Computer system modeling and simulation

4. Random variable generation

Sosina M.
Addis Ababa institute of technology (AAiT)
2012 E.C.

Random variables generation

- Modeling activities that are unpredictable
  - Example inter-arrival times and service times at queues
  - Such variables are modeled as random variables with some specified statistical distribution
- Activity of generating samples from a specified distribution
- IID U[0, 1] are the basic ingredient needed for any random variable generation

Random variables generation

- Methods
  - Inverse transform technique
  - Acceptance-rejection technique
  - Composition method
  - Convolution method

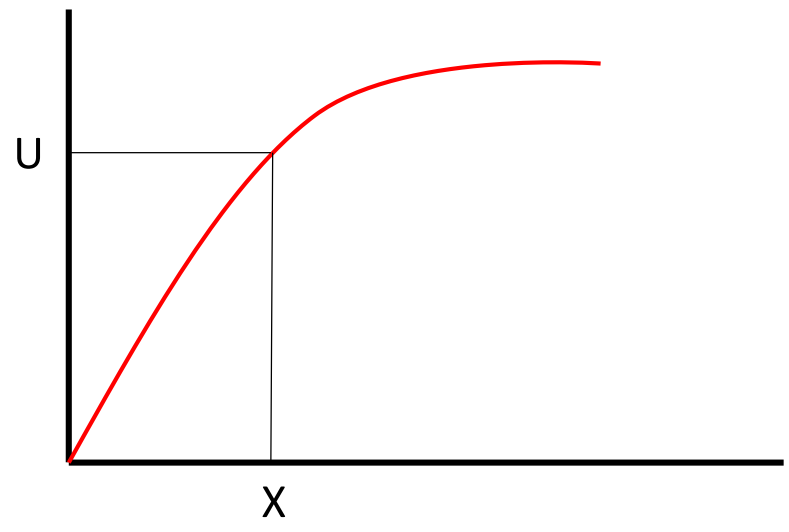
- All the methods assume that a source of uniform [0, 1] random numbers is readily available

Requirements

- Exactness
  - Use methods that result in random variates with exactly the desired distribution
- Efficiency
  - Storage space and execution time
- Simplicity
  - Conceptual and implementational factors
- Mathematical validity

Inverse-transform technique

- We want to generate instances of a random variable X with cumulative distribution F(x)
- *Algorithm*
  - *U=F(x)*
  - *Generate U~U(0, 1)*
  - *Return* $X=F^{-1}(U)$
- Can be utilized for any distribution – but most useful when F^{-1} can be computed easily



Exponential distribution

- $F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$
  - Λ = the mean number of occurrences per time unit
- Example: *exponential arrival times*
  - Inter-arrival times X_1, X_2, X_3
  - Λ=mean arrivals per time unit (rate of arrival)
  - $E(X_i) = 1/\lambda$

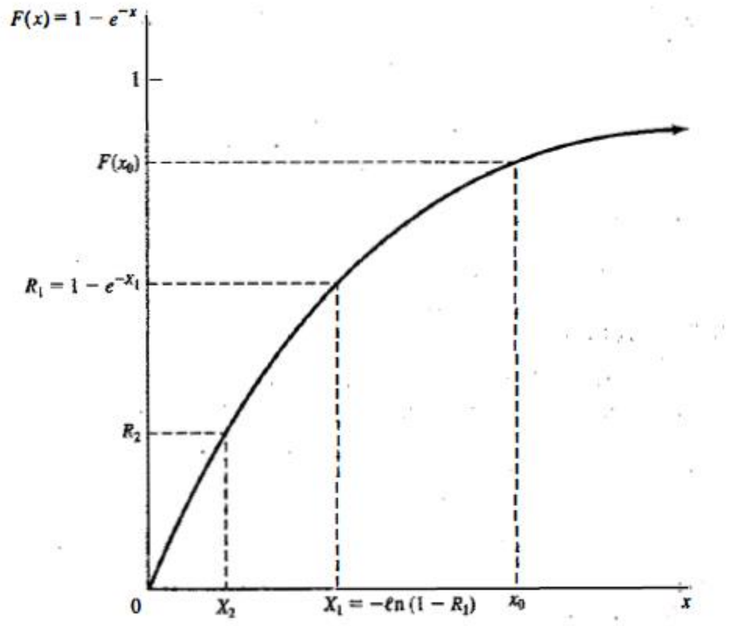
Exponential distribution

1. Compute the cdf ⟶ $F(x) = 1 - e^{-\lambda x}, x \geq 0$
2. Set F(x)=$1 - e^{-\lambda x}$=R

   R has a uniform distribution over the interval [0, 1]
3. $X = -\frac{1}{\lambda}\ln(1 - R)$

   both R and 1-R are uniformly distributed in [0, 1] =>

   $X = -\frac{1}{\lambda}\ln R$

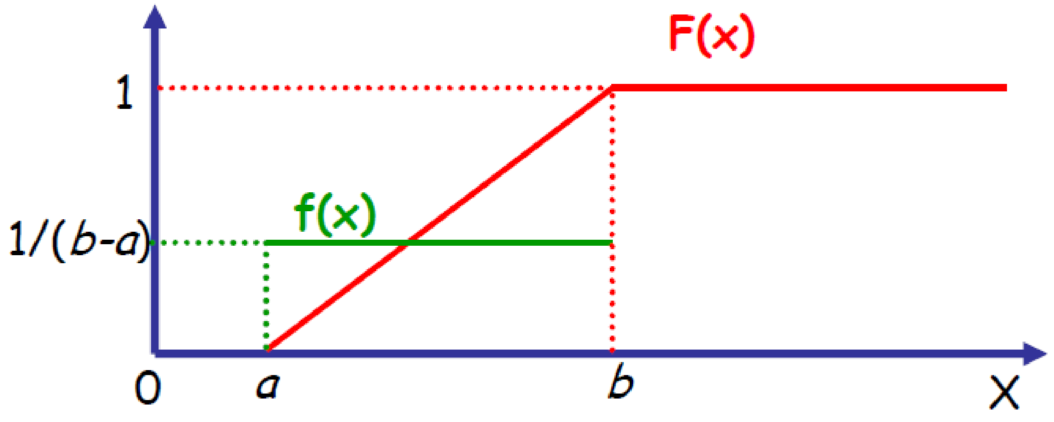


❑ ***Algorithm***

- *Generate u~U(0, 1)*
- *Calculate* $X = -\frac{1}{\lambda} ln\, u$
- *Return X*

Uniform distribution

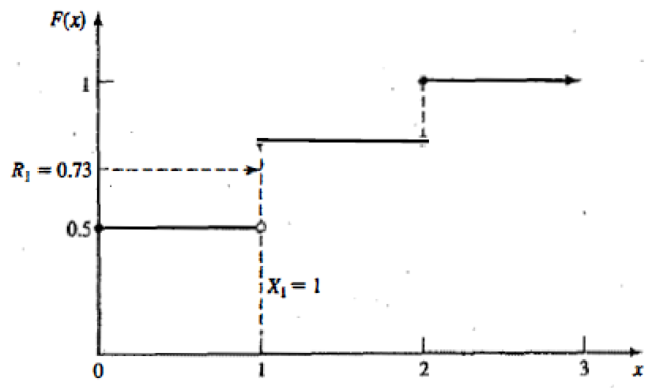
- We want to generate instances of a uniform random variable on the interval [a, b]
  - $F(x) = \frac{x-a}{b-a}\ , \quad a \le x \le b$



- ***Algorithm***
  - *Generate u~U(0, 1)*
  - *Calculate X = u(b-a)+a*
  - *Return X*

Discrete distribution

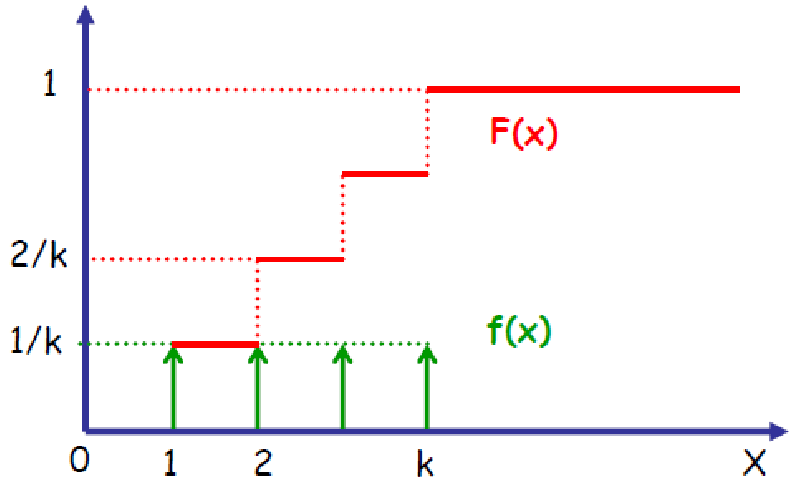
- All discrete distributions can be generated via the inverse-transform technique
- ***Empirical discrete distribution***
  - Example
    - P(X=0)=0.5, P(X=1)=0.3, P(X=2)=0.2
    - $F(x)=\begin{cases} 0 & x < 0 \\ 0.5 & 0 \le x < 1 \\ 0.80 & 1 \le x < 2 \\ 1 & x \ge 2 \end{cases}$ $\quad X=\begin{cases} 0 & R \le 0.5 \\ 1 & 0.5 < R \le 0.8 \\ 2 & 0.8 < R \le 1 \end{cases}$



Discrete uniform distribution

- Generate an instance x of X with x in {1, 2, …,k) where P(x)=1/k, x=1,2,…,k

- F(x)= $\begin{cases} 0 & x < 1 \\ \frac{1}{k} & 1 \leq x < 2 \\ \frac{2}{k} & 2 \leq x < 3 \\ & \ldots. \\ 1 & x \geq k \end{cases}$ X= $\lceil ku \rceil$



- $\frac{x-1}{k} < u \leq \frac{x}{k} \Rightarrow x - 1 < ku \leq x = ku \leq x < ku + 1$

Geometric

- Consider a geometric distribution with pmf

$$p(x) = p(1-p)^x$$

$$F(x) = \sum_{j=0}^{x} p(1-p)^j = 1-(1-p)^{x+1}$$

$$\mathrm{X} = \frac{\ln(1-R)}{\ln(1-p)} - 1 \Rightarrow X = \left\lceil \frac{\ln(R)}{\ln(1-p)} - 1 \right\rceil$$

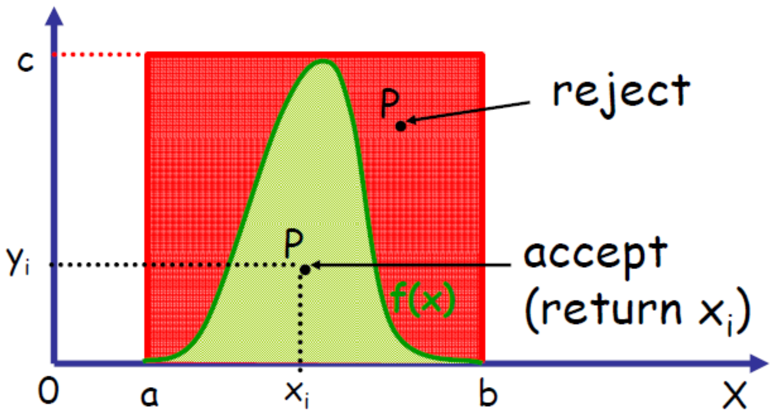
- Algorithm
  - Generate R=U(0, 1)
  - Compute X=ceil ($\frac{\ln(R)}{\ln(1-p)} - 1$)
  - Return X

Remarks

- Intuitively easy to understand
- To apply inverse-transform technique F(x) must be invertible
- The computational cost depends on the computational complexity of the inverse function

Acceptance-rejection technique

- The acceptance rejection technique can be applied to random variables with pdf f(x) and limited support [a, b]
- c=max(f(x)), apply the following procedure
  - *Generate xi=U(a, b)*
  - *Generate yi=U(0, c)*
  - *If yi≤ f (xi) return xi, otherwise go back to step 1*



Acceptance-rejection technique

- The efficiency of an acceptance-rejection technique depends heavily on being able to minimize the number of rejections

Composition method

- This method is applied to random variables whose CDF can be expressed as weighted sum of other CDFs

$$F(x) = \sum_{i=1}^{\infty} p_i F_i(x) \quad p_i\text{=the probability of generating from } F_i$$

- ***Poisson distribution***

$$P(N=n)=\frac{e^{-\lambda}\lambda^n}{n!}, \quad n=0, 1, 2, \ldots$$

  - N=the number of Poisson arrivals in a unit time interval => the interarrival times $A_1, A_2, \ldots$ will be exponentially distributed with mean $1/\lambda$
  - N=n, iff

$$A_1 + A_2 + \cdots + A_n \leq 1 < A_1 + A_2 + \cdots + A_{n+1}$$

Poisson distribution

- $A_i = -\frac{1}{\lambda}\ln R$

  $\sum_{i=1}^{n} -\frac{1}{\lambda}\ln R_i \le 1 < \sum_{i=1}^{n+1} -\frac{1}{\lambda}\ln R_i$

  $\ln\prod_{i=1}^{n} R_i \ge -\lambda > \ln\prod_{i=1}^{n+1} R_i$

  $\prod_{i=1}^{n} R_i \ge e^{-\lambda} > \prod_{i=1}^{n+1} R_i$

Poisson distribution

- The procedure for generating a Poisson random variate N
  - *Set n=0, p=1*
  - *Generate a random number* R_{n+1}, *and replace p=p.* R_{n+1}
  - *If p<*$e^{-\lambda}$*, then accept N=n, otherwise reject the current n, increase n by one and return to step 2*

- To generate one Poisson variate N=n, on average n+1 random number will be required

Poisson distribution

- ***Example***
  - *Generate three Poisson variates with mean λ=0.2,* $e^{-0.2} = 0.8187$
    R values = [0.4357, 0.4146, 0.8353, 0.9952, 0.8004]

Convolution method

- Convolution method refers to adding together two or more random variable to obtain a new random variable with the desire distribution

- ***Erlang-k*** – the sum of k random variable with exponential distribution
  - The Erlang-k with mean 1/ λ is the sum of k exponential random variables with mean 1/k λ

$$X = \sum_{i=1}^{k} X_i$$

Erlang-k distribution

- To generate X, generate $X_1, X_2, \dots, X_k$ and sum them

$$X = \sum_{i=1}^{k} -\frac{1}{k\lambda}\ln(R_i) = -\frac{1}{k\lambda}\ln\prod_{i=1}^{k} R_i$$

- K random numbers are required
- If k is large, it might be inefficient