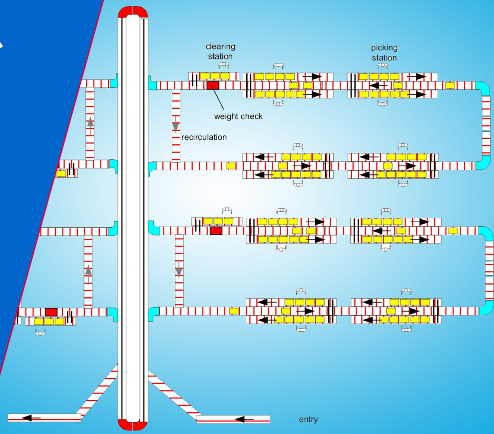
Analysis of Manufacturing Systems 4AB00

Ivo Adan



Technische Universiteit
Eindhoven
University of Technology

Contents

- Chapter 1
- Chapter 2: 2.1, 2.2, 2.3 (till 2.3.1), 2.8 (till 2.8.1), 2.9
- Chapter 7: 7.1, 7.2 (till 7.2.1), 7.3
- Appendix
  (Permutations, Combinations, Exponential function, Geometric series)

Understanding Probability

- Book consists of two parts:
  - Part one is intuitive and helps to develop a feel for probabilities
  - Part two is more formal and presents the basics of probability
- Book contains many, many challenging problems!
- It is about probability, not about manufacturing
- Probability is not only relevant in manufacturing, but in many more areas!
  - Computer networks (Internet)
  - Supply chains
  - Statistical physics
  - Quantum physics
  - Random heterogeneous materials
  - Biology
  - Computer simulation, and so on.

Recap

Factory physics will be grounded on

- Probability theory
- Queueing theory

Manufacturing

- Manufacturing system transforms material to meet demand
- Buffer is excess resource to correct misalignment between
  - transformation
  - demand

Performance

- Throughput: Number of jobs produced per time unit
- Flow time or cycle time: time it takes a job to go through the system
- Utilization: Fraction of time machine is producing
- Work-In-Process (WIP) level: Number of jobs in manufacturing system

Probabilistic problems

Test your probabilistic intuition:

- Magician
- Birthday problem
- Coin-tossing
- Scratch-and-win lottery
- Coincidence problem
- Boarding pass problem
- Monty Hall dilemma

Magician

A magician takes a random card from a thoroughly shuffled deck of 52 cards. You have to guess which card he is holding.

Before guessing you may ask one of the following two questions:

- Is the card a black one (spades or clover)?
- Is the card jack of hearts?

Which question will maximize the probability of guessing the right card?

Birthday problem

Consider a group of N randomly chosen persons.
What is the probability that at least 2 persons have the same birthday?

Almost birthday problem

What is the probability that at least 2 persons have their birthday within r days of each other?

Coin tossing

Two players A and B throw a *fair* coin N times.
If Head, then A gets 1 point; otherwise B.

- What happens to the absolute difference in points as N increases?
- What is the probability that one of the players is leading between 50% and 55% of the time? Or more than 95% of the time?
- In case of 20 trials, say, what is the probability of 5 Heads in a row?
- Suppose that n out of N times Head appears.
  For which values n do you believe the coin is indeed fair?

Scratch-and-win lottery

Each week a very popular lottery in Andorra prints 10^4 tickets. Each tickets has two 4-digit numbers on it, one visible and the other covered. The numbers are randomly distributed over the tickets. If someone, after uncovering the hidden number, finds two identical numbers, he wins a large amount of money.

- What is the average number of winners per week?
- What is the probability of at least one winner?

The same lottery prints 10^7 tickets in Spain.
What about the answers to the questions above?

Coincidence problem

Two people, strangers to one another, both living in Eindhoven, meet each other in the train. Each has approximately 200 acquaintances in Eindhoven.

What is the probability of the two people having an acquaintance in common?

Boarding pass problem

100 people line up to board an airplane with 100 seats. Each passenger gets on one at a time to select his assigned seat. The first one has lost his boarding pass and takes a random seat. Each subsequent passenger takes his own seat if available, and otherwise takes a random unoccupied seat.

You are the last passenger...

What is the probability that you can get your own seat?

Monty Hall dilemma

It is the climax of a game-show:

You have to choose a door out of three, behind one of them is the car of your dreams and behind the others a can of dog food.

You choose a door without opening it. The host (knowing what is behind the doors) then opens one of the remaining doors, showing a can of dog food.

Now you are given the opportunity to switch doors: Are you going to switch?

Empirical law of large numbers

Throwing a fair coin: Fraction of Heads should be $\frac{1}{2}$ in the long run.

Relative frequency of event H (Head) in n repetitions of throwing a coin is

$$f_n(H) = \frac{n(H)}{n}$$

where $n(H)$ is number of times Head occurred in the n repetitions. Then,

$$f_n(H) \to \frac{1}{2} \quad \text{as } n \to \infty.$$

More generally:

The relative frequency of event E approaches a limiting value as the number of repetitions tends to infinity.

Intuitively we would define the probability of event E as this limiting value.

Ingredients of a probability model

- Sample space S (often denoted by Ω):
  - Flipping a coin, $S = \{H, T\}$
  - Rolling a die, $S = \{1, 2, \ldots, 6\}$
  - Rolling a die twice, $S = \{(i, j), i, j = 1, 2, \ldots, 6\}$
  - Process times, $S = [0, \infty)$
  - Throwing darts, $S = \{(x, y), \sqrt{x^2 + y^2} \leq 1\}$
- Sample space S can be discrete or continuous.
- Events are subsets of S:
  $E = \{H\}$, $E = \{1, 2\}$, $E = \{(1, 2), (3, 4), (5, 6)\}$,
  $E = (0, 1)$, $E = \{(x, y), 0 \leq x \leq \frac{1}{4}, 0 \leq y \leq \frac{1}{4}\}$.
- Get new events by union, intersection, complement, and so on.

Probabilities on Events

For each event E there is a number $P(E)$ (the probability of E) such that:

1. $0 \leq P(E) \leq 1$
2. $P(S) = 1$
3. $E_1, E_2, \ldots$ mutually disjoint, then

$$P(E_1 \cup E_2 \cup \cdots) = P(E_1) + P(E_2) + \cdots$$

Examples:

- Flipping a coin, $P(\{H\}) = P(\{T\}) = \frac{1}{2}$
- Rolling a die, $P(\{1\}) = \frac{1}{6}$, $P(\{1, 2\}) = P(\{1\}) + P(\{2\}) = \frac{1}{3}$
- Rolling a die twice, $P(i, j) = \frac{1}{36}$
- Throwing darts, $P(E) =$ area of E, divided by area of unit disk

Probabilities on Events

For each event E there is a number $P(E)$ (the probability of E) such that:

1. $0 \leq P(E) \leq 1$
2. $P(S) = 1$
3. $E_1, E_2, \ldots$ mutually disjoint, then

$$P(E_1 \cup E_2 \cup \cdots) = P(E_1) + P(E_2) + \cdots$$

Examples:

- Discrete S, assign $p(s)$ for each $s \in S$. Then

$$P(E) = \text{sum of probabilities of outcomes in } E = \sum_{s \in E} p(s)$$

- Equally likely outcomes $s_1, \ldots, s_N$, so $p(s_i) = 1/N$ and

$$P(E) = \frac{N(E)}{N}$$

where $N(E)$ is the number of outcomes in the set E.

Some basic rules

- $P(E) \leq P(F)$ if $E \subset F$
- If $E_1, E_2, \ldots, E_n$ are mutually disjoint, then

$$P(E_1 \cup E_2 \cup \cdots \cup E_n) = P(E_1) + P(E_2) + \cdots + P(E_n)$$

- $P(E) = 1 - P(E^c)$ where E^c is the complement of E, so $E^c = S \setminus E$
- $P(E \cup F) = P(E) + P(F) - P(EF)$ where $EF = E \cap F$

Theoretical law of large numbers

Flipping a coin an unlimited number of times, then an outcome is an infinite sequence of Heads and Tails, for example

$$s = (H, T, T, H, H, H, T, \ldots).$$

Let $K_n(s)$ denote the number if Heads in the first n flips of outcome s.
Then according the theoretical (strong) law of large numbers,

$$\lim_{n\to\infty} \frac{K_n(s)}{n} = \frac{1}{2}$$

with probability 1.

More generally:

If an experiment is repeated an unlimited number of times, and if the experiments are independent of each other, then the fraction of times event E occurs converges with probability 1 to $P(E)$.

The method of computer simulation is based on this law!

Random variable

Function that assigns a numerical value to each outcome of an experiment: X

Examples:

- Rolling a die twice, X is sum of outcomes, so $X = i + j$
- Repeatedly flipping a coin, N is number of flips until first H

Discrete random variable X can only take discrete values, $x_1, x_2, \ldots$, and the function $p_j = P(X = x_j)$ is the probability mass function of X.

Examples:

- Rolling a die twice, $P(X = 2) = \frac{1}{36}$, $P(X = 3) = \frac{2}{36}$, $P(X = 5) = \frac{4}{36}$
- Number of coin flips until first H, with $P(H) = 1 - P(T) = p$,

$$P(N = n) = (1 - p)^{n-1} p, \quad n = 1, 2, \ldots$$

Expected value

For a random variable X with probability mass function $p_j = P(X = x_j)$,

$$E(X) = \sum_{j=1}^{\infty} x_j p_j$$

is its expected value or expectation or mean value.

Example:

- Rolling a die, X is the number of points,

$$E(X) = 1 \times \frac{1}{6} + 2 \times \frac{1}{6} + \cdots + 6 \times \frac{1}{6} = 3.5$$

Remarks:

- Expected value is the weighted average of the possible values of X
- Expected value is not the same as “most probable value”
- Expected value is not restricted to possible values of X

Expected value

Example:

- By repeatedly rolling a die, the average value of the points obtained in the rolls gets closer and closer to 3.5 as the number of rolls increases.
- This is the *empirical law of large numbers for expected value*.

More generally, let X_k be the outcome of the kth repetition of the experiment:

The average $\frac{1}{n}(X_1 + \cdots + X_n)$ over the first n repetitions converges with probability 1 to $E(X)$.

Remarks:

- This is the *theoretical law of large numbers for expected value*.
- The expected value $E(X)$ can thus be interpreted as the long run average.

Generating random numbers

Start with non-negative integer number z_0 (seed).
For $n = 1, 2, \ldots$

$$z_n = f(z_{n-1})$$

f is the pseudo-random generator

In practice, the following function is often used:

$$z_n = az_{n-1}(\text{modulo } m)$$

with $a = 630360016$, $m = 2^{31} - 1$.

Then $u_n = z_n/m$ is "random" on the interval $(0, 1)$.

"random" means here "passes lots of statistical tests..."

Simulating from other distributions

Let U be uniform on $(0, 1)$. Then simulating from:

- Interval (a, b):

$$a + (b - a)U$$

- Integers $1, \ldots, M$:

$$1 + \lfloor MU \rfloor$$

- Discrete probability distribution: $p_j = P(X = x_j) = p_j$, $j = 1, \ldots, M$

$$\text{if } U \in \left[\sum_{i=1}^{j-1} p_i, \sum_{i=1}^{j} p_i\right), \text{ then } X = x_j$$

Array method

Suppose $p_j = k_j/100$, $j = 1, \ldots, M$,
where k_js are integers with $0 \leq k_j \leq 100$

Construct list (array) $a[i]$, $i = 1, \ldots, 100$, as follows:

- set $a[i] = x_1$ for $i = 1, \ldots, k_1$
- set $a[i] = x_2$ for $i = k_1 + 1, \ldots, k_1 + k_2$, and so on.

Then, first, sample random index I from $1, \ldots, 100$:

$$I = 1 + \lfloor 100U \rfloor \text{ and set } X = a[I]$$

Random permutation

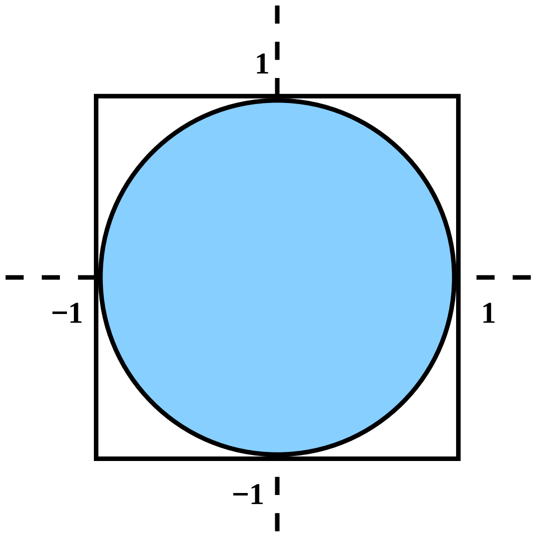
Algorithm for generating random permutation of $1, \ldots, n$:

1. Initialize $t = N$ and $a[i] = i$ for $i = 1, \ldots, N$;
2. Generate a random number u between 0 and 1;
3. Set $k = 1 + \lfloor tu \rfloor$; swap values of $a[k]$ and $a[t]$;
4. Set $t = t - 1$;
   If $t > 1$, then return to step 2;
   otherwise stop and $a[1], \ldots, a[N]$ yields a permutation.

Remark:

- Idea of algorithm: randomly choose a number from $1, \ldots, N$ and place that number at position N, then randomly choose a number from the remaining $N - 1$ and place that one at position $N - 1$, and so on.
- The number of operations is of order N.

Estimating π



- Sample (x, y) uniform from square $[-1, 1] \times [-1, 1]$
- Experiment is successful if (x, y) falls in unit circle
- Then by law of large numbers:

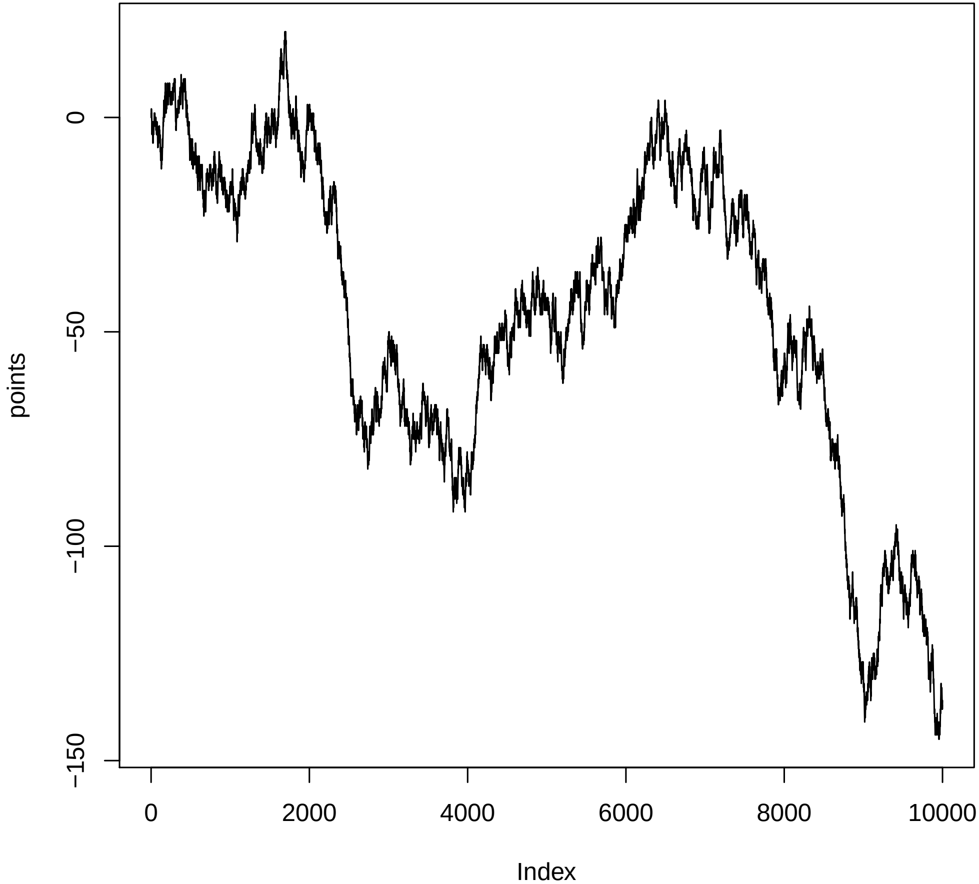
$$\frac{\pi}{4} \approx \frac{\text{number of successful experiments}}{\text{total number of experiments}}$$

Simulation of coin tossing


```
model coin():
    int n = 0, N = 10000, points_A = 0, points_B = 0;
    dist real u = uniform (0.0, 1.0);
    file f = open("data.txt", "w");

    while n < N:
        if sample u < 0.5:
            points_A = points_A + 1
        else:
            points_B = points_B + 1
        end;
        n = n + 1;
        write(f, "%d ", points_A - points_B);
    end
    close(f)
end
```

Simulation of coin tossing



Simulation of coin tossing

$P(\alpha, \beta)$ is the probability that one of the players is leading between $100\alpha\%$ and $100\beta\%$ of the time.

To determine $P(\alpha, \beta)$ do many times the experiment:

Toss a coin N times.

An experiment is successful if one of the players is leading between $100\alpha\%$ and $100\beta\%$ of the time.

Then by the law of large numbers:

$$P(\alpha, \beta) \approx \frac{\text{number of successful experiments}}{\text{total number of experiments}}$$

Simulation of coin tossing


```
xper X():
    int n, success, M = 1000, N = 10000, time_A, time_B;
    real a = 0.5, b = 0.6;

    while n < M:
        time_A = coin(N);
        time_B = N - time_A;
        if (a < time_A / N) and (time_A / N < b):
            success = success + 1;
     end;
        if (a < time_B / N) and (time_B / N < b):
            success = success + 1;
        end;
        n = n + 1
    end

    writeln("P(a,b) = %g", success / M);
end
```

Simulation of coin tossing


```
model int coin(int N):
    int n, points_A, points_B, time_A;
    dist real u = uniform (0.0, 1.0);

    while n < N:
        if sample u < 0.5:
            points_A = points_A + 1
        else:
            points_B = points_B + 1
        end;
        if points_A >= points_B:
            time_A = time_A + 1;
        end;
        n = n + 1;
    end
    exit time_A;
end
```

Simulation of coin tossing

Results for $M = 10^3$ and $N = 10^4$

(α, β)	$P(\alpha, \beta)$
(0.50,0.55)	0.06
(0.50,0.60)	0.13
(0.90,1.00)	0.42
(0.95,1.00)	0.26
(0.98,1.00)	0.16

Successive heads

$P(k)$ is the probability of at least k successive Heads in case of 20 trials

To determine $P(k)$ do many time the experiment:

Toss a coin 20 times.

An experiment is successful if at least k Heads in a row appear

Then by the law of large numbers:

$$P(k) \approx \frac{\text{number of successful experiments}}{\text{total number of experiments}}$$

Successive heads


```
xper X():
    int n, success, M = 1000, N = 20, k = 5;

    while n < M:
        if coin(k, N):
            success = success + 1
        end;
        n = n + 1
    end

    writeln("P(%d) = %g", k, success / M);
end
```

Successive heads


```
model bool coin(int k, N):
    bool k_row;
    int n, nr_Heads;
    dist real u = uniform (0.0, 1.0);

    while n < N and not k_row:
        if sample u < 0.5:
            nr_Heads = nr_Heads + 1
        else:
            nr_Heads = 0
        end;
        if nr_Heads >= k:
            k_row = true;
        end;
        n = n + 1;
    end
    exit k_row;
end
```

Successive heads

Results for $M = 10^3$

k	$P(k)$
1	1.00
2	0.98
3	0.80
4	0.46
5	0.25
6	0.13
7	0.05

Birthday problem

$P(N)$ is the probability that at least two persons have the same birthday in a group of size N

To determine $P(N)$ do many times the experiment:

> Take a group of N randomly chosen persons and compare their birthdays.

An experiment is successful is at least two persons have the same birthday.

Then by the law of large numbers:

$$P(N) \approx \frac{\text{number of successful experiments}}{\text{total number of experiments}}$$

Birthday problem


```
xper X():
    int n, success, M = 1000, N = 25;

    while n < M:
        if (birthday(N)):
            success = success + 1;
        end;
        n = n + 1
    end

    writeln("P(%d) = %g", N, success / M);
end
```

Generating a random group


```
model bool birthday(int N):
    bool same;
    int n, new;
    list(365) bool day;
    dist int u = uniform (0, 365);

    while n < N and not same:
        new = sample u;
        if day[new]:
            same = true
        else:
            day[new] = true
        end;
        n = n + 1;
    end
    exit same;
end
```

Birthday problem

Results for $M = 10^3$

N	$P(N)$
10	0.13
15	0.25
20	0.40
25	0.56
30	0.72
40	0.90
50	0.97

Birthday problem

Probability of all having a different birthday is

$$\frac{365 \times 364 \times \cdots \times (365 - N + 1)}{365^N}$$

so the probability of *at least* two people having the same birthday is

$$1 - \frac{365 \times 364 \times \cdots \times (365 - N + 1)}{365^N}$$

Question:

What is the probability of *exactly* two people having the same birthday?

Question:

In the German Lotto 6/49 it happened on June 21, 1995, that exactly the same numbers had been drawn as previously on December 20, 1986. This was the first time out of 3.016 drawings! Incredible! Or not?

Almost birthday problem


```
model bool birthday(int N, r):
    bool almost;
    int n, new;
    list(365) bool day;
    dist int u = uniform (0, 365);

    while n < N and not almost:
        new = sample u;
        for i in range(new-r, new+r+1):
            if day[i mod 365]:
                almost = true
            end;
        end;
        day[new] = true;
        n = n + 1;
    end
    exit almost;
end
```

Almost birthday problem

Results for $M = 10^3$

N	r	$P(N)$
10	0	0.11
	1	0.32
	2	0.52
	7	0.87
20	0	0.40
	1	0.80
30	0	0.70
	1	0.98