Commonly Used Distributions

- Random number generation algorithms for distributions commonly used by computer systems performance analysts.
- Organized alphabetically for reference
- For each distribution:
  - Key characteristics
  - Algorithm for random number generation
  - Examples of applications

Bernoulli Distribution

- Takes only two values: failure and success or $x = 0$ and $x = 1$, respectively.
- Key Characteristics:
  1. Parameters: p = Probability of success $(x = 1)$ $0 \leq p \leq 1$
  2. Range: $x = 0, 1$
  3. pmf: $f(x) = \begin{cases} 1-p, & \text{if } x = 0 \\ p, & \text{if } x = 1 \\ 0, & \text{Otherwise} \end{cases}$
  4. Mean: p
  5. Variance: $p(1-p)$

- Applications: To model the probability of an outcome having a desired class or characteristic:

  1. A computer system is up or down.
  2. A packet in a computer network reaches or does not reach the destination.
  3. A bit in the packet is affected by noise and arrives in error.

- Can be used only if the trials are independent and identical
- Generation: Inverse transformation
  Generate $u \sim U(0, 1)$
  If $u \leq p$ return 0. Otherwise, return 1.

Beta Distribution

- Used to represent random variates that are bounded
- Key Characteristics:
  1. Parameters: a, b = Shape parameters, $a > 0$, $b > 0$
  2. Range: $0 \leq x \leq 1$
  3. pdf: $f(x) = \frac{x^{a-1}(1-x)^{b-1}}{\beta(a,b)}$
     $\beta(.)$ is the beta function and is related to the gamma function as follows:
     $$\beta(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}dx = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$
  4. Mean: $a/(a+b)$
  5. Variance: $ab/\{(a+b)^2(a+b+1)\}$
- Substitute $(x - x_{min})/(x_{max} - x_{min})$ in place of x for other ranges

- Applications: To model random proportions
  1. Fraction of packets requiring retransmissions.
  2. Fraction of remote procedure calls (RPC) taking more than a specified time.
- Generation:
  1. Generate two gamma variates $\gamma(1, a)$ and $\gamma(1, b)$, and take the ratio:

$$BT(a, b) = \frac{\gamma(1, a)}{\gamma(1, a) + \gamma(1, b)}$$

  2. If a and b are integers:
     (a) Generate $a + b + 1$ uniform U(0,1) random numbers.
     (b) Return the the a^{th} smallest number as BT(a, b).

3. If a and b are less than one:
   (a) Generate two uniform U(0,1) random numbers u_1 and u_2
   (b) Let $x = u_1^{1/a}$ and $y = u_2^{1/b}$. If $(x + y) > 1$, go back to the previous step. Otherwise, return $x/(x + y)$ as BT(a, b).
4. If a and b are greater than 1:
   Use rejection

Binomial Distribution

- The number of successes x in a sequence of n Bernoulli trials has a binomial distribution.
- Characteristics:

  1. Parameters:

     p = Probability of success in a trial, $0 < p < 1$.

     n = Number of trials; n must be a positive integer.

  2. Range: $x = 0, 1, \ldots, n$
  3. pdf: $f(x) = \binom{n}{x} p^x (1-p)^{n-x}$
  4. Mean: np
  5. Variance: $np(1-p)$

- Applications: To model the number of successes
  1. The number of processors that are up in a multiprocessor system.
  2. The number of packets that reach the destination without loss.
  3. The number of bits in a packet that are not affected by noise.
  4. The number of items in a batch that have certain characteristics.
- Variance $<$ Mean $\Rightarrow$ Binomial
  Variance $>$ Mean $\Rightarrow$ Negative Binomial
  Variance $=$ Mean $\Rightarrow$ Poisson
- Generation:
  1. Composition: Generate n U(0,1). The number of RNs that are less than p is BN(p, n)

2. For small p:
   (a) Generate geometric random numbers $G_i(p) = \lceil \frac{\ln(u_i)}{\ln(1-p)} \rceil$.
   (b) If the sum of geometric RNs so far is less than or equal to n, go back to the previous step. Otherwise, return the number of RNs generated minus one. If $\Sigma_{i=1}^{m} G_i(p) > n$, return $m - 1$.
3. Inverse Transformation Method: Compute the CDF F(x) for $x = 0, 1, 2, \ldots$, n and store in an array. For each binomial variate, generate a U(0,1) variate u and search the array to find x so that $F(x) \leq u < F(x + 1)$; return x.

Chi-Square Distribution

- Sum of squares of several unit normal variates
- Key Characteristics:
  1. Parameters: ν=degrees of freedom, ν must be a positive integer.
  2. Range: $0 \leq x \leq \infty$
  3. pdf: $f(x) = \frac{x^{(\nu-2)/2}e^{-x/2}}{2^{\nu/2}\Gamma(\nu/2)}$

     Here, $\Gamma(.)$ is the gamma function defined as follows:

     $$\Gamma(b) = \int_0^\infty e^{-x}x^{b-1}dx$$
  4. Mean: ν
  5. Variance: 2ν

- Application: To model sample variances.
- Generation:

1. $\chi^2(\nu) = \gamma(2, \nu/2)$:
   For ν even:
   $\chi^2(\nu) = -\frac{1}{2}\ln\left(\Pi_{i=1}^{\nu/2} u_i\right)$
   For ν odd:
   $\chi^2(\nu) = \chi^2(\nu - 1) + [N(0, 1)]^2$
2. Generate ν N(0,1) variates and return the sum of their squares.

Erlang Distribution

- Used in queueing models
- Key characteristics:

1. Parameters:
   $a =$ Scale parameter, $a > 0$
   $m =$ Shape parameter;
   m is a positive integer
2. Range: $0 \leq x \leq \infty$
3. pdf: $f(x) = \frac{x^{m-1}e^{-x/a}}{(m-1)!a^m}$
4. CDF: $F(x) = 1 - e^{-x/a}\left[\Sigma_{i=0}^{m-1} \frac{(x/a)^i}{i!}\right]$
5. Mean: am
6. Variance: a^2m

- Application: Extension to the exponential distribution if the coefficient of variation is less than one

  1. To model service times in a queueing network model.
  2. A server with Erlang(a, m) service times can be represented as a series of m servers with exponentially distributed service times.
  3. To model time-to-repair and time-between-failures.

- Generation: Convolution
  Generate m U(0,1) random numbers u_i and then:

$$Erlang(a, m) \sim -a \ln \left(\prod_{i=1}^{m} u_i \right)$$

Exponential Distribution

- Used extensively in queueing models.
- Key characteristics
  1. Parameters: a = Scale parameter = Mean, $a > 0$
  2. Range: $0 \leq x \leq \infty$
  3. pdf: $f(x) = \frac{1}{a}e^{-x/a}$
  4. CDF: $F(x) = 1 - e^{-x/a}$
  5. Mean: a
  6. Variance: a^2
- Memoryless Property: Past history is not helpful in predicting the future

- Applications: To model time between successive events

  1. Time between successive request arrivals to a device.
  2. Time between failures of a device.

  The service times at devices are also modeled as exponentially distributed.

- Generation: Inverse transformation
  Generate a U(0,1) random number u and return $-a\ln(u)$ as Exp(a)

Memoryless Property

- Remembering the past does not help in predicting the time till the next event.

$$F(\tau) = P(\tau < t) = 1 - e^{-\lambda t} t \geq 0$$

- At $t = 0$, the mean time to the next arrival is $1/\lambda$.
- At $t = x$, the distribution of the time remaining till the next arrival is:

$$\begin{aligned} & P(\tau - x < t | \tau > x) \\ &= \frac{P(x < \tau < x + t)}{P(\tau > x)} \\ &= \frac{P(\tau < x + t) - P(\tau < x)}{P(\tau > x)} \\ &= \frac{(1 - e^{-\lambda(x+t)}) - (1 - e^{-\lambda x})}{e^{-\lambda t}} \\ &= 1 - e^{-\lambda x} \end{aligned}$$

This is identical to the situation at $t = 0$.

F Distribution

- The ratio of two chi-square variates has an F distribution.
- Key characteristics:

1. Parameters:
   n = Numerator degrees of freedom;
   n should be a positive integer.
   m = Denominator degrees of freedom;
   m should be a positive integer.
2. Range: $0 \leq x \leq \infty$
3. pdf: $f(x) =$
   $\frac{(n/m)^{n/2}}{\beta(n/2,m/2)} x^{(n-2)/2} (1 + \frac{n}{m}x)^{-(n+m)/2}$
4. Mean: $\frac{m}{m-2}$, provided $m > 2$.
5. Variance: $\frac{2m^2(n+m-2)}{n(m-2)^2(m-4)}$, provided $m > 4$.

- High quantiles:

$$F_{[1-\alpha;n,m]} = \frac{1}{F_{[\alpha;m,n]}}$$

- Applications: To model ratio of sample variances
  In the F-test for regression and analysis of variance

- Generation: Characterization
  Generate two chi-square variates $\chi^2(n)$ and $\chi^2(m)$ and compute:

$$F(n,m) = \frac{\chi^2(n)/n}{\chi^2(m)/m}$$

Gamma Distribution

- Generalization of Erlang distribution
  Allows noninteger shape parameters
- Key Characteristics:

  1. Parameters:
     a = Scale parameter, $a > 0$
     b = Shape parameter, $b > 0$
  2. Range: $0 \leq x \leq \infty$
  3. pdf: $f(x) = \frac{\left(\frac{x}{a}\right)^{b-1} e^{-x/a}}{a\Gamma(b)}$
     $\Gamma(.)$ is the gamma function.
  4. Mean: ab
  5. Variance: a^2b.

- Applications: To model service times and repair times
- Generation:
  1. If b is an integer, the sum of b exponential variates has a gamma distribution.
     $$\gamma(a, b) \sim -a \ln \left[\prod_{i=1}^{b} u_i \right]$$
  2. For $b < 1$, generate a beta variate $x \sim \mathrm{BT}(b, 1-b)$ and an exponential variate $y \sim \mathrm{Exp}(1)$. The product axy has a gamma(a,b) distribution.
  3. For non-integer values of b:
     $$\gamma(a, b) \sim \gamma(a, \lfloor b \rfloor) + \gamma(a, b - \lfloor b \rfloor)$$

Geometric Distribution

- Discrete equivalent of the exponential distribution
- Key characteristics:
  1. Parameters: p = Probability of success, $0 < p < 1$.
  2. Range: $x = 1, 2, \ldots, \infty$
  3. pmf: $f(x) = (1-p)^{x-1}p$
  4. CDF: $F(x) = 1 - (1-p)^x$
  5. Mean: $1/p$
  6. Variance: $\frac{1-p}{p^2}$
- memoryless
- Applications: Number of trials up to and including the first success in a sequence of Bernoulli trials
  Number of attempts between successive failures (or successes)

1. Number of local queries to a database between successive accesses to the remote database.
2. Number of packets successfully transmitted between those requiring a retransmission.
3. Number of successive error-free bits between in-error bits in a packet received on a noisy link.

Also to model batch sizes with batches arriving in a Poisson stream

- Generation: Inverse transformation
  Generate $u \sim$ U(0,1) and compute:

$$G(p) = \left\lceil \frac{\ln(u)}{\ln(1-p)} \right\rceil$$

$\lceil . \rceil \Rightarrow$ rounding up

Lognormal Distribution

- Log of a normal variate
- Key characteristics:
  1. Parameters:
     μ = Mean of $\ln(x)$, $\mu > 0$
     σ = Standard deviation of $\ln(x)$, $\sigma > 0$
  2. Range: $0 \leq x \leq \infty$
  3. pdf: $f(x) = \frac{1}{\sigma x \sqrt{2\pi}} e^{\frac{-(\ln x - \mu)^2}{2\sigma^2}}$
  4. Mean: $e^{\mu + \sigma^2/2}$
  5. Variance: $e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$
- Note: μ and σ are the mean and standard deviation of $\ln(x)$ not x

- Applications: The product of a large number of positive random variables tends to have an approximate lognormal distribution
  To model multiplicative errors that are a product of effects of a large number of factors
- Generation: Log of a normal variate
  Generate $x \sim N(0, 1)$ and return $e^{\mu+\sigma x}$.

Negative Binomial Distribution

- Number of failures x before the m^{th} success
- Key characteristics:

1. Parameters:

$$\begin{aligned} p &= \text{Probability of success,} \\ & \quad 0 < p < 1 \\ m &= \text{Number of successes,} \\ & \quad m \text{ must be a positive integer.} \end{aligned}$$

2. Range: $x = 0, 1, 2, \ldots, \infty$
3. pmf:

$$f(x) = \binom{m+x-1}{m-1} p^m (1-p)^x = \frac{\Gamma(m+x)}{(\Gamma m)(\Gamma x)} p^m (1-p)^x$$

   The second expression allows a negative binomial to be defined for noninteger values of x.

4. Mean: $m(1-p)/p$

5. Variance: $m(1-p)/p^2$

- Applications:

1. Number of local queries to a database system before m^{th} remote query.
2. Number of retransmissions for a message consisting of m packets.
3. Number of error-free bits received on a noisy link before the m in-error bit.

Used if variance > mean
Otherwise use Binomial or Poisson.

- Generation:

1. Generate $u_i \sim U(0,1)$ until m of the u_i's are greater than p. Return the count of u_i's less than or equal to p as NB(p, m).
2. The sum of m geometric variates G(p) gives the total number of trials for m

successes

$$NB(p, m) \sim \left(\sum_{i=1}^{m} G(p)\right) - m$$

3. Composition:
   (a) Generate a gamma variate $y \sim \Gamma(p/(1-p), m)$
   (b) Generate a Poisson variate $x \sim \text{Poisson}(y)$
   (c) Return x as $\text{NB}(p, m)$

Normal Distribution

- Also known as Gaussian distribution
- Discovered by Abraham De Moivre in 1733
- Rediscovered by Gauss in 1809 and by Laplace 1812
- N(0,1) = unit normal distribution or standard normal distribution.
- Key characteristics:
  1. Parameters:
     μ = Mean
     σ = Standard deviation $\sigma > 0$
  2. Range: $-\infty \leq x \leq \infty$
  3. pdf: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$
  4. Mean: μ
  5. Variance: σ^2

- Applications:

  1. Errors in measurement.
  2. Error in modeling to account for a number of factors that are not included in the model.
  3. Sample means of a large number of independent observations from a given distribution.

- Generation:

  1. Using the sum of a large number of uniform $u_i \sim U(0,1)$ variates:

  $$N(\mu, \sigma) \sim \mu + \sigma \frac{(\Sigma_{i=1}^{n} u_i) - \frac{n}{2}}{\left(\frac{n}{12}\right)^{1/2}}$$

     Generally, $n = 12$ is used.

  2. Box-Muller Method: Generate two uniform variates u_1 and u_2 and compute two independent normal

variates N(μ, σ) as follows:

$$x_1 = \mu + \sigma cos(2\pi u_1)\sqrt{-2\ln(u_2)}$$
$$x_2 = \mu + \sigma sin(2\pi u_1)\sqrt{-2\ln(u_2)}$$

There is some concern that if this method is used with u's from an LCG, the resulting x's may be correlated.

3. Polar Method:
   (a) Generate two U(0,1) variates u_1 and u_2.
   (b) Let $v_1 = 2u_1 - 1$, $v_2 = 2u_2 - 1$, and $r = v_1^2 + v_1^2$.
   (c) If $r \geq 1$, go back to step 3a; otherwise let $s = \left(\frac{-2\ln r}{r}\right)^{1/2}$ and return.

$$x_1 = \mu + \sigma v_1 s$$
$$x_2 = \mu + \sigma v_2 s$$

   x_1 and x_2 are two independent N(μ, σ) variates.

4. Rejection Method:

(a) Generate two uniform U(0,1) variates u_1 and u_2.

(b) Let $x = -\ln u_1$.

(c) If $u_2 > e^{\frac{-(x-1)^2}{2}}$, go back to Step 4a.

(d) Generate u_3.

(e) If $u_3 > 0.5$, return $\mu + \sigma x$; otherwise return $\mu - \sigma x$.

Pareto Distribution

- Pareto CDF is a power curve
  $\Rightarrow$ Fit to observed data
- Key characteristics:
  1. Parameters: a=shape parameter, $a > 0$
  2. Range: $1 \leq x \leq \infty$
  3. pdf: $f(x) = ax^{-(a+1)}$
  4. CDF: $F(x) = 1 - x^{-a}$
  5. Mean: $\frac{a}{a-1}$, provided $a > 1$
  6. Variance: $\frac{a}{(a-1)^2(a-2)}$, provided $a > 2$
- Application: To fit a distribution
  The maximum likelihood estimate:
  $$a = \frac{1}{\frac{1}{n}\Sigma_{i=1}^{n} \ln x_i}$$
- Generation: Inverse transformation
  Generate $u \sim U(0,1)$ and return $1/u^{1/a}$.

Pascal Distribution

- Extension of the geometric distribution
- Number of trials up to and including the m^{th} success
- Key characteristics:
  1. Parameters:
     p = Probability of success, $0 < p < 1$
     m = Number of successes, m should be a positive integer.
  2. Range: $x = m, m+1, \ldots, \infty$
  3. pmf: $f(x) = \binom{x-1}{m-1} p^m (1-p)^{x-m}$
  4. Mean: m/p
  5. Variance: $m(1-p)/p^2$

- Applications:
  1. Number of attempts to transmit an m packet message.
  2. Number of bits to be sent to successfully receive an m-bit signal.
- Generation: Generate m geometric variates G(p) and return their sum as Pascal(p, m).

Poisson Distribution

- Limiting form of the binomial distribution
- Key characteristics:
  1. Parameters: λ = Mean, $\lambda > 0$
  2. Range: $x = 0, 1, 2, \ldots, \infty$
  3. pmf: $f(x) = P(X = x) = \lambda^x \frac{e^{-\lambda}}{x!}$
  4. Mean: λ
  5. Variance: λ
- Applications: To model the number of arrivals over a given interval
  1. Number of requests to a server in a given time interval t.
  2. Number of component failures per unit time.
  3. Number of queries to a database system over t seconds.
  4. Number of typing errors per form.

Particularly appropriate if the arrivals are from a large number of independent sources

- Generation:

1. Inverse Transformation Method:
   Compute the CDF F(x) for $x = 0, 1, 2, \ldots$ up to a suitable cutoff and store in an array.
   For each Poisson random variate, generate a U(0,1) variate u, and search the array to find x such that $F(x) \leq u < F(x+1)$, return x.
2. Starting with $n = 0$, generate $u_n \sim U(0,1)$ and compute the product $\Pi_{i=0}^{n} u_i$. As soon as the product becomes less than $e^{-}\lambda$, return n as the Poisson(λ) variate.
   Note that n is such that
   $u_0 u_1 \cdots u_{n-1} > e^{-}\lambda \geq u_0 u_1 \cdots u_n$

Student's t-Distribution

- Derived by W. S. Gosset (1876-1937)
  Published under a pseudonym of 'Student'
  Used symbol t
- Key characteristics:
  1. Parameters: ν=Degrees of freedom,
     ν must be a positive integer.
  2. Range: $-\infty \leq x \leq \infty$
  3. pmf:
     $f(x) = \frac{\{\Gamma[(\nu+1)/2]\}[1+(x^2/\nu)]^{-(\nu+1)/2}}{(\pi\nu)^{1/2}\Gamma(\nu/2)}$
  4. Variance: $\nu/(\nu-2)$, for $\nu > 2$.

$$\frac{N(0,1)}{\sqrt{\chi^2(\nu)/\nu}} \sim t(\nu)$$

- For $(\nu > 30)$, a $t \approx N(0,1)$

- Applications: In setting confidence intervals and in t-tests
- Generation: Characterization Generate $x \sim N(0,1)$ and $y \sim \chi^2(\nu)$ and return $x/\sqrt{y/\nu}$ as t(ν).

Uniform Distribution (Continuous)

- Key characteristics:
  1. Parameters: a = Lower limit
     b = Upper limit, $b > a$
  2. Range: $a \leq x \leq b$
  3. pdf: $f(x) = \frac{1}{b-a}$
  4. CDF: $F(x) = \begin{cases} 0, & \text{If } x < a \\ \frac{x-a}{b-a}, & \text{If } a \leq x < b \\ 1, & \text{If } b \leq x \end{cases}$
  5. Mean: $\frac{a+b}{2}$
  6. Variance: $(b-a)^2/12$
- Applications: Bounded random variables with no further information:
  1. Distance between source and destinations of messages on a network.
  2. Seek time on a disk.

- Generation: To generate $U(a, b)$, generate $u \sim U(0, 1)$ and return $a + (b - a)u$.

Uniform Distribution (Discrete)

- Discrete version of the uniform distribution
- Takes a finite number of values, each with the same probability.
- Key characteristics:

1. Parameters:
   m = Lower limit; m must be an integer.
   n = Upper limit; n must be an integer
   $n > m$
2. Range: $x = m, m+1, m+2, \ldots, n$
3. pmf: $f(x) = \frac{1}{n-m+1}$
4. CDF: $F(x) = \begin{cases} 0, & \text{If } x < m \\ \frac{x-m+1}{n-m+1}, & \text{If } m \le x < n \\ 1, & \text{If } n \le x \end{cases}$

5. Mean: $(n+m)/2$
6. Variance: $\frac{(n-m+1)^2-1}{12}$

- Applications:
  1. Track numbers for seeks on a disk.
  2. I/O device number selected for the next I/O.
  3. The source and destination node for the next packet on a network.
- Generation: To generate UD(m,n), generate $u \sim U(0,1)$, return $\lfloor m+(n-m+1)u \rfloor$.

Weibull Distribution

- Key characteristics:
  1. Parameters:
     $$a = \text{Scale parameter } a > 0$$
     $$b = \text{Shape parameter } b > 0$$
  2. Range: $0 \leq x \leq \infty$
  3. pdf: $f(x) = \frac{bx^{b-1}}{a^b} e^{-(x/a)^b}$
  4. CDF: $F(x) = 1 - e^{-(x/a)^b}$
  5. Mean: $\frac{a}{b}\Gamma(1/b)$
  6. Variance: $\frac{a^2}{b^2}\left[2b\Gamma(2/b) - \{\Gamma(1/b)\}^2\right]$
- If $b = 3.602$, the Weibull distribution is close to a normal. For $b > 3.602$, it has a long left tail. For $b < 3.602$, it has a long right tail.
  For $b \leq 1$, the Weibull pdf is L-shaped, and for $b > 1$, it is bell-shaped.

For large b, the Weibull pdf has a sharp peak at the mode.

- Applications: To model lifetimes of components.
  $b < 1 \Rightarrow$ failure rate increasing with time
  $b > 1 \Rightarrow$ failure rate decreases with time
  $b = 1 \Rightarrow$ failure rate is constant
  $\Rightarrow$ life times are exponentially distributed.
- Generation: Inverse transformation
  Generate $u \sim U(0, 1)$ and return
  $a(\ln u)^{1/b}$ as Weibull(a, b).

Relationships Among Distributions

Relationships Among Distributions

Exercise 29.1

W hat distribution would you use to model the following:

1. Number of requests between typing errors, given that each request has a certain probability of being in error?
2. Number of requests in error among m requests, given that each request has a certain probability of being in error?
3. The minimum or the maximum of a large set of IID observations?
4. The mean of a large set of observations from uniform distribution?
5. The product of a large set of observatiosn from uniform distribution?
6. To empirically fit the distribution using a power curve for CDF?

7. The stream resulting from a merger of two Poisson streams?
8. Sample variances from a normal population?
9. Ratio of two sample variances from normal population?
10. Time between successive arrivals, given that the arrivals are memoryless?
11. Service time of a device that consists of m memoryless servers in series?
12. Number of systems that are idle in a distributed system, given that each system has a fixed probability of being idle?
13. Fraction of systems that are idle in a distributed system, given that each system has a fixed probability of being idle?

Exercise 29.2

L et x,y,z,w be four unit normal variates. Find the distribution and 90-percentiles for the following quantities:

1. $(x + y + z + w)/4$
2. $x^2 + y^2 + z^2 + w^2$
3. $(x^2 + y^2)/(z^2 + w^2)$
4. $w/\sqrt{(x^2 + y^2 + z^2)/4}$

Further Reading

- Books on simulations: Law and Kelton (1982) and Brately, Fox, and Schrage (1986)
- Lavenberg (1983): transient removal, variance estimation, and random-number generation.
- Languages: GPSS in O'Donovan (1980)
SIMSCRIPT II in CACI (1983)
SIMULA by Birtwistle, Dahl, Myhrhaug, and Nygaard (1973)
GASP by Pritsker and Young (1975)
- Sherman and Browne (1973): trace-driven computer simulations
- Adam and Dogramaci (1979) include papers describing the simulation languages SIMULA, SIMSCRIPT, and GASP by their respective language designers.

Bulgren (1982) discusses SIMSCRIPT and GPSS.

- Event-set algorithms: Frata and Maly (1977), Wyman (1975), and Vaucher and Duval (1975).
- Mitrani (1982) and Rubinstein (1986): Variance reduction techniques.
- Random Number Generation: Knuth (1981) Vol. 2
  Greenberger (1961)
  Lewis, Goodman, and Miller (1969)
  Park and Miller (1988)
  Lamie (1987)
- Generalized feedback shift registers:
  Bright and Enison (1979)
  Fushimi and Tezuka (1983)
  Fushimi (1988), and Tezuka (1987)
  Golomb (1982)
- Kreutzer (1986): Ready-made Pascal

routines for common simulation tasks such as event scheduling, time advancing, random-number generation

- Distributions: Hastings and Peacock (1975)
- Distributed simulation and knowledge-based simulations: Unger and Fujimoto (1989)
  Webster (1989)

Current Areas of Research in Simulation

- Distributed simulations
- Knowledge-based simulations
- Simulations on microcomputers
- Object-oriented simulation
- Graphics and animation for simulations
- Languages for concurrent simulations.

Sequential Simulation

- The events are processed sequentially.
- Not efficient on parallel or multiprocessor systems
- Two global variables shared by all processes: the simulation clock and the event list.

Distributed Simulation

- Also known as concurrent simulation or parallel simulation
- Global clock times are replaced by several (distributed) "channel clock values"
- Events are replaced by messages between processes ıAllows splitting a simulation among an arbitrary number of computer systems
- Introduces the problem of deadlock ⇒ Schemes for deadlock detection, deadlock recovery, and deadlock prevention
- Survey by Misra (1986)
- See also Wagner and Lazowska (1989).

Knowledge-based Simulations

- Artificial intelligence techniques are used for simulation modeling.
- Allow specifying the system at a very high level
- Questions are interpreted intelligently by the simulation system
- Provide automatic verification and validation
- Automatic design of experiments, data analysis and interpretation See Ramana Reddy et al (1986) and Klahr and Fought (1980)

Bibliography

[1] N. R. Adam and A. Dogramaci, eds., *Current Issues in Computer Simulation*, Academic Press, New York, 1979.

[2] J. S. Annino and E. C. Russell, "The Ten Most Frequent Causes of Simulation Analysis Failure," CACI Report 7, 1979.

[3] G. Birtwistle, O. Dahl, B. Myhrhaug, and K. Nygaard, *SIMULA Begin*, Auerbach, Philadelphia, 1973.

[4] L. Blum, M. Blum, and M. Shub, "A Simple Pseudo-Random Number Generator," SIAM J. Comput. Vol. 15, No. 2, May 1986, pp. 364-383.

[5] P. A. Bobillier, B. C. Kahan, and A. R. Probst, *Simulation with GPSS and GPSS V*, Prentice-Hall, Englewood-Cliffs, NJ, 1976.

[6] G. E. P. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates," *Ann. Math. Stat.*, Vol. 29, 1958, pp. 610-611.

[7] P. Bratley, B. L. Fox, and L. E. Schrage, *A Guide to Simulation*, Springer-Verlag, New York, 1986.

[8] H. S. Bright and R. L. Enison, "Quasi-Random Number Sequences from a Long-Period TLP Generator with Remarks on Application to Cryptography," ACM Comput. Surveys, Vol. 11, 1979, pp. 357-370.

[9] R. Brown, "Calendar Queues: A Fast O(1) Priority Queue Implementation for the Simulation Event Set Problem," Comm. of ACM, Vol. 31, No. 10, October 1988, pp. 1220-1227.

[10] W. G. Bulgren, *Discrete System Simulation*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[11] C.A.C.I., *SIMSCRIPT II.5 Programming Language*, C. A. C. I., Los Angeles, CA, 1983.

[12] R. R. Conveyou and R. D. McPherson, "Fourier Analysis of Uniform Random Number Generators," Journal of ACM, Vol 14, 1967, pp. 100-119.

[13] M. A. Crane and A. J. Lemoine, *An Introduction to the Regenerative Method for Simulation Analysis*, Springer-Verlag, New York, 1977.

[14] O-J. Dahl, B. Myhrhaug, and K. Nygaard, *Common Base Language*, Norwegian Computing Center, Oslo, Norway, 1982.

[15] R. L. Edgeman, "Random Number Generators and the Minimal Standard," Communications of ACM, Vol. 32, No. 8, August 1989, pp. 1020-21.

[16] G. S. Fishman and L. R. Moore, "An Exhaustive Analysis of Multiplicative Congruential Random Number Generators with Modulus 2^{31}-1," SIAM J. on Sci. Statist. Comput., Vol 7, 1986, pp. 24-45.

[17] B. L. Fox, "Generation of Random Samples from the Beta and F distributions," *Technometrics*, Vol. 5, 1963, pp. 269-270.

[18] W. R. Franta and K. Maly, "An Efficient Data Structure for the Simulation Event Set," Communications of ACM, Vol. 20, No. 8, August 1977, pp. 596-602.

[19] W. R. Franta, *The Process View of Simulation*, North-Holland, New York, 1977.

[20] A. M. Frieze, R. Kannan, and J. C. Lagarias, "Linear Congruential Generators Do Not Produce Random Sequences," Proc. 25th Symp. on Foundations of Computer Sci., Boca Raton, FL, October 24-26, 1984, pp. 480-484.

[21] M. Fushimi and S. Tezuka, "The k-Distribution of Generalized Feedback Shift Register Pseudorandom Numbers," Communications of ACM, Vol. 26, No. 7, July 1983, pp. 516-523.

[22] M. Fushimi, "Designing a Uniform Random Number Generator Whose Subsequences are k-Distributed," SIAM J. Comput., Vol. 17, No. 1, February 1988, pp. 89-99.

[23] S. W. Golomb, *Shift Register Sequences*, Aegean Park Press, Laguna Hills, CA, 1982.

[24] M. Greenberger, "An A Priori Determination of Serial Correlation in Computer Generated Random Numbers," Math. Comp., Vol. 15, 1961, pp. 383-389.

[25] C. Hastings, Jr. *Approximations for Digital Computers*, Princeton University Press, Princeton, NJ, 1955.

[26] N. A. J. Hastings and J. B. Peacock, *Statistical Distributions*, Wiley, New York, 1975.

[27] IBM, *System/360 Scientific Subroutine Package, Version III, Programmer's Manual*, IBM, White Plains, NY, 1968, p. 77.

[28] *IMSL Library*, Vol. I, 8th Edn., Distributed by International Mathematical and Statistical Libraries, Inc., Houston, TX.

[29] R. K. Jain, "A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks," IEEE Journal on Selected Areas in Communications, Vol. SAC-4, No. 7, Oct. 1986, pp. 1162-1167.

[30] M. D. Jöhnk, "Erzeugung von Betaverteilten und Gammaverteilten Zufallszahlen," *Metrika*, Vol. 8, 1964, pp. 5-15.

[31] H. Katzan, Jr., *APL User's Guide*, Van Nostrand Reinhold, New York, 1971.

[32] P. Klahr and W. S. Fought, "Knowledge-Based Simulation," Proc. First Conf. AAAI, Stanford, CA, 1980, pp. 181-183.

[33] D. E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, Addison-Wesley, Reading, MA, 1981.

[34] W. Kreutzer, *System Simulation Programming Styles and Languages*, Addison-Wesley, Reading, MA, 1986.

[35] P. L'Ecuyer, "Efficient and Portable Combined Random Number Generators," Communications of ACM, Vol. 31, No. 6, June 1988, pp. 742-774.

[36] E. L. Lamie, *Pascal Programming*, Wiley, New York, 1987, p. 150.

[37] S. S. Lavenberg, ed., *Computer Performance Modeling Handbook*, Academic Press, New York, 1983.

[38] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, New York, 1982.

[39] A. M. Law, "Statistical Analysis of Simulation Output," Operations Research Vol.19, No. 6, pp. 983-1029, Nov.-Dec., 1983.

[40] D. H. Lehmer, "Mathematical Methods in Large-Scale Computing Units," Ann. Comput. Lab., Harvard Univ., Vol. 26, 1951, pp. 141-146.

[41] P. A. Lewis, A. S. Goodman, and J. M. Miller, "A Pseudo-Random Number Generator for the System/360," IBM Systems Journal, Vol. 8, No. 2, 1969, pp. 136-146.

[42] T. G. Lewis and W. H. Payne, "Generalized Feedback Shift Register Pseudo-Random Number Algorithm," Journal of ACM, Vol. 20, No. 3, July 1973, pp. 456-468.

[43] H. M. Markowitz, B. Hausner, and H. W. Karr, *SIMSCRIPT: A Simulation Programming Language*, Prentice-Hall, Englewood Cliffs, NJ, 1963.

[44] G. Marsaglia and T. A. Bray, "A Conveniently Method for Generating Normal Variables," *SIAM Rev.*, Vol. 6, 1964, pp. 260-264.

[45] G. Marsaglia, "Random Numbers Fall Mainly in the Planes," Proc. Nat. Acad. Sci., Vol. 60, No. 5, September 1968, pp. 25-28.

[46] G. Marsaglia, "Random Number Generation," in A. Ralston and E. D. Reilly, Jr., Eds, *Encyclopedia of Computer Science and Engineering*, Van Nostrand Reinhold, New York, 1983, pp. 1260-1264.

[47] W. M. McCormack and R. G. Sargent, "Comparison of Future Event Set Algorithms for Simulations of Closed Queueing Systems," in N. R. Adam and A. Dogramaci (Eds), *Current Issues in Computer Simulation*, Academic Press, New York, 1979, pp. 71-82.

[48] J. Misra, "Distributed Discrete-Event Simulation," ACM Computing Surveys, Vol. 18, No. 1, March 1986, pp. 39-66.

[49] I. Mitrani, *Simulation Techniques for Discrete-Event Systems*, Cambridge U. Press, London, 1982.

[50] T. M. O'Donovan, *GPSS Simulation Made Simple*, Wiley, Chichester, U.K., 1980.

[51] S. K. Park and K. W. Miller, "Random Number Generators: Good Ones Are Hard to Find," Communications of ACM, Vol. 31, No. 10, October 1988, pp. 1192-1201.

[52] S. Pasupathy, "Glories of Gaussianity," IEEE Communications Magazine, Vol. 27, No. 8, August 1989, pp. 37-38.

[53] Prime Computer, *Subroutines Reference Guide*, 3rd Ed, 1984, p. 12.45.

[54] A. Pritsker and R. E. Young, *Simulation with GASP - PL/I: A PL/I Based Continuous/Discrete Simulation Language*, Wiley-Interscience, New York, 1975.

[55] Y. V. Ramana Reddy, M. S. Fox, N. Husain, and M. McRoberts, "The Knowledge-Based Simulation System," IEEE Software, March 1986, pp. 26-37.

[56] C. M. Reeves, "Complexity Analyses of Event Set Algorithms," The Computer Journal, Vol. 27, No. 1, 1984, pp. 72-79.

[57] R. Y. Rubinstein *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks*, Wiley, New York, 1986.

[58] M. Santha and U. V. Vazirani, "Generating Quasi-Random Sequences from Slightly Random Sources," Proc. 25th Symp. on Foundations of Computer Sci., Boca Raton, FL, October 24-26, 1984, pp. 434-440.

[59] L. Schrage, "A More Portable FORTRAN Random Number Generator," ACM Transactions on Mathematical Software, Vol. 5, No. 2, June 1979, pp. 132-138.

[60] S. W. Sherman and J. C. Browne, "Trace-Driven Modeling: Review and Overview," Proc. Symp. on the Simulation of Computer Systems, pp. 201-207, June, 1973.

[61] R. C. Tausworthe, "Random Numbers Generated by Linear Recurrence Mod Two," Math. Comput. Vol. 19, 1965, pp. 201-209.

[62] S. Tezuka, "Walsh-Spectral Test for GFSR Pseudorandom Numbers," Communications of ACM, Vol. 30, No. 8, August 1987, pp. 731-735.

[63] J. P. R. Tootill, W. D. Robinson, and A. G. Adams, "The Runs Up and Down Performance of Tausworthe Pseudo-Random Number Generators," Journal of ACM, Vol. 18, 1971, pp. 381-399.

[64] J. P. R. Tootill, W. D. Robinson, and D. J. Eagle, "An Asymptotically Random Tausworthe Sequence," Journal of ACM, Vol. 20, No. 3, July 1973, pp. 469-481.

[65] B. Unger and R. Fujimoto, Eds., *Distributed Simulation, 1989*, The Society for Computer Simulation, San Diego, CA, 1989, 204 pp.

[66] J. G. Vaucher and P. Duval, "A Comparison of Simulation Event List Algorithms," Communications of ACM, Vol. 18, No. 4, April 1975, pp. 223-230.

[67] U. V. Vazirani and V. V. Vazirani, "Efficient and Secure Pseudo-Random Number Generation," Proc. 25th Symp. on Foundations of Computer Sci., Boca Raton, FL, October 24-26, 1984, pp. 458-463.

[68] D. B. Wagner and E. D. Lazowska, "Parallel Simulation of Queueing Networks: Limitations and Potentials," Proc. SIGMETRICS'89, May 23-26, 1989, Berkeley, CA. (Also published as Performance Evaluation Review, Vol. 17, No. 1, May 1989), pp. 146-155.

[69] W. Webster, Ed., *Simulation and AI, 1989*, Society for Computer Simulations, San Diego, CA, 1989, 139 pp.

[70] F. P. Wyman, "Improved Event Scanning Mechanisms for Discrete-Event Simulation," Communications of ACM, Vol. 18, No. 6, June 1975, pp. 350-353.

