Geração de uma Distribuição Discreta Usando Moedas Desbalanceadas

Danielle P. B. de A. Camara, Valdemar C. da Rocha Jr. e Cecilio Pimentel

***Resumo*— A geração eficiente de uma distribuição de probabilidade discreta é de interesse atual nas áreas de criptografia e de geração de números aleatórios, para testes e simulação de sistemas de comunicações. Neste trabalho é apresentado um algoritmo para gerar uma distribuição discreta através do lançamento de duas ou mais moedas, sendo algumas delas desbalanceadas. Em particular, esta abordagem contribui com uma solução alternativa do problema clássico da geração de uma distribuição discreta uniforme usando duas ou mais moedas desbalanceadas.**

***Palavras-Chave*— Geração de números aleatórios, criptografia, teoria da informação.**

***Abstract*— The efficient generation of a discrete probability distribution is of current interest in the areas of cryptography and random number generation. This paper presents an algorithm for generating a discrete distribution using two or more coins, being one of them unbiased. In particular, this approach contributes to an alternative solution to the classical problem of generating a discrete uniform probability distribution using two or more unbiased coins.**

***Keywords*— Random number generation, cryptography, information theory**

I. INTRODUÇÃO

O problema da geração de uma distribuição de probabilidade discreta usando lançamentos de uma moeda desbalanceada é antigo e de grande importância nas áreas de criptografia e de geração de números aleatórios, usados para testes e simulação de sistemas de comunicações, assim como em muitas outras aplicações computacionais. Há mais de quarenta anos von Neumann [1] introduziu um algoritmo simples para gerar uma seqüência de bits estatisticamente independentes e equiprováveis a partir do lançamento de uma moeda com viés desconhecido. A partir daí, muitos pesquisadores têm considerado o problema e estudado a geração de variáveis aleatórias uniformes sob diferentes pontos de vista [2]-[6],[8]-[13].

Basicamente, dois aspectos são levados em conta neste tipo de problema: a geração considerando um tempo limite curto (short bounded time) ou mais tradicionalmente considerando um tempo esperado curto, que será o nosso caso. Feldman et al. [7] provaram em seu trabalho, entre outros resultados, que os resultados do lançamento de um dado honesto de *n* faces, i.e., os valores de uma variável aleatória que assume *n* valores equiprováveis, podem ser simulados em um intervalo de tempo limitado usando lançamentos de apenas um tipo de moeda, com distribuição racional apropriada de cara e coroa, se e só se, *n* é uma potência de 2 ([7], teorema 2), e que um dado honesto de *n* faces sempre pode ser simulado usando duas moedas, com distribuição de cara e coroa apropriadas, usando no máximo $\lceil 2\log n\rceil+1$ lançamentos, onde $\lceil x\rceil$ denota o menor número inteiro maior ou igual a x.

Um dos resultados obtidos por Gargano e Vaccaro [14] foi a melhoria de tal cota. Através do algoritmo proposto em [14], a geração de um dado honesto de *n* faces usando duas moedas, sendo uma delas honesta e a outra com viés, é feita com um número máximo de lançamentos igual a

$$1+\lfloor\log n\rfloor+\lceil\log(n-2^{\lfloor\log n\rfloor})\rceil+pw(n) \qquad (1)$$

e um número médio de lançamentos igual a

$$1+\lfloor\log n\rfloor+(2^{pw(n)}/n)(\lceil\log r(n)\rceil 2^{\lceil\log r(n)\rceil} -r(n)(\lfloor\log r(n)\rfloor-pw(n))), \qquad (2)$$

onde $pw(n)$ =max$\{i : 2^i$ divides $n\}$ (Teorema 1, [14]).

Na *Seção II* introduzimos um novo algoritmo para a geração de uma distribuição de probabilidade discreta através do lançamento de duas ou mais moedas, algumas delas com viés (ou desbalanceadas). Convém ressaltar que o algoritmo proposto por Gargano e Vaccaro é específico para uma dada escolha de moedas. Na *Seção III* a aplicação deste algoritmo será ilustrada através de exemplos, nos quais gera-se uma distribuição de probabilidade uniforme usando duas moedas, uma honesta e outra desbalanceada. Finalizando, na *Seção IV* apresentaremos algumas conclusões sobre este trabalho, assim como sugestões para pesquisas futuras.

II. NOVO ALGORITMO

O algoritmo introduzido em [15] trata de um esquema de substituição homofônica no qual cada palavra binária de homofonema tem como símbolos variáveis aleatórias independentes e identicamente distribuídas, obedecendo a uma distribuição de probabilidade arbritária. Em outras palavras, trata da geração de uma distribuição de probabilidade discreta através do uso de apenas uma moeda com viés.

Nossa proposta nesta seção é a generalização do algoritmo MIN-ENT por passo apresentado em [15], a fim de gerar uma distribuição discreta, utilizando duas ou mais moedas com viés, obtendo resultados equivalentes aos obtidos por Gargano e Vaccaro [14], com a diferença de não necessariamente usar a distribuição de cara e coroa dependente de *n*, como no algoritmo sugerido em [14].

Danielle P. B. de A. Camara, Valdemar C. da Rocha Jr e Cecílio Pimentel, Grupo de Pesquisa em Comunicações - CODEC, Departmento de Eletrônica e Sistemas, Caixa Postal 7800,Universidade Federal de Pernambuco, CEP:50711-970, Recife PE, BRASIL. E-mails: dpbac@ufpe.br, vcr@ufpe.br e cecilio@ufpe.br. Os autores agradecem ao CNPq pelo apoio parcial recebido através dos projetos 141215/2002-0, 304214/77-9 e 300987/96-0, respectivamente.

A. Descrição do algoritmo

O algoritmo aqui proposto segue essencialmente os mesmos passos daquele apresentado em [15] com a importante distinção de que, ao invés de usar apenas uma moeda, são usadas duas ou mais moedas e a cada passo observa-se qual moeda deve ser escolhida para lançamento, levando em conta a minimização da entropia naquele passo.

Sejam $m_1 = \{p_1, 1-p_1\}$, $m_2 = \{p_2, 1-p_2\}$, ..., $m_r = \{p_r, 1-p_r\}$ as distribuições de probabilidade das moedas e $P_U = P_U(u_1), \ldots, P_U(u_K)$ a distribuição de probabilidade a ser gerada.

Segue-se a descrição do algoritmo proposto.

1) Lançar cada uma das moedas, associando a cada uma delas uma árvore.
2) Para cada uma das árvores, verificar se a maior probabilidade dos ramos é menor ou igual ao maior valor de $\Gamma_0 = \{P_U(u_1), P_U(u_2), \ldots, P_U(u_K)\}$.
   (Obs: Depois da primeira iteração será considerada uma expressão Γ, em substituição a Γ_0, a qual é descrita em detalhes no apêndice.)
   a) Se alguma(s) árvore(s) verificar(em) esta condição, manter a(s) mesma(s) e eliminar as outras. Continuar o algoritmo aplicando às árvores que foram mantidas o procedimento indicado no Apêndice A.
   b) Caso contrário, ir para o passo 3.
3) Fazer a expansão de cada árvore, considerando todas as moedas. Ir para o passo 2.

III. Exemplos ilustrativos

A título comparativo, iremos aplicar o algoritmo por nós sugerido usando as moedas que seriam usadas no algoritmo de Gargano e Vaccaro para a geração de uma distribuição de probabilidade uniforme usando duas moedas, uma honesta e outra com distribuição dada por

$$\left(2^{\lceil \log r(m) \rceil}/m, 1 - 2^{\lceil \log r(m) \rceil}/m\right), \tag{3}$$

onde $n = 2^t m$ e $r(m) = m - 2^{\lfloor \log m \rfloor}$. Observando que o valor de n usado em [14] refere-se ao produto de uma potência de 2 por um número ímpar m.

Chamamos atenção mais uma vez para o fato que o algoritmo aqui proposto funciona para qualquer escolha das moedas, enquanto o algoritmo proposto por Gargano e Vaccaro é específico para uma dada escolha de moedas.

Exemplo 1: Consideremos $n = 6$, logo as moedas a serem usadas serão $m_1 = (1/2, 1/2)$ e $m_2 = (2/3, 1/3)$.

Pelo primeiro passo do algoritmo, observamos que $1/2 > 1/6$ (Árvore A1, mostrada na Fig. 1.) e $2/3 > 1/6$ (Árvore A2 e Árvore A3 mostradas, respectivamente, nas Fig.2. e Fig.3.), desta forma é necessária a expansão.

Feita a primeira expansão observa-se nas árvores A1 e A2 que as maiores probabilidades obtidas ainda são maiores que $1/6$, sendo assim necessária mais uma expansão desses elementos. Observa-se também que se expandido o nó da árvore A3 cuja probabilidade é 4/9, seja usando a moeda honesta ou a desbalanceada, o resultado obtido ainda será

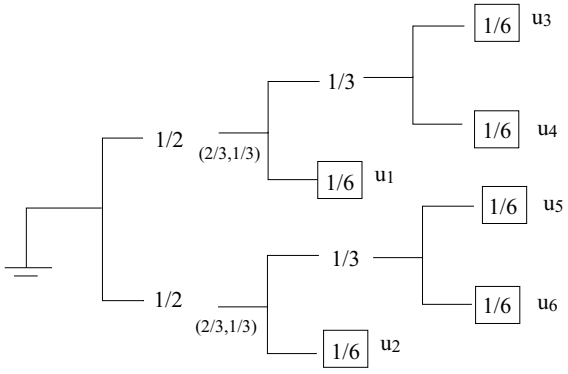

Fig. 1. Árvore A1 referente a *n*=6.

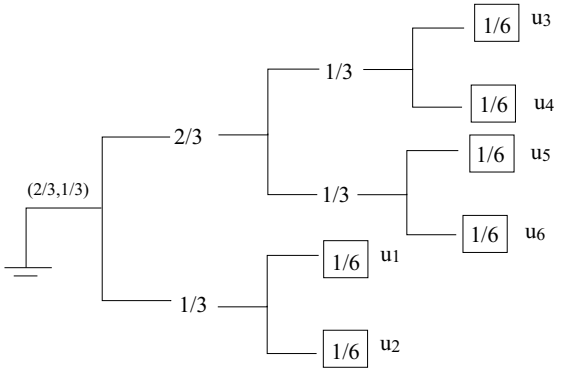

Fig. 2. Árvore A2 referente a *n*=6.

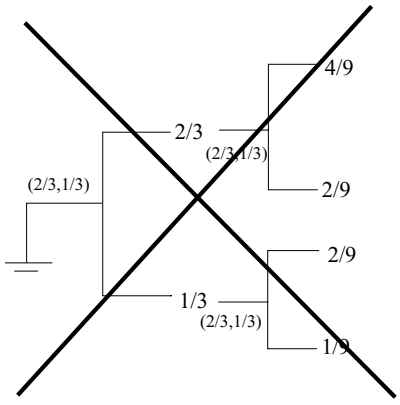

Fig. 3. Árvore A3 referente a *n*=6, eliminada por não atender o critério de diferença mínima.

maior que 1/6, diferentemente do que ocorrerá nas outras duas árvores. Desta forma, a árvore A3 é eliminada.

Segue-se com o algoritmo, resultando assim duas possíveis árvores, ambas com mesmo comprimento médio $E[T]$ e comprimento máximo L_{max}, dados por:

$$E[T] = \sum_x p(x) l_T(x), \tag{4}$$

$$L_{max} = \max_x l_T(x) \tag{5}$$

onde $p(x)$ é a probabilidade da seqüência de cara e coroa associada ao único caminho da fonte ao ramo x da árvore e $l_T(x)$, é o comprimento deste caminho.

Assim, temos como resultados para este exemplo:

$$E[T] = 2,67.$$

$$L_{max} = 3.$$

Pelo algoritmo sugerido por Gargano e Vaccaro (seção II, [14]) é obtida a mesma árvore ilustrada na Figura 1, assim ambos os algoritmos apresentam mesmo comprimento médio e comprimento máximo.

Exemplo 2: Consideremos $n = 7$. Neste caso o nosso algoritmo produz uma árvore de comprimento infinito, porém o valor $E[T]$ obedece a expressão (2).

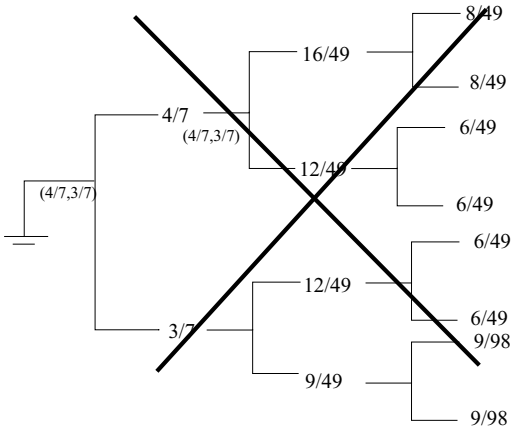

Fig. 4. Árvore A4 referente a *n*=7, eliminada por não atender o critério de diferença mínima.

Pelos mesmos motivos apontados no exemplo anterior escolhemos a árvore A5 (Fig. 5) e eliminamos a árvore A4 (Fig. 4).

Observa-se que o comprimento médio da árvore da Figura 5 é dado por

$$\begin{aligned} E[T] &= 1+\frac{4}{7}+\frac{3}{7}+\frac{2}{7}+\frac{2}{7}+\frac{3}{14}+\frac{3}{14}+\frac{3}{28}+\frac{3}{56} \\ &\quad +\frac{3}{56}+\frac{3}{112}+\frac{3}{224}+\frac{3}{224}+\frac{3}{448}+... \\ &= 3+\frac{6}{7}\sum_{i=1}^{\infty}(\frac{1}{4})^i \\ &= 3+0,29 = 3,29 \end{aligned}$$

o mesmo comprimento médio obtido pela árvore da Figura 7, construída usando o algortimo de Gargano e Vaccaro.

IV. Conclusões

Introduzimos neste trabalho um algoritmo para gerar uma distribuição discreta através do lançamento de duas ou mais moedas, sendo algumas delas desbalanceadas. Tal algoritmo mostrou ter resultados equivalentes aos obtidos em [14], para a geração de uma distribuição de probabilidade discreta uniforme usando duas moedas. Em contraposição ao algoritmo de Gargano e Vaccaro, nosso algoritmo tem a vantagem de poder operar com quaisquer que sejam as moedas disponíveis, independentes de n. Notou-se que o desempenho do algoritmo, medido em termos de tempo esperado curto, varia dependendo

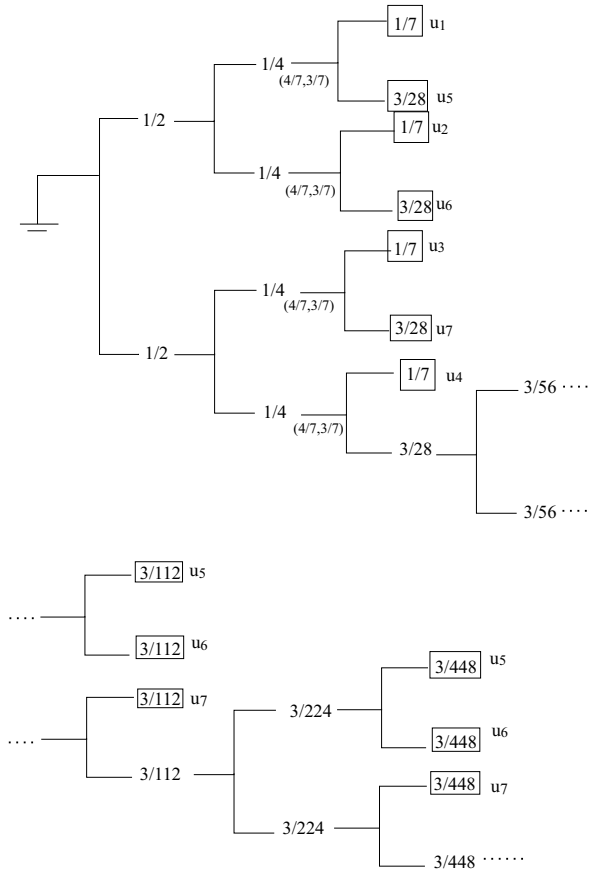

Fig. 5. Árvore A5 referente a *n*=7.

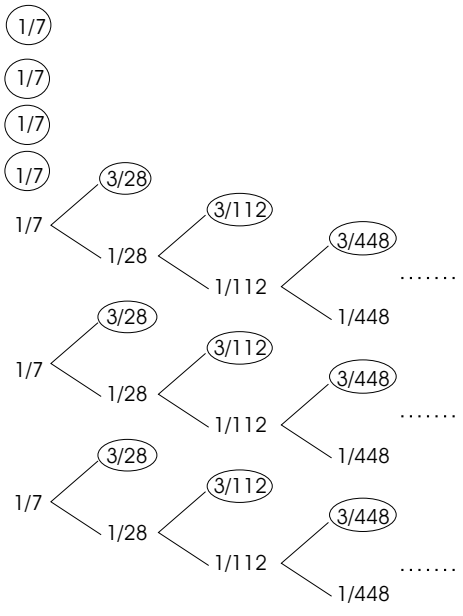

Fig. 6. Geração de fonte uniforme com $n = 7$ usando o algoritmo MIN-ENT por passo.

do grupo de moedas usadas. Esta observação sugere a existência de um grupo de moedas desbalanceadas que otimize o mesmo. Um possível critério de escolha de moedas a fim de otimizar o algoritmo ainda encontra-se sob investigação. Sugerimos para trabalhos futuros, além da definição do grupo

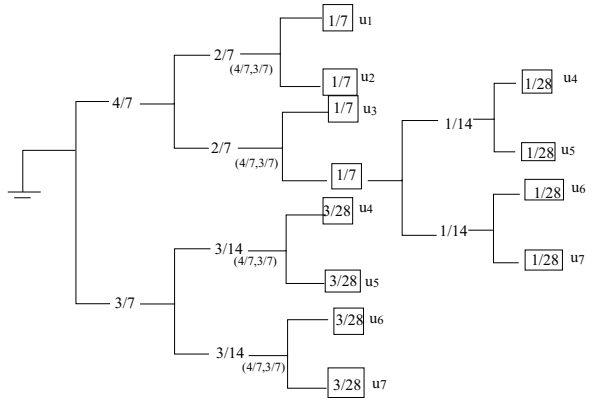

Fig. 7. Árvore referente ao *n*=7 usando o algoritmo proposto por Gargano e Vaccaro.

de moedas que otimiza o algoritmo, caso o mesmo exista, uma investigação da possibilidade do algoritmo MIN-ENT por passo ser implementado com árvores finitas quando n for primo, mesmo sabendo que $E[T]$ satisfaz a expressão (2).

APÊNDICE

A. Algoritmo MIN-ENT por passo

Seja $\Pi_D = \{\pi_0, \pi_1, \ldots, \pi_{D-1}\}$ a distribuição de probabilidade dos dígitos das palavras de homofonema. Para uma dada fonte o algoritmo MIN-ENT por passo simultaneamente encontra a decomposição da probabilidade de cada símbolo da fonte, como uma soma finita ou infinita de termos $\pi_0^{\lambda_0}\pi_1^{\lambda_1}\ldots\pi_{D-1}^{\lambda_{D-1}}$, e a correspondente palavra livre de prefixo, na qual o dígito $i,\ 0 \leq i \leq D-1$ ocorre λ_i vezes. Os homofonemas são selecionados como nós terminais na árvore D-ária enraízada T com probabilidades, de tal modo que de cada nó emanam D ramos com probabilidades $\pi_0, \pi_1, \ldots, \pi_{D-1}$, respectivamente. Denotemos por $v(i,j)$ o j-ésimo homofonema alocado ao símbolo da fonte $u_i,\ 1 \leq i \leq K,\ j = 1, 2, \ldots$, e denotemos por $\alpha(i,j)$ a probabilidade de $v(i,j)$.

Definição 1: Definimos a soma corrente de símbolo $\gamma_m(i)$ para $U = u_i$ na m-ésima iteração do algoritmo MIN-ENT como

$$\gamma_m(i) = P_U(u_i) - \sum_{k=1}^{j-1} \alpha(i,k),$$

com $\gamma_m(i) = P_U(u_i)$ para $j = 0$, na qual j denota o número de homofonemas alocados a u_i até a m-ésima iteração.

Definição 2: Definimos o conjunto de soma corrente Γ_m na m-ésima iteração do algoritmo MIN-ENT por passo como

$$\Gamma_m = \{\gamma_m(i) | \gamma_m(i) > 0, 1 \leq i \leq K\},$$

com $\Gamma_0 = \{P_U(u_1), P_U(u_2), \ldots, P_U(u_K)\}$.

Seja $\gamma_{\max} = \max\ \gamma_m(i) \in \Gamma_m,\ 1 \leq i \leq K$. Quando $m = 0$ no algoritmo MIN-ENT por passo nós construímos T a partir da raiz, começando com apenas D folhas. Daí então nós expandiremos cada nó terminal em T, cuja probabilidade exceda $\gamma_{\max}$, em um número mínimo de ramos suficiente para fazer com que as probabilidades dos nós terminais estendidos resultantes sejam iguais ou menores que $\gamma_{\max}$. Chamaremos a árvore resultante de *árvore D-ária enraízada e processada com probabilidades*, T_p. Na m-ésima iteração, $m \geq 1$, um homofonema é alocado a um nó terminal da correspondente T_p, de modo que o nó terminal não utilizado que possua a maior probabilidade, denotada por P_M, seja alocado como um homofonema para o símbolo u_r que apresente o mínimo valor não negativo para a diferença entre sua soma corrente de símbolo $\gamma_m(r)$ e P_M, i.e., tal que $\min_i\{\gamma_m(i) - P_M | (\gamma_m(i) - P_M) \geq 0\} = \gamma_m(r) - P_M \geq 0,\ 1 \leq i \leq K$. O algoritmo consiste dos seguintes passos.

1) Faça $m = 0$. Faça $\gamma_0(i) = P_U(u_i),\ 1 \leq i \leq K$. Faça $\Gamma_0 = \{P_U(u_1), P_U(u_2), \ldots, P_U(u_K)\}$.
2) Determine $\gamma_{\max}$ e construa a árvore T_p para a m-ésima iteração expandindo cada nó terminal não usado, na árvore construída para a $(m-1)$-ésima iteração, $m \geq 1$, cuja probabilidade exceda $\gamma_{\max}$, em um número mínimo de ramos suficiente para fazer a probabilidade dos nós terminais resultantes menores ou iguais a $\gamma_{\max}$.
3) Encontre em T_p, o caminho não usado E_l cuja probabilidade $P(E_l)$ seja a maior dentre as dos caminhos não usados, i.e., $P(E_l) = P_M$. Denotemos por l comprimento de E_l.
4) Se, para $1 \leq i \leq K$, $\min_i\{\gamma_m(i) - P_m | (\gamma_m(i) - P_m) \geq 0\} = \gamma_m(r) - P_m \geq 0$, nós então associamos a u_r o homofonema (nó terminal) $v(r,j)$ e a palavra de homofonema D-ária de comprimento l, cujos símbolos constituem o rótulo de E_l em T_p. Isto implica $\alpha(r,j) = P_M$. Compute a soma corrente de símbolo $\gamma_m^{'}(r)$ após esta decomposição e faça $\Gamma_m^{'} = \Gamma_m - \{\gamma_m(r)\}$. Se $\gamma_m^{'}(r) = 0$ então faça $\Gamma_{m+1} = \Gamma_m^{'}$. A decomposição de $P_U(u_r)$ estará agora concluída e conterá j homofonemas, e se $\Gamma_{m+1} = \phi$ então FIM. Em caso contrário, i.e., se $\gamma_m^{'}(r) > 0$, então faça $\Gamma_{m+1} = \Gamma_m^{'} \cup \{\gamma_m^{'}(r)\}$.
5) Faça $m \leftarrow m + 1$.
6) Vá para o passo 2.

REFERÊNCIAS

[1] J.von Neumann, " Various techniques used in connection with random digits, notes by G. E. Forsythe, National Bureau of Standards ", Applied Math Ser., vol. 12, pp. 36-38; reprinted in von Neumann's Collected Works., vol. 5. Oxford, U.K.: Pergamon, 1963, pp. 768-770.

[2] J. Abrahams, " Generation of discrete distributions from biased coins ", *IEEE Trans. on Inform. Theory*, vol. 42, pp. 1541-1546, 1996.

[3] M. Blum, "Independent unbiased coin flips from a correlated biased source-A finite state Markov chain ", *Combinatorica*, vol. 6, no. 2, pp. 97-108, 1986.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[5] E. W. Dijkstra, "Making a fair roulette from a possibly biased coin ", *Inform. Processing Lett.*, vol. 36, p. 193, 1990.

[6] P.Elias, "The efficient computation of an unbiased random sequence ", *Ann. Math. Statist.*, vol. 43, pp. 865-870, 1972.

[7] D. Feldman,R. Impagliazzo, M. Naor, N. Nisan, S. Rudich, and A. Shamir, "On dice and coins: Models of computation for random generation ", *Inform. Comput.*, vol. 104, pp. 159-174, 1993.

[8] W. Hoeffding and G. Simons, "Unbiased coin tossing with a biased coin ", *Ann. Math. Statist.*, vol. 41, pp. 341-352, 1970.

[9] T. S. Han and M. Hoshi, "Interval algorithm for random number generation ", *IEEE Trans. on Inform. Theory*, vol. 43, pp. 599-611, March 1997.

[10] Y. Horibe, "Entropy and optimal random number transformation ", *IEEE Trans. on Inform. Theory*, vol. 27,pp.527-529, July 1981.

[11] D.E. Knuth and A. C. Yao, "The complexity of nonuniform random number generation", in Algorithms and Complexity, New Directions and Results, J. F. Traub, Ed. New York: Academic, 1976, pp.357-428.
[12] R. Motwani and P. Raghavan, *Randomized Algorithms. Cambridge*, U.K.: Cambridge Univ. Press, 1996.
[13] Q. F. Stout and B. Warren, "Tree algorithms for unbiased coin tossing with a biased coin ", *Ann. Probab.*, vol. 12, pp. 212-222, 1984.
[14] L. Gargano and Ugo Vaccaro, "Efficient generation of fair dice with few biased coins ", *IEEE Trans. on Inform. Theory*, vol. 45, pp. 1600-1606, July 1999.
[15] V. C. da Rocha Jr., C. Pimentel e M. M. Vasconcelos, "Substituição homofônica ótima com restrição", XX Simpósio Brasileiro de Telecomunicações, págs. 273 - 277, Rio de Janeiro, Brasil, 05-08 de outubro de 2003.