * a[3][3] - a[3][1] * a[2][3];
    temp3 = a[2][1] * a[3][2] - a[3][1] * a[2][2];
    x = a[1][1]*temp1 - a[1][2]*temp2 + a[1][3]*temp3;
    if (x > 0)
        count++;
}
probability = (double)count/N;
```

Output From det

- Want N sufficiently large for a good point estimate
- Avoid recycling random number sequences
- Nine calls to Random() per 3×3 matrix $\rightarrow Nm/9 \cong 239000000$
- For initial seed 987654321 and $N = 200000000$,

$$Pr(\mathcal{A}) \cong 0.05017347$$

Point Estimate Considerations

- How many significant digits should be reported?
- Solution: run the simulation multiple times
- One option: use different initial seeds for each run
  - Caveat: Will the same squences of random numbers appear?
- Another option: use different a for each run
  - Caveat: Use a that gives a good random sequence
- For two runs with $a = 16807$ and 41214

$$Pr(\mathcal{A}) \cong 0.0502$$

Example 2: Craps

- Standard Craps Table

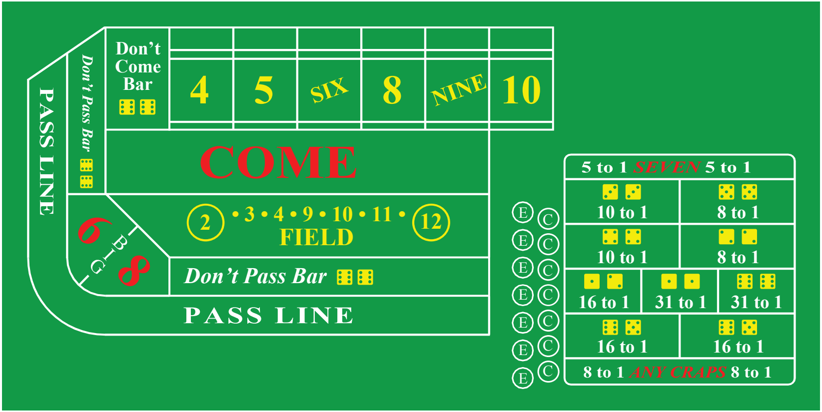


Figure: Retrieved from
http://en.wikipedia.org/wiki/File:Craps_table_layout.svg

Example 2: Craps

- Toss a pair of fair dice and sum the up faces
- Two phases: "come-out" and "point"
- Come-out. A new round starts after a come-out roll.
  - A roll of 2, 3 or 12 is a came-out roll called "craps" or "crapping out". Anyone betting the Pass line loses and anyone betting the Don't Pass line wins.
  - A roll of 7 or 11 is also a come-out roll called "natural". Anyone betting The Pass line wins and any one betting the Don't Pass line loses.
- Otherwise, sum becomes "point"
  - Roll until the point is matched or 7, and a new round starts. Anyone betting the Pass line wins if it is a match, loses if it is 7.
- What is $Pr(\mathcal{A})$, the probability of winning at craps if I am betting the Pass line?

Craps: Axiomatic Solution

- Requires conditional probability
- Axiomatic solution: $244/495 \cong 0.493$
- Underlying mathematics must be changed if assumptions change
  - Example: unfair dice
- Axiomatic solution provides a nice consistency check for (easier) Monte Carlo simulation

Craps: Specification Model

- Model one die roll with Equilikely(1, 6)

Algorithm 2.4.1

```
wins = 0;
for (i = 1; i <= N; i++) {
    roll = Equilikely(1, 6) + Equilikely(1, 6);
    if (roll = 7 or roll = 11)
        wins++;
    else if (roll != 2 and roll != 3 and roll != 12) {
        point = roll;
        do {
            roll = Equilikely(1, 6) + Equilikely(1, 6);
            if (roll == point) wins++;
        } while (roll != point and roll != 7)
    }
} return (wins/N);
```

Craps: Computational Model

- Program craps: uses switch statement to determine rolls
- For $N = 10000$ and three different initial seeds (see text)

$$Pr(\mathcal{A}) = 0.497, 0.485, \text{and } 0.502$$

- These results are consistent with 0.493 axiomatic solution
- This (relatively) high probability is attractive to gamblers, yet ensures the house will win in the long run
- If using simulation, how big should N be?

Example 3: Hatcheck Girl

- A hatcheck girl at a fancy restaurant collects n hats and returns them at random. What is the probability that everyone receives the wrong hat?
- Let $\mathcal{A}$ be that all checked hats are returned to wrong owners
- Without loss of generality, let the checked hats be numbered $1, 2, \ldots, n$
- The girl selects (equally likely) one of the remaining hats to return $\rightarrow n!$ permutations, each with probability $1/n!$
- Example: When $n = 3$ hats, possible return orders are

$$1,2,3 \qquad 1,3,2 \qquad 2,1,3 \qquad 2,3,1 \qquad 3,1,2 \qquad 3,2,1$$

- Only $2,3,1$ and $3,1,2$ correspond to all hats returned incorrectly

$$Pr(\mathcal{A}) = 1/3$$

Hatcheck: Specification Model

- Generate a random permutation of the first n integers
- The permutation corresponds to the order of hats returned

Clever Shuffling Algorithm (see Section 6.5)

```
for (i = 0; i < n - 1; i++) {
    j = Equilikely(i, n - 1);
    hold = a[j];
    a[j] = a[i]; /* swap a[i] and a[j] */
    a[i] = hold;
}
```

Generates a random permutation of an array a

- Check the permuted array to see if any element matches its index

Hatcheck: Computational Model

- Program hat: Monte Carlo simulation of hatcheck problem
- Uses shuffling algorithm to generate random permutation of hats
- For $n = 10$ hats, $N = 10,000$ replications, and three different seeds

$$Pr(\mathcal{A}) = 0.369, 0.369, \text{and } 0.368$$

- What happens to the probability as $n \to \infty$?
- If using simulation, how big should n and N be?

Hatcheck: Axiomatic Solution

- The probability $Pr(\mathcal{A})$ of no hat returned correctly is

$$1 - \left(1 - \frac{1}{2!} + \frac{1}{3!} - \ldots + (-1)^{n+1}\frac{1}{n!}\right)$$

- for $n = 10$, $Pr(\mathcal{A}) \cong 0.36787946$
- Important consistency check for validating craps
- As $n \to \infty$, the probability of no hat returned is

$$1/e \cong 0.36787944$$

Exercise L13-4

- Design an approach to show that the shuffle algorithm in slide 37 is correct.
- Implement the approach and graph the results.

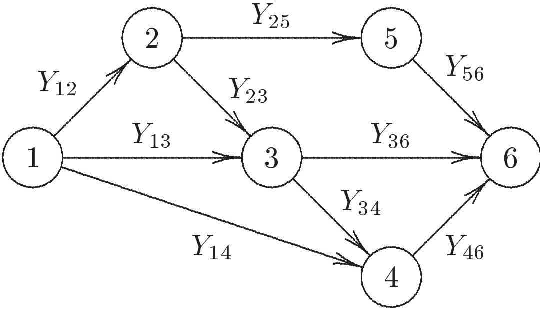
Example 4: Stochastic Activity Network

- Concerning project management. A project consists of many activities, and one may depend on another.
- Modeling project management with a network.
- Activity durations are positive random variables
- n nodes, m arcs (activities) in the network
- Single source node (labeled 1), single terminal node (labeled n)
- Y_{ij} : positive random activity duration for arc a_{ij}
- T_j : completion time of all activities entering node j
- A path is critical with a certain probability

$$p(\pi_k) = Pr(\pi_k \equiv \pi_c), k = 1, 2, \ldots, r$$

Example Stochastic Activity Network

- Each activity duration is a uniform random variate



Example: Y_{12} has a *Uniform*$(0, 3)$ distribution

Conceptual Model

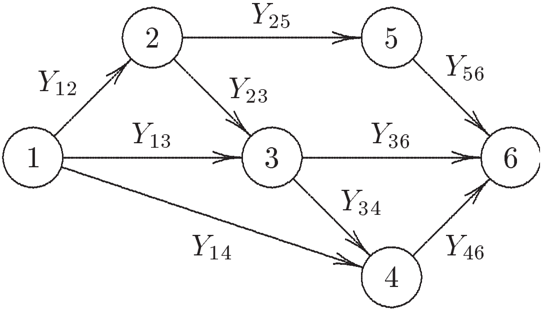
- Represent the network as an $n \times m$ node-arc incidence matrix N

$$N[i,j] = \begin{cases} 1 & \text{arc } j \text{ leaves node } i \\ -1 & \text{arc } j \text{ enters node } i \\ 0 & \text{otherwise} \end{cases}$$

- Use Monte Carlo simulation to estimate:
  - mean time to complete the network
  - probability that each path is critical

Conceptual Model

- Each activity duration is a uniform random variate



Example: Y_{12} has a $\mathit{Uniform}(0, 3)$ distribution

Specification Model

- Completion time T_j relates to incoming arcs

$$T_j = \max_{i \in \mathcal{B}(j)} \{T_i + Y_{ij}\} \quad j = 2, 3, \ldots, n$$

  where $B(j)$ is the set of nodes immediately before node j
- Example: in the previous six-node example

$$T_6 = \max\{T_3 + Y_{36}, T_4 + Y_{46}, T_5 + Y_{56}\}$$

- We can write a recursive function to compute the T_j

Conceptual Model

- The previous 6-node, 9-arc network is represented as follows:

$$N = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}$$

- In each row:
  - 1's represent arcs exiting that node
  - -1's represent arcs entering that node
- Exactly one 1 and one -1 in each column

Algorithm 2.4.2

- Returns a random time to complete all activities prior to node j for a single SAN with node-arc incidence matrix N

Algorithm 2.4.2

```
k = 1; l = 0; tmax = 0.0;
while (l < |$\mathcal{B}$(j)|) {
    if (N[j][k] == -1) {
        i = 1;
        while (N[j ][k] != 1)
            i++;
            t = Ti + Yi ;
            if (t >= $t_{max}$) $t_{max}$ = t;
            l++;
    }
    k++;
}
return tmax;
```

Computational Model

- Program *san*: MC simulation of a stochastic activity network
- Uses recursive function to compute completion times T_j (see text)
- Activity durations Y_{ij} are generated at random a priori
- Estimates T_n, the time to complete the entire network
- Computes critical path probabilities $p(\pi_k)$ for $k = 1, 2, \ldots, r$
- Axiomatic approach does not provide an analytic solution

Computational Model

- For 10000 realizations of the network and three initial seeds

$$T_6 = 14.64, 14.59, \text{and } 14.57$$

- Point estimates for critical path probabilities are

k	π_k	$\hat{p}_1(m_k)$	$\hat{p}_2(c_k)$	$\hat{p}_3(a_k)$	$\hat{p}_4(i_k)$
1	$\{a_{13}, a_{36}\}$	0.0168	0.0181	0.0193	0.0181
2	$\{a_{12}, a_{23}, a_{36}\}$	0.0962	0.0970	0.0904	0.0945
3	$\{a_{12}, a_{25}, a_{56}\}$	0.0013	0.0020	0.0013	0.0015
4	$\{a_{14}, a_{46}\}$	0.1952	0.1974	0.1907	0.1944
5	$\{a_{13}, a_{34}, a_{46}\}$	0.1161	0.1223	0.1182	0.1189
6	$\{a_{12}, a_{23}, a_{34}, a_{46}\}$	0.5744	0.5632	0.5801	0.5726

- Path π_6 is most likely to be critical – 57.26% of the time

Summary

- Concept of Monte Carlo simulation
- A few Monte Carlo simulation examples