Simulation techniques
- A primer

Harry Perros

Topics

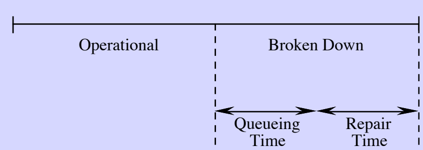
- A hand simulation
- Random numbers and random variates
- Simulation design techniques
- Statistical estimations
- Simulation languages

What is a simulation model?

- A simulation model is a computer program that depicts a real system.
- It uses random numbers to represent the occurrence of events and it racks the evolution of the various events that occur in the real system.
- Statistical techniques are used to obtain estimates of performance measures.
- Typically, in the industry one uses a simulation language for rapid model building.

The machine repairman problem

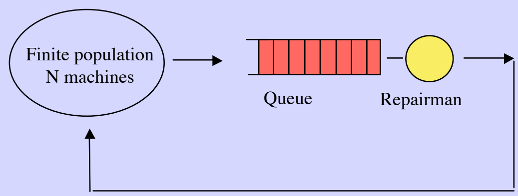
- There are N machines and a one repairman.
- Machines are repaired in FIFO.
- A machine has the following cycle:



- Operational and repair time are known. What is the queueing time in the repairman's queue?

The queueing system

- The repairman's queue is a single server queue fed by a finite population.



Building the model

- The first and most important step in building a simulation model is to identify the basic *events* whose occurrence will alter the *status* of the system
- *System status* - this is depicted by a set of variables that describe the state of the system.

- The system status: Machine repairman

  Number n of broken down machines (queueing plus the one being repaired) = *Number of operational machines*

  – n=0: repairman's queue is empty, repairman is idle. Number of operational machines is N

  – n=1, repairman's queue is empty, repairman is busy. Number of operational machines is N-1

  – n>1, repairman is busy, n-1 broken down machines in the queue. Number of operational machines is N-n

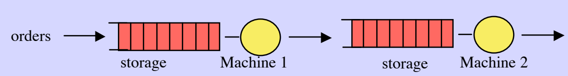
- Events:

  – *A machine breaks down* (arrival occurs at the queue).

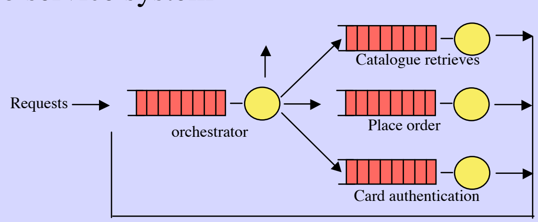
  – *A machine is fixed* (departure occurs from the queue).

Other examples of system status and events

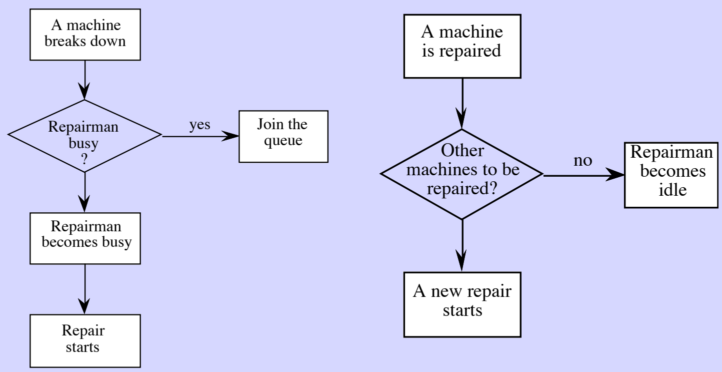
- Two stage production system



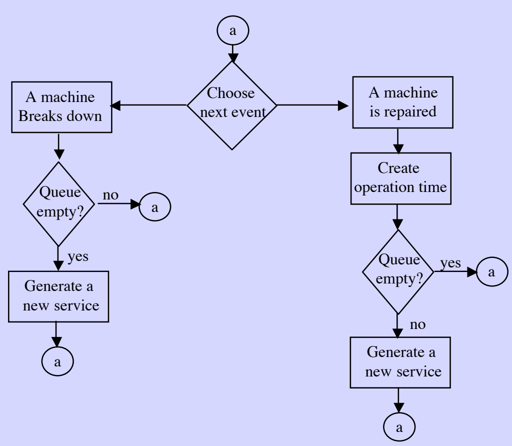
- Web service system



What happens when an event occurs



Event manipulation logic



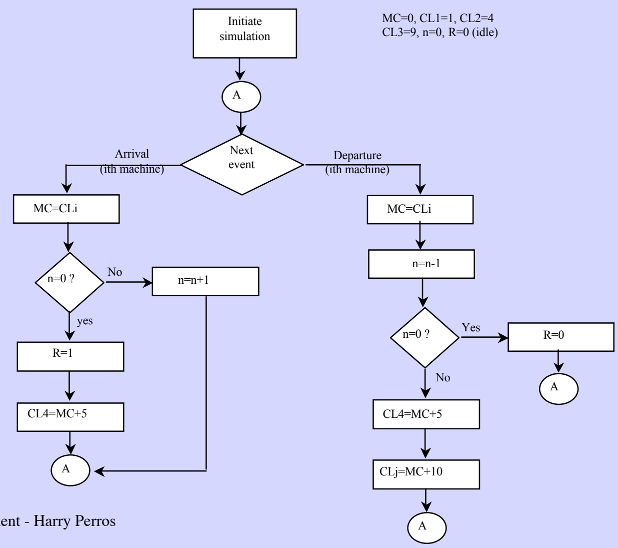
Event clocks and master clock

- A clock is associated with each operational machine, to show the time instant at which the machine will break down.
- In addition another clock will be used to show the time instant at which a machine currently being repaired will become operational.
- A *master clock* is used to simply keep track of the simulated time.

- **A hand simulation - three machines**
  - Operational time = 10, and repair time = 5 units of time.
  - CL1, CL2, CL3 are associated with machines 1, 2, 3
  - CL4 repairman's completion time
  - At t=0 all three machines are operational, CL1=1, CL2=4, CL3=9

MC	CL1	CL2	CL3	CL4	n	
0	1	4	9	-	0	idle
1	-	4	9	6	1	busy
4	-	-	9	6	2	busy
6	16	-	9	11	1	busy
9	16	-	-	11	2	busy
11	16	21	-	16	1	busy
16	-	21	26	21	1	busy

Flow-chart for computer program



Generating random numbers

- In the example we assumed that the operational times and repair times are all constant.
- However, in real life, they may follow an empirical distribution
- We need to have a technique for generating numbers from empirical and theoretical distributions in a random fashion..

Random numbers

- *Pseudo-random numbers*:
  - These are uniformly distributed random numbers in [0,1]. They are usually referred to as *random numbers*
- *Random variates* or *stochastic variates:*
  - These are random numbers that follow any distribution other than the above uniform distribution in [0,1].

Pseudo-random numbers

- In a sense, there is no such a thing as a single random number.
- Rather, we speak of a sequence of random numbers which pass a battery of statistical tests.
- Pseudo random numbers are generated in a deterministic fashion using a relationship, such as:

$$x_{i+1} = ax_i + c \pmod{m}$$

where a = 314,159,269, c = 453,806, 245, and m = 2^{31}

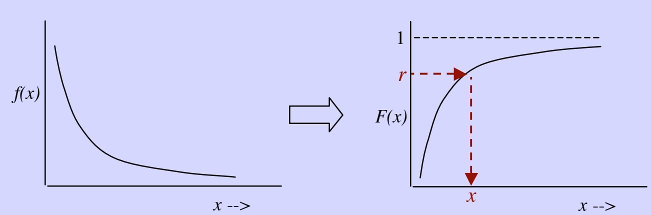
The starting value x_0 is known as the *seed*.

Random variates

- Depending upon the situation, we may need random variates from a given histogram (calculated using real-life observations) or from a theoretical distribution.
- Random variates are generated using the pseudo-random numbers in a variety of ways.

The inverse method

- This is a popular technique, and it requires the ability to invert a function.



Sampling from the exponential distribution

- Prob. density function: $f(x) = ae^{-ax}$, $a > 0$, $x \geq 0$.
- Cumulative density function: $F(x) = 1 - e^{-ax}$.
- Generate a pseudo-random number r. Then

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

or

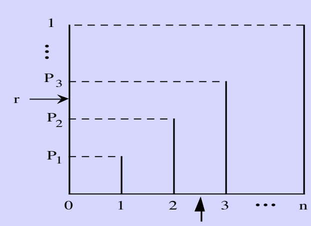
$$1 - r = e^{-ax}$$

or

$$x = -\log_e(1-r) = -(1/a)\log_e(1-r).$$

Sampling from a discrete probability distribution

- X is a discrete random variable with $p(X = i) = p_i$. Let $p(X \leq i) = P_i$ be the cumulative probability.



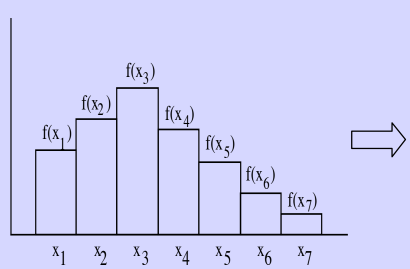
The newsboy problem

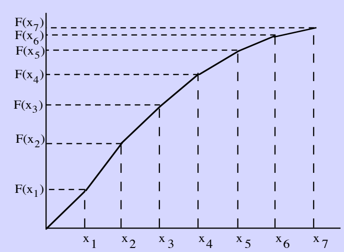
X	1	2	3	4	5
f(x)	0.20	0.20	0.30	0.15	0.15

X	1	2	3	4	5
F(x)	0.20	0.40	0.70	0.85	1.00

- Generate a random number r. Then:
  - If $0.85 < r \le 1.00$ then $x = 5$
  - If $0.70 < r \le 0.85$ then $x = 4$
  - If $0.40 < r \le 0.70$ then $x = 3$
  - If $0.20 < r \le 0.40$ then $x = 2$
  - Otherwise $x = 1$

Sampling from a histogram

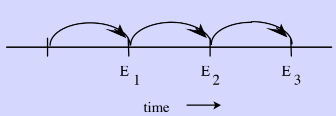


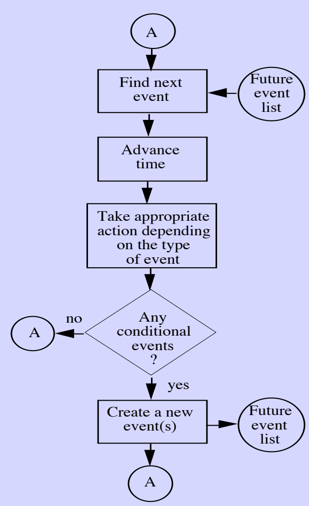


Simulation designs

- Various ways to track the occurrence of events:
  - *Event-advance design*
  - *Unit-time advance design*
  - *Activity-based*

Event-advance design

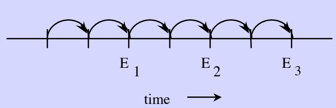


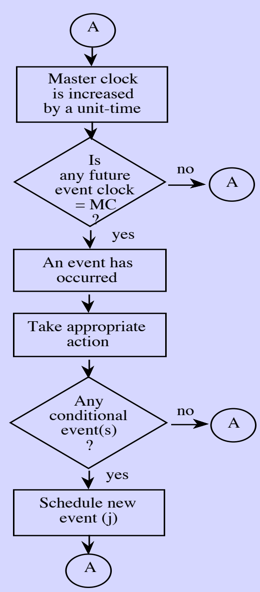


Data structures

- This approach requires locating the event that will occur next in the future.
- CPU intensive if there are lots of events
- Special data structures can be used to manipulate the events, i.e.
  - Locate the next event
  - Appropriately insert new events

Unit-time advance design





Statistical estimation

- After the simulation model has been built and debugged, and it can be used to generate data, which is then analyzed statistically.
- For instance, the repairman queue simulation will yield a sequence of waiting times. From this we can compute the mean waiting time and its confidence intervals

Confidence Intervals

- Let $x_1,x_2,...,x_n$ be n consecutive endogenously obtained observations of a random variable. Then

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

is an unbiased estimate of the true population mean, i.e., the expectation of the random variable.

If the observations $x_1, x_2, \ldots, x_n$ are independent of each other, then the standard deviation is given by:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

and, therefore, we obtain the confidence interval

$$(\bar{x} - 1.96\frac{s}{\sqrt{n}}, \bar{x} + 1.96\frac{s}{\sqrt{n}})$$

Confidence intervals: The Batch means method

- This is a fairly popular technique when successive observations are not independent, which is often the case.
- Successive observations are divided into batches of observations b.

| $x_1, x_2, \ldots, x_b$ | $x_{b+1}, x_{b+2}, \ldots, x_{2b}$ | . . . | $x_{kb+1}, x_{kb+2}, \ldots, x_{(k+1)b}$ |

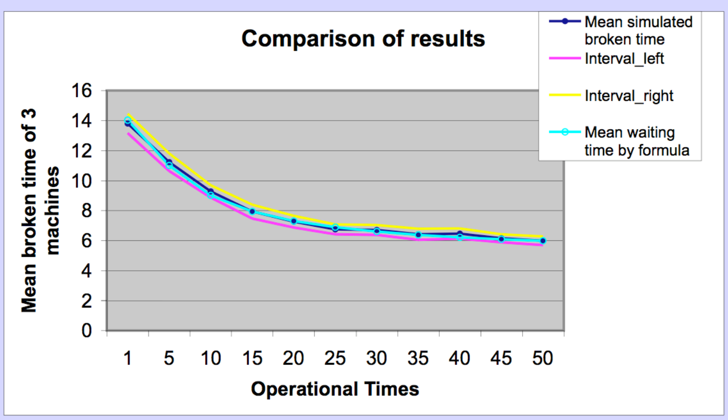
batch 1 *batch 2* *batch k*

- Each sample mean is treated as an individual observations and the set of all sample means is treated as set of independent observations.

Validation

- How do you validate a simulation model of a system that does not exist??
  - Debug code by hand
  - Run test cases, for which the results can be predicted in advance
  - Interpret intuitively the obtained results
  - If applicable, simulate first an existing system, validate the model against actual data, and then model the future system under study.

Validation for the machine repairman problem
This is a simple problem can be solved using queueing theory!



Simulation Languages

- Permit rapid development of a simulation model.
- Typically, they have an animation component that shows how the system evolves.
- Various languages are available: GPSS/H, SLAM II, SIMAN/ARENA

GPSS/H

- This language has been around for over 25 years.
- It's process oriented. That is, it follows the path of customers through service stations.
- The language provides standard modules, referred to as *blocks*, and each blocks performs a specific action on the flow of customers.

An example: the M/M/1 queue

```
SIMULATE
GENERATE   RVEXPO(1, 1.0)
QUEUE            QUEUE
SEIZE            SERVER
DEPART     QUEUE
ADVANCE    RVEXPO(2, 0.5)
RELEASE          SERVER
TERMINATE 1
START            1000
END
```

The program will run for 1000 customers and automatically generate statistics, such as the mean queue length and server utilization.

SIMAN/ARENA - M/M/1

```
BEGIN;
    CREATE,, EX(1,1) : EX(1,1) : MARK (1);
    QUEUE, 1;
    SEIZE: SERVER;
    TALLY;1, INT(1)
    COUNT: 1,1;
    DELAY: EX (2,2)
    RELEASE:SERVER:DISPOSE;
END;
```