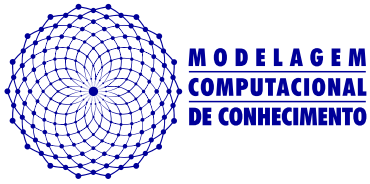

Dissertação de Mestrado

Geradores de Números Aleatórios, um estudo comparativo utilizando a linguagem R

Marcelo Queiroz de Assis Oliveira
marceloqao@gmail.com

Orientadores:

Dr. Alejandro C. Frery
Dr. Heitor Soares Ramos Filho

Maceió, Março de 2015

Marcelo Queiroz de Assis Oliveira

Geradores de Números Aleatórios, um estudo comparativo utilizando a linguagem R

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Orientadores:
Dr. Alejandro C. Frery

Dr. Heitor Soares Ramos Filho

Maceió, Março de 2015

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Fabiana Camargo dos Santos

T693n Oliveira, Marcelo Queiroz de Assis.

Geradores de Números Aleatórios, um estudo comparativo utilizando a linguagem R / Marcelo Queiroz de Assis Oliveira. – 2015
73 f. : il.

Orientadores: Alejandro C. Frery.
Heitor Soares Ramos Filho.
Dissertação (Mestrado em Modelagem Computacional de Conhecimento) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2015.

Bibliografia: f. 69–74.

1. Geradores de Números Aleatórios. 2. Testes Teóricos. 3. Testes Estatísticos.

CDU: 004.932:004.852 – VERIFICAR –

Membros da Comissão Julgadora da Dissertação de Mestrado de Marcelo Queiroz de Assis Oliveira, intitulada “Geradores de Números Aleatórios, um estudo comparativo utilizando a linguagem R”, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 30 de marco de 2015, às 10h00min, na sala de aula do Mestrado em Modelagem Computacional de Conhecimento. Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Curso de Mestrado em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

Dr. Alejandro C. Frery – Orientador
Instituto de Computação
Universidade Federal de Alagoas

Dr. Heitor Soares Ramos Filho – Orientador
Instituto de Computação
Universidade Federal de Alagoas

Dr. Osvaldo Anibal Rosso – Examinador
Pesquisador Visitante – Instituto de Computação
Universidade Federal de Alagoas

Dr. Raydonal Ospina Martínez – Examinador
Departamento de Estatística
Universidade Federal de Pernambuco

Maceió, Março de 2015

RESUMO

Este trabalho busca comparar “alguns” geradores de números aleatórios bem conhecidos pela comunidade acadêmica. Geradores de números aleatórios são amplamente utilizados em algoritmos probabilísticos como "Simulated Annealing", Algoritmos Genéticos e GRASP, assumindo assim um importante papel na utilização destes. Primeiramente falaremos sobre o histórico dos geradores de números aleatórios e de geradores em ambientes paralelos. Em seguida faremos a avaliação de alguns geradores segundo as abordagens clássica e numa segunda abordagem baseada na teoria da informação, procurando, desta forma, uma melhoria nos geradores de números pseudo aleatórios. Na sequência, discutiremos a avaliação paralela dos geradores utilizando três geradores disponíveis no R através dos trinta e um testes disponíveis no pacote "Dieharder", além de realizar um teste baseado da teoria da informação, encontrar regiões de confiança no plano (H,C) (Entropia x Complexidade) utilizando como dados de entrada as sequências aleatórias oriundas dos geradores testados. Como contribuição, proporemos uma ferramenta web para realizar os testes de sequências aleatórias.
Palavras-chave: Geradores de Números Aleatórios. Testes Teóricos. Testes Estatísticos.

ABSTRACT

This work aims to compare some Random Numbers Generators well known by the Scientific Community. Random Numbers Generators (RNG's) are used in probabilistic algorithms, such the Simulated Annealing, Genetics algorithms and GRASP. Also, the use of a robust RNG method is key factor to the success of a simulation. To introduce, some theoretic concepts about generators will be showed **Keywords**: Random Number Generators. Theoretical tests.

Statistical Tests.

AGRADECIMENTOS

Marcelo Queiroz de A. Oliveira

LISTA DE FIGURAS

LISTA DE TABELAS

Lista de Equações

SUMÁRIO

1

Introdução

1.1 Definição do Problema

1.2 Revisão Bibliográfica

1.3 Contribuições

1.4 Riscos

Neste capítulo tratamos de aspectos introdutórios do trabalho como a Definição do Problema, uma Revisão Bibliográfica, Contribuições e Riscos envolvidos no mesmo, no próximo capítulo trataremos da metodologia utilizada do desenvolvimento do mesmo.

2

Fundamentação

Este capítulo tem como objetivo apresentar a revisão bibliográfica necessária para a realização deste trabalho. Principalmente o fornecimento de subsídios suficientes e para que o leitor possa ser introduzido ao tema e se aprofunde nos conhecimentos abordados pelos autores citados e compreenda as contribuições propostas, sendo capaz de dar continuidade na pesquisa e desenvolvimento dos problemas deixados em aberto.

2.1 Quão bons são os geradores de Números Aleatórios Disponíveis no R ?

No R há alguns geradores de números pseudo aleatórios bem como um pacote de testes, conhecido como Dieharder, para aferir a sua qualidade. Há por outro lado, uma metodologia para análise de séries temporais baseada em ferramentas da teoria da informação que se mostrou útil para avaliar geradores de números pseudo aleatórios. A proposta é avaliar os geradores disponíveis no R comparando os testes disponíveis no Dieharder com o teste baseado em ferramentas da teoria da informação.

2.2 O que são PRNG's

2.2.1 Geradores de Números Aleatórios?

A necessidade de números aleatórios e pseudo-aleatórios surge em inúmeras aplicações, dentre elas a criptografia, modelagem e simulação. Existem dois tipos básicos de geradores utilizados para produzir sequências aleatórias: geradores de números aleatórios (RNGs) e geradores de números pseudo-aleatórios (PRNGs). Ambos podem produzir um fluxo de zeros e uns, que podem ser divididos em blocos ou subcorrentes de números aleatórios. Uma seqüência de bits aleatórios poderia ser interpretada como o resultado dos lançamentos de uma

moeda “justa” imparcial com os lados que são identificados como “0” e “1”, onde cada caso tem uma probabilidade de exatamente ½. Além disso, os lançamentos são independentes um do outro: o resultado de qualquer lançamento anterior não prejudica futuros lançamentos. A moeda imparcial “justa” é, portanto, o gerador de fluxo de bits aleatórios perfeito, uma vez que “0’s” e “1’s” serão distribuídos aleatoriamente e [0,1] distribuida uniformemente. Todos os elementos da sequência são gerados independentemente uns dos outros, e o valor do elemento seguinte na sequência não pode ser previsto, independentemente do número de elementos que já foram produzidos. Obviamente, a utilização de moedas imparciais para fins de criptografia, modelagem ou simulação é impraticável. No entanto, a saída hipotética de um gerador de uma seqüência realmente aleatória serve como um ponto de referência para a avaliação dos geradores de números aleatórios e pseudo-aleatórios.

Imprevisibilidade

Números aleatórios e pseudo-aleatórios gerados para quaisquer aplicações devem ser imprevisíveis. No caso de PRNGs, se a semente é desconhecida, o próximo número na sequência de saída deve ser imprevisível, apesar de qualquer conhecimento de números aleatórios anteriores na sequência. Esta propriedade é conhecida como imprevisibilidade para a frente. Também não deve ser viável para determinar a semente conhecendo os valores gerados (ou seja, a imprevisibilidade para trás também é obrigatória). Nenhuma correlação entre uma semente e qualquer valor gerado a partir da mesma devem ser evidentes; cada elemento da sequência deve parecer ser o resultado de um evento aleatório independente, cuja probabilidade é de 1/2. Para garantir a imprevisibilidade para a frente, o cuidado deve ser exercido na obtenção de sementes. Os valores produzidos por um PRNG são completamente previsíveis se a semente e algoritmo de geração são conhecidos. Uma vez que em muitos casos

2.2.2 Propriedades desejadas de um gerador ideal

Um gerador de números pseudo aleatórios deve possuir algumas propriedades que garantam sua qualidade, para um leigo a construção de um gerador de números aleatórios pode parecer uma tarefa simples e alguns programadores têem demonstrado ser relativamente fácil escrever programas que gerem estranhas sequências de números aparentemente imprevisíveis. Entretanto, é bastante complexo escrever um bom programa que gere sequências satisfatórias, ou seja, uma sequência virtualmete infinita de números aleatórios estatisticamente independentes entre 0 e 1. Pois estranhas e aparentemente imprevisíveis não são necessariamente aleatórios. Vamos parafrasear algumas afirmações feitas por dois dos diversos autores que trataram o tema.

- *D. H. Lehmer (1951)*: "Uma sequência aleatória é uma vaga noção baseada na ideia

de uma sequência onde cada termo é imprevisível e cujos dígitos passam em um certo número de testes, tradicionais com estatísticas ou dependendo do uso no qual a sequência será utilizada."

- *J. N. Franklin (1962)*: "A sequência (1) é aleatória se possuir todas as propriedades compartilhadas por todas as infinitas sequências de amostrass independentes de variáveis aleatórias sobre a distribuição uniforme.

A afirmação de Franklin essencialmente generaliza a de Lehmer ao dizer que a sequência precisa satisfazer *todos* os testes estatísticos. Esta definição não é completamente precisa e uma interpretação sensata leva-nos a concluir que uma sequência aleatória simplesmente não existe! Portanto, iniciemos com a primeira e menos restritiva afirmação de Lehmer e tentemos torná-la mais precisa. O que realmente queremos é uma pequena lista de propriedades matemáticas, cada uma delas satisfeita por nossas noções intuitivas de uma sequência aleatória; além disso, a lista precisa ser completa o bastante para que qualquer sequência que a satisfaça possa ser considerada aleatória

2.2.3 Histórico

2.2.4 Geradores disponíveis no R

- a
- b
- c

2.3 Verificação clássica das propriedades dos geradores

2.3.1 Repetibilidade, portabilidade, eficiência computacional

2.3.2 Testes

Vários testes estatísticos podem ser aplicados a uma sequência na tentativa de comparar e avaliá-la com uma sequência verdadeiramente aleatória. A aleatoriedade é uma propriedade probabilística; isto é, as propriedades de uma sequência aleatória podem ser caracterizadas e descritas em termos de probabilidades. O resultado provável de testes estatísticos, quando aplicados a uma sequência verdadeiramente aleatória, é conhecido a priori e pode ser descrito em termos de probabilidades. Existe um número infinito de possíveis testes estatísticos, cada avaliação da presença ou ausência de um "padrão", que, se for detectado, indica que a sequência não é aleatória. Por haver tantos testes para julgar se uma seqüência é aleatória ou não, nenhum conjunto finito específico de testes é considerado "completo". Além disso, os

resultados dos testes estatísticos devem ser interpretados com algum cuidado e cautela para evitar conclusões incorretas sobre um gerador específico. Um teste estatístico é formulado para testar a hipótese nula específica (H0). Para efeitos do presente documento, a hipótese nula sob teste é de que a sequência a ser testada é aleatória. Associada a esta hipótese nula está a hipótese alternativa (Ha), a qual, para este documento, é que a sequência não é aleatória. Para cada ensaio foi aplicada uma decisão ou conclusão é derivado que aceita ou rejeita a hipótese nula, isto é, se o gerador estiver (ou não) produzir valores aleatórios, com base na sequência que foi produzido. Para cada teste, uma estatística aleatoriedade relevante deve ser escolhido e utilizado para determinar a aceitação ou rejeição da hipótese de nulidade. De acordo com uma hipótese de aleatoriedade, tal estatística tem uma distribuição de valores possíveis. A distribuição referencial teórico desta estatística sob a hipótese nula é determinado por métodos matemáticos. A partir desta distribuição de referência, um valor crítico é determinado (normalmente, este valor está "longe" nas caudas da distribuição, ou seja, para fora no ponto 99%). Durante um teste, um valor estatístico do ensaio é calculado sobre os dados (a seqüência que está sendo testada). Este valor estatístico do ensaio é comparado com o valor crítico. Se o valor da estatística de teste excede o valor crítico, a hipótese nula de aleatoriedade é rejeitada. Caso contrário, a hipótese nula (a hipótese de aleatoriedade) não é rejeitada (ou seja, a hipótese é aceito). Na prática, a razão pela qual as hipóteses estatísticas funcionam é que a distribuição de referência e o valor crítico são dependentes e gerados, pressupondo-se a ocorrência de aleatoriedade. Se a suposição de aleatoriedade é, de fato, verdadeira para os dados que se têem, em seguida, o valor calculado resultante estatística sobre os dados de teste terão uma probabilidade muito baixa (por exemplo, de 0,01 %) de exceder o valor crítico. Por outro lado, se o valor calculado teste estatístico não exceda o valor crítico (isto é, se o evento de baixa probabilidade de ocorrer de fato), em seguida, a partir de um ponto de testes de hipóteses de vista estatístico, o evento baixa probabilidade deve ocorrer não naturalmente. Portanto, quando o valor de teste estatística calculada excede o valor crítico, a conclusão é feita de que a suposição original da aleatoriedade é suspeito ou com defeito. Neste caso, o teste de hipótese estatística leva as seguintes conclusões: rejeitar H0 (aleatoriedade) e aceito Ha (não-aleatoriedade). Teste de hipóteses de Estatística é um procedimento conclusão geração que tem dois resultados possíveis, aceite H0 (os dados são aleatórios) ou aceitar Ha (os dados não são aleatórios). A tabela 2.1 relaciona o verdadeiro estado (desconhecido) dos dados em questão para a conclusão a que chegou usando o procedimento de teste.

Tabela 2.1: Teste de Hipóteses

Real Situação	~~**Aceite**~~ H_0	**Aceite** H_a **(Rejeite** H_0**)**
Dados são aleatórios H_0 verdadeiro	Sem erro	Erro tipo I
Dados não são aleatórios H_A verdadeiro	Erro tipo II	Sem erro

2.3.3 Testes anterior

2.3.4 Testes Diehard

2.3.5 Testes NIST

2.3.6 Testes Dieharder

Como substituto aos testes anteriores RGB reescreveu-os numa linguagem portável como C e acrescentou ao conjunto de testes disponíveis no Diehard alguns outros testes do NIST e mais alguns de sua autoria, listados na tabela 2.2 e explicados abaixo.

- **"Birthdays" test (modificado). Id(0)** - Cada teste determina o número de intervalos que combinam de 512 "aniversários" (por padrão) tomados num "ano" fictício de 24 bits (por padrão). Este processo é repetido (por padrão) 100 vezes e o resultado é acumuldo em um histograma. Intervalos repetidos podem ser distribuidos em uma distribuição Poisson se o gerador em questão for aleatório o suficiente, e em uma Chi Quadrado com o p-valor avaliado relativamente à hipótese nula. É recomendado rodar este teste próximo ou com exatamente 100 amostras por p-valor com `-t 100`. Dois parametros adicionais foram incluidos. No Diehard, nms=512, porém isto pode ser variado e todas as fórmulas de Marsaglia continuam a funcionar. Pode ser ajustado para valores diferentes com `-x nmsvalue`. Similarmente, o parâmetro `nbits`pode ser 24, mas podemos fazê-lo assumir qualquer valor desde que seja menor ou igual a `rmax_bits = 32`. E pode ser atribuido qualquer valor com o parâmetro `-y nbits`. Ambos são padrão para os valores do Diehard se as opções `-x e -y` não forem utilizadas.

2.4 Verificação das propriedades com ferramentas da teoria da informação

2.4.1 aaa

Neste capítulo tratamos da Revisão Bibliográfica realizada para o desenvolvimento do trabalho, no capítulo seguinte tratamos da metodologia utilizada do desenvolvimento do mesmo.

Tabela 2.2: Testes disponíveis no Dieharder

Seq	Nome	ID
1	diehard birthdays	0
2	diehard operm5	1
3	diehard rank 32x32	2
4	diehardrank 6x8	3
5	diehard bitstream	4
6	diehard opso	5
7	diehard oqso	6
8	diehard dna	7
9	diehard count 1s stream	8
10	diehard count 1s byte	9
11	diehard parking lot	10
12	diehard 2dsphere	11
13	diehard 3dsphere	12
14	diehard squeeze	13
15	diehard sums	14
16	diehard runs	15
17	diehard craps	16
18	marsaglia tsang gcd	17
19	sts monobit	100
20	sts runs	101
21	sts serial	102
22	rgb bitdist	200
23	rgb minimum distance	201
24	rgb permutations	202
25	rgb lagged sum	203
26	rgb kstest test	204
27	dab bytedistrib	205
28	dab dct	206
29	dab filltree	207
30	dab filltree2	208
31	dab monobit2	209

3

Metodologia

ESTE capítulo tem como objetivo apresentar os materiais e métodos utilizados no trabalho. Como principal objetivo, este capítulo visa fornecer subsídios suficientes e para que o mesmo possa ser reproduzido e a continuidade do mesmo na pesquisa e desenvolvimento dos problemas deixados em aberto possa ser alcançada.

3.1 Materiais e Métodos

Em relação a fundamentação teórica, utilizou-se como principal fonte de pesquisa a área de indexação de periódicos científicos ISI *Web of Knowledge*, onde foram obtidas a grande maioria das referências, usando como parâmetros o fator de impacto dos periódicos pesquisados, a quantidade de citações de cada publicação, o grau de relevância para o tema pesquisado e o nível de produtividade (fator-H) dos autores envolvidos. O apoio em livros, surveys, lecture notes e ferramentas complementares de busca, como o *google acadêmico* foram utilizadas para complementar esta pesquisa.

Para organizar, catalogar e facilitar a consulta a todo material obtido, as referências foram gerenciadas com a ferramenta *Mendeley*. Um gerenciador de referências bibliográficas multiplataforma gratuito que permite organizar de maneira centralizada vários vínculos entre as referências utilizadas, bem como visualizar e anotar as mesmas dentro da própria ferramenta. Quanto à editoração eletrônica do trabalho, fez-se uso da plataforma LaTeX, com editor de textos *Kile* de código aberto. Este trabalho foi desenvolvido num equipamento com as seguintes configurações:

Arquitetura	`Intel i7 64 bits`
S.O.	Linux Mint 17.1 - kernel 3.13.0 – 43 – $generic$
Editor	Kile versão 2.1.3 e LaTeXtexlive 2013.20140215 – 1

Do ponto de vista técnico deste trabalho, com ênfase em Geradores de Números Pseudo Aleatórios, é utilizada a plataforma de análise estatística R . Esta plataforma foi desenvolvida originalmente por Ross Ihaka e Robert Gentleman, com o intuito de ser uma linguagem de código aberto voltada para a análise estatística e, consequentemente, a precisão numérica com fortes características funcionais (**?**). Por este motivo, ela será utilizada para a geração e manipulação das sequências pseudo aleatórias, análise dos dados e geração dos gráficos desse trabalho. A precisão numérica desta ferramenta, sendo aferida por **?**, é adequada para essa abordagem.

As ferramentas utilizadas no desenvolvimento deste trabalho, são preferencialmente multiplataforma e código aberto com licença de uso *GNU General Public License* (GPL).

Todos esses aplicativos, métodos e informações obtidas, forneceram grandes contribuições para o traçado da linha mestra deste trabalho, indicando que o mesmo está na fronteira do conhecimento produzindo um estado da arte fidedigno aos temas e ferramentas adotadas para norteá-lo.

Neste capítulo tratamos da metodologia utilizada do desenvolvimento do trabalho, no capítulo seguinte analisaremos os impactos esperados com a realização do mesmo.

4

Resultados Esperados

UMA vez com a modelagem concluída e os dados para simulação, serão realizados os experimentos afim de comprovar a eficácia do modelo proposto.

4.1 Resultados Esperados

Neste capítulo tratamos os resultados esperados com a realização do trabalho enquanto que no próximo faremos a conclusão alcançada com o mesmo.

5

Conclusão

A análise dos dados deve evidenciar a importância de trabalhar a redução dos dados numa rede de sensores, visando a redução do consumo de energia e maximizando o tempo de vida da rede.

5.1 Impactos Esperados

A análise dos dados deve evidenciar a importância de trabalhar a redução dos dados numa rede de sensores, visando a redução do consumo de energia e maximizando o tempo de vida da rede.

Apêndice A

Ambiente Reprodutível e Computacional

O?

Este trabalho foi redigido em LaTeX utilizando uma modificação do estilo IC-UFAL. As referências bibliográficas foram preparadas no Mendeley e administradas pelo BibTeX com o estilo LaCCAN. O texto utiliza fonte Fourier-GUT*enberg* e os elementos matemáticos a família tipográfica Euler Virtual Math, ambas em corpo de 12 pontos.