Geração de Números Randômicos na Simulação

ADS29009-Avaliação de Desempenho de Sistemas

Eraldo Silveira e Silva

24 de junho de 2025

INSTITUTO FEDERAL

Outline

INSTITUTO FEDERAL

Bibliografia para esta aula

- William J.Stewart. Probability, Markov Chains, Queues and Simulation.
- David J.Lilja. Measuring Computer Performance. A practitioner's guide.
- Michael K.Molly. Fundamentals of Performance Modeling.
- The Art Of Systems Performance Analysis. Raj Jain.1991.

Outline

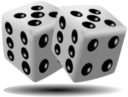
Introdução

Simulação de um PE

Para simular um processo estocástico em um programa de computador tem-se que que gerar sequências de valores (realizações) associados a uma variável randômica que possui determinadas propriedades.

Exemplo

Para simular o lançamento de dois dados (soma dos dois) tem-se que gerar os valores $2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$ em uma sequência imprevisível com frequência de $1/36, 1/18, 1/12, 1/9, 5/36, 1/6, 5/36, 1/9, 1/12, 1/18, 1/36$.



INSTITUTO FEDERAL

ATENÇÃO

É impossível gerar números randômicos de forma perfeita através de um programa de computador. O programa sempre será determinístico. O que se tem é PSEUDOGERADORES (PRNGs)que geram sequências que apresentam propriedades estatísticas. Quando executados na mesma condição inicial (**semente**), SEMPRE geram a mesma sequência.

☺

Os PRNGS são, no entanto, interessantes para uma **simulação**: Pode-se repetir o experimento obtendo-se os mesmos resultados desde que se use as mesmas sementes.

INSTITUTO FEDERAL

Está necessitando de um verdadeiro gerador de número randômico(TRNG)?

Veja o "free service" de www.random.org

Gera números a partir de ruídos atmosféricos...

Abordagem na geração PRNG

- gerar "pseudorandomicamente" sequências uniformemente distribuídas (usando alguma função). Pode ser $U(0,1)$ pois a partir desta pode-se gerar outras distribuições;
- validar a uniformidade da sequência aplicando alguma técnica;
- verificar a independência;
- usar a sequência para gerar outras distribuições;

INSTITUTO FEDERAL

Abordagem histórica: Método Midsquare (Von Neumann)

Inicia-se a geração escolhendo um número (semente) e eleva-se ao quadrado. Seleciona-se os dígitos do meio e repete-se-se o processo.

Exemplo

Toma-se o número 12. O quadrado é 0144. Toma-se 14 com quadrado 0196. Toma-se 19 e obtém-se 0361. Obtém-se portanto uma sequência como $12, 14, 19, 36, 29, 84, \ldots$

Problemas

Pode aparecer um 00 na seleção...

Características de um bom pseudogerador

- eficiente em termos computacionais;
- período longo: a sequência gerada é finita. Então o ciclo k deve ser longo: $x_{n+k} = x_n, x_{n+k+1} = x_{n+1} \ldots$
- independente e uniformemente distribuído;
- reprodutível (que pode ser repetido);

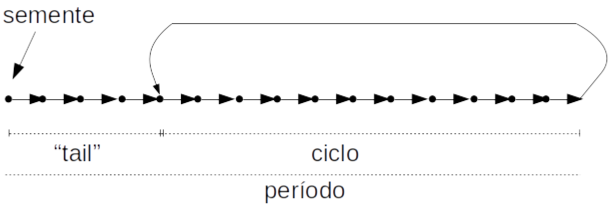


Figura: Modificado de (Jain,91)

Outline

INSTITUTO FEDERAL

Método Congruente Linear

Função Geradora

$$z_{n+1} = (az_n + c) mod\ \ m$$

- a,b e m são constantes a serem cuidadosamente escolhidas: a é o multiplicador, c é o incremento. O tamanho máximo (possível) de uma sequência é determinado por m.
- z_0 é a **semente** geradora.
- Se $c > 0$ o método é chamado misto congruente.
- Se $c = 0$ o método é chamado congruente multiplicativo.
- geração periódica garantida: quando a semente for reproduzida o ciclo se repete.

Método Congruente Linear

Tarefa em sala

- Implementar em C++ uma classe geradora de números randômicos usando o método congruente linear. Fazer uma função para setar a semente e outra para gerar (similar ao *srnd* e *rnd*)
- Testar com diferentes *seeds* (sementes) e parâmetros. Gerar com $a = 1103515245$, $c = 12345$, $m = 2147483648$, $seed = 0$ e comparar com outras equipes.
- Implementar uma função para descobrir o período (ciclo) do gerador.

Método Congruente Aditivo

Função Geradora

$$z_n = (z_{n-1} + z_{n-k}) mod\ m$$

Ou seja, z_n a base de geração é a soma do valor prévio de z_n com o *kth* valor.

INSTITUTO FEDERAL

Outline

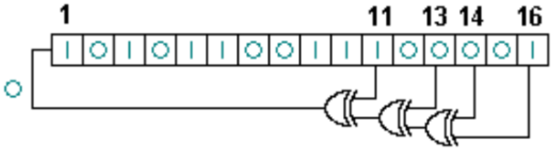
INSTITUTO FEDERAL

Linear-feedback shift register

- Baseado em um registrador de deslocamento cujo bit de entrada é uma função do estado atual do registro;
- Função normalmente usada é um XOR;
- Facilmente implementado em hardware.

INSTITUTO FEDERAL

Linear-feedback shift register



Fonte: Wikipedia

Mersenne Twister

- Baseado em um registrador de deslocamento;
- Longo período;
- Baixa correlação entre números sucessivos;
- Maior complexidade de implementação;
- Proposto por Matsumoto, Nishimura 1997.

INSTITUTO FEDERAL

Outline

Outline

4. Geração de Números Randômicos a partir de uma Distribuição Uniforme
    - Método da Função Inversa

Gerando Números Randômicos a partir de uma Distribuição Uniforme

Um problema que surge na simulação de modelos estocásticos é o da geração randômica de números com distribuição qualquer. Por exemplo como gerar números a partir de uma distribuição exponencial? Uma possibilidade é aplicar o método da função inversa da CDF:

Método da Inversão da CDF

Baseia-se no fato de que a a **variável aleatória** Y computada como $Y = F_x(X)$ a partir de uma CDF de uma **variável aleatória** X qualquer, é **uniforme** no intervalo $[0, 1]$.
Pode-se ter observações de X usando a inversa de sua CDF:

$$X = F_x^{-1}(Y)$$

Gerando Números Randômicos a partir de uma Distribuição Uniforme

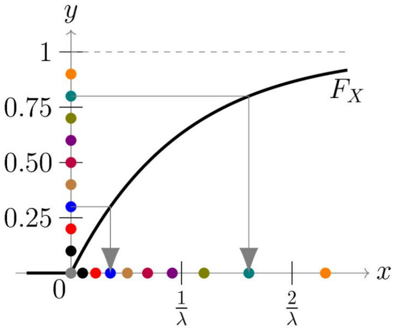


Figura: Mapeamento inverso de observações

[Fonte: By LarsWinterfeld - Own work, CC BY-SA 4.0, `https://commons.wikimedia.org/w/index.php?curid=50228282`]

INSTITUTO FEDERAL

Gerando Números Randômicos a partir de uma Distribuição Uniforme

Demonstração

Sabe-se que podemos construir variáveis aleatórias como função de outras variáveis. Imagine que X é uma V.A. com CDF $F_X(x)$. Seja Y uma V.A. elaborada a partir da CDF de X:

$$Y = F_X(X)$$

Por definição da CDF tem-se:

$$F_Y(y) = Prob\{Y \leq y\}$$

Se $F_X(.)$ possui inversa e sendo estritamente crescente tem-se que:

$$Prob\{Y \leq y\} = Prob\{F_X(X) \leq y\} = Prob\{F_X^{-1}(F_X(X)) \leq F_X^{-1}(y)\}$$

Gerando Números Randômicos a partir de uma Distribuição Uniforme

Cont. da Demonstração

Portanto:

$$F_Y(y) = Prob\{Y \leq y\} = Prob\{X \leq F_X^{-1}(y)\} \; para \; 0 \leq y \leq 1$$

Mas $Prob\{X \leq x\} = F_X(x)$, então:

$$F_Y(y) = F_X(F_X^{-1}(y)) = y \; para \; 0 \leq y \leq 1$$

O que caracteriza uma distribuição uniforme!!!

INSTITUTO FEDERAL

Gerando Números Randômicos a partir de uma Distribuição Uniforme

Problemas associados ao método

Pode ser difícil ou impossível obter a inversa da função...
Felizmente não é o caso de uma distribuição exponencial

Gerando Números Randômicos com Distribuição Exponencial com método da Inversa

Lembrando a PDF da Distribuição Exponencial:

$$f_X(x) = Pr[X = x] = \lambda e^{-\lambda x} \quad \lambda, x \geq 0 \tag{1}$$

E a CDF da Distribuição Exponencial:

$$F_X(x) = \int_0^x \lambda e^{-\lambda t} dt = 1 - e^{-\lambda x} \tag{2}$$

INSTITUTO FEDERAL

Gerando Números Randômicos a partir de uma Distribuição Uniforme

Invertendo a $F_X(x)$ da exponencial tem-se

$$x = \frac{ln(1 - F_X(x))}{-\lambda} \tag{3}$$

INSTITUTO FEDERAL

Gerando Números Randômicos a partir de uma Distribuição Uniforme

Observar que $1 - F_X(x)$ também é uniformemente distribuída então também pode-se também usar:

Invertendo a $F_X(x) =$ da exponencial tem-se

$$x = \frac{ln(y)}{-\lambda} \quad (4)$$

Ou seja, **pode-se gerar uma sequência uniformemente distribuída entre** $[0, 1]$ **e aplicar a equação acima para gerar números randômicos com distribuição exponencial**.

INSTITUTO FEDERAL

Gerando Números Randômicos a partir de uma Distribuição Uniforme

Tarefa em sala

- Acrescentar na classe criada anteriormente para geração de números randômicos, um gerador de números que seguem uma distribuição exponencial;

INSTITUTO FEDERAL

Outline

Validação de Sequências de Números Randômicos

- abordagem empírica: testes envolvendo várias gerações e aplicando testes estatísticos sobre os dados gerados;
- análise matemática das funções geradas (não será visto aqui);

Abordagem Empírica

Se enquadram em duas ctegorias:

- testes com objetivo de validar sequências distribuídas uniformemente em $[0, 1]$:
  - Teste Chi-square "goodness-of-fit";
  - Teste de Komolgorov-Smirnov, entre outras;
- testes com objetivo de verificar a independência na sequência: *run test*, *gap test* e poker test

INSTITUTO FEDERAL

Abordagem Empírica

testes para validar sequências distribuídas

- Utilizam-se de um nível de significância α (normalmente entre 0.01 e 0.05 que é a probabilidade de rejeitar a hipótese nula dado que ela é na realidade verdadeira:

$$\alpha = Prob\{H_0 \ ser \ rejeitada | H_0 \ é \ verdadeira\}$$

- Exemplo: Para $\alpha = 0.05$ tem-se:
  - Em 20 sequências espera-se rejeitar 1;
  - Em 100 sequências espera-se rejeitar 5;

INSTITUTO FEDERAL

Teste Chi-Square "Goodness-of-Fit"

Compara uma amostra de uma distribuição gerada com uma teórica (resultado teórico);

Particiona-se (*binning*) um intervalo de n números pseudorandômicos em k subintervalos iguais e compara-se a contagem de números em cada intervalo com contagem teórica n/k

Para ser significativo tem-se:

- $k >> 10$
- $n >> 10k$

Teste Chi-Square "Goodness-of-Fit"

A variável *Chi-Square*

Pode -se usar uma variável randômica definida da forma:

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - \bar{n}_i)}{\bar{n}_i}$$

Onde n_i e $\bar{n}_i$ são respectivamente a quantidade de números pesudorandômicos no *bin* i e a quantidade teórica no *bin* i ; A hipótese de que a sequência é uniformemente distribuída é provada quando:

$$Prob\{\chi^2 \leq \chi^2_\alpha\} = 1 - \alpha$$

Teste Chi-Square "Goodness-of-Fit"

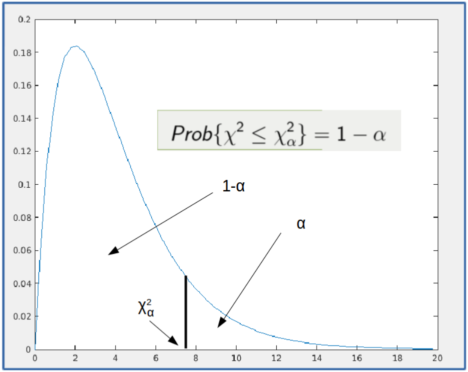


Figura: Significado do teste Chi-Square "Goodness-of-Fit"