



Códigos LDPC e LDGM: Construção, Codificação e Decodificação

Bolsista: Paulo Alexandre
Orientador: Jaime Portugheis
Faculdade de Tecnologia

Resumo— O artigo considera duas classes de códigos de canal para transmissão digital: códigos LDPC e códigos LDGM. As características das técnicas de construção destes códigos são descritas de maneira sucinta e a complexidade dos métodos de codificação é comparada. Um software de simulação computacional para obtenção dos desempenhos de quatro algoritmos de decodificação iterativa para códigos LDPC foi desenvolvido. A comparação dos desempenhos destes algoritmos para um código LDPC específico utilizado num canal AWGN foi obtida. A comparação mostra o compromisso entre desempenho e complexidade de decodificação.

I. INTRODUÇÃO

Os códigos LDPC (do inglês, Low-Density Parity-Check Codes) foram propostos por Gallager em [1] decorrente do trabalho de sua tese de doutorado. Tanner [2] generalizou o trabalho de Gallager e introduziu uma representação gráfica para códigos LDPC, chamada atualmente de grafos de Tanner. Em meados dos anos 90, a pesquisa sobre estes códigos ressurgiu em diversos trabalhos [3, 4, 5, 6, 7, 8]. Novas classes de códigos foram definidas, critérios de projeto e métodos de codificação eficientes foram propostos e novos algoritmos de decodificação foram desenvolvidos. Principalmente, mostrou-se que o uso de códigos LDPC (ou de suas generalizações) e decodificação iterativa pode atingir valores extremamente próximos do limite de capacidade previsto por Shannon para um sistema de comunicação utilizando transmissão digital. Códigos LDGM (do inglês, Low-Density Generator Matrices) foram introduzidos em [9] com o intuito de obter-se uma classe de códigos que também poderia atingir valores próximos do limite de capacidade de Shannon e ao mesmo tempo possuísse um método eficiente de codificação. O artigo está assim organizado. Na seção II, as características das técnicas de construção destes códigos são descritas de maneira sucinta e a complexidade dos métodos de codificação é comparada na seção III. Na seção IV, quatro algoritmos de decodificação iterativa para códigos LDPC são descritos de maneira sucinta. E finalmente, a seção V mostra a comparação dos desempenhos destes algoritmos para um código LDPC específico utilizado num canal AWGN (do inglês, Additive White Gaussian Noise).

II. CONSTRUÇÃO DE CÓDIGOS LDPC E LDGM

No que segue, iremos descrever as construções de Gallager e de Mackay [10] para códigos LDPC e a construção de códigos LDGM.

A. Construção de Gallager

Os códigos LDPC originais desenvolvidos por Gallager são códigos LDPC regulares, ou seja, com o mesmo número de 1's por linhas e também por colunas da matriz de verificação de paridade. Para um exemplo, iremos analisar uma matriz pequena. Ela será dividida em três submatrizes. A primeira submatriz é da seguinte forma: a primeira linha iniciará com bit 1 até a terceira coluna, a segunda linha terá bit 1 da quarta até a sexta coluna, e assim sucessivamente. As demais submatrizes têm os bits 1's espelhados aleatoriamente, como demonstrado a seguir:

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

B. Construção de Mackay

MacKay desenvolveu um grande número de códigos LPDC para aplicar em comunicação e armazenamento de dados, a maioria dos quais é regular. Ele propôs algoritmos de construção de códigos. Os ítems a seguir descrevem em ordem crescente de complexidade de alguns destes algoritmos:

1. H é criada gerando colunas aleatórias, tendo cada uma mesmo peso de 1's, e o mais próximo possível das linhas terem o número de 1's uniforme, ou seja, não necessariamente terão o mesmo peso de 1's.

2. H é criada gerando aleatoriamente colunas de peso w_c , assegurando que o peso w_r seja fixo.

3. H é gerada como em 2, além de ciclos curtos serem evitados.



4. H é gerada como em 3, fazendo $H = [H_1 \mid H_2]$ e restringido para que H_2 seja invertível.

C. Construção de códigos LDGM

Códigos LDGM sistemáticos são códigos lineares com matriz geradora, sendo $G = [I \ P]$, em que P é uma matriz esparsa que pode ser obtida de maneira aleatória. Observe que a matriz H , correspondente a G , também é esparsa e, portanto, os códigos LDGM podem ser considerados como uma classe específica de códigos LDPC, que podem ser decodificados, portanto, através de um algoritmo de decodificação para códigos LDPC padrão.

III. COMPLEXIDADE DE CODIFICAÇÃO

A construção de Gallager impôs restrições nas formas das submatrizes que compõem a matriz H . Esta abordagem possui a vantagem de tornar o processo de codificação mais simples já que os bits de paridade podem ser obtidos como função dos bits de informação. Entretanto, a abordagem não leva aos melhores códigos possíveis de serem construídos. Já as construções 1. e 2. de MacKay, descritas acima, introduzem a geração aleatória do número de 1's nas colunas da matriz H . Códigos melhores podem ser construídos, mas o método de codificação será mais complexo.

Impondo algumas restrições nas submatrizes da matriz H (como a invertibilidade da construção 4. descrita acima), Richardson e Urbanke [11] propuseram um algoritmo de codificação com complexidade linear. A principal vantagem dos códigos LDGM sobre os códigos LDPC reside na simplicidade dos algoritmos de codificação dos primeiros se a matriz geradora for escolhida na sua forma sistemática.

IV. ALGORITMOS DE DECODIFICAÇÃO

Gallager introduziu dois algoritmos de baixa complexidade para decodificar códigos LDPC. O primeiro utiliza decisões abruptas no receptor (trabalham apenas com os bits na saída do demodulador) enquanto que o segundo é um algoritmo de decisão suave (os valores reais na saída do demodulador também são utilizados). Ambos estão relacionados com algoritmos de propagação de probabilidades em grafos. Outras denominações comuns são: algoritmos de passagem de mensagens e algoritmos soma-produto [12]. Os algoritmos de Gallager foram melhorados mais recentemente [13]. No que segue, descreveremos sucintamente um dos algoritmos de Gallager, duas das suas modificações e o algoritmo soma-produto.

A. Algoritmo Bit-Flipping

O algoritmo Bit-Flipping utiliza o mesmo procedimento de um algoritmo de passagem de mensagens. Entretanto ele

trabalha com decisão abrupta. A mensagem do nó j de função para o nó i de variável é dada por:

$$E_{j,i} = \sum_{i' \in B_{j,i'} \neq i} M_{i'} \bmod 2$$

sendo M o vetor de decisão abrupta e B é o conjunto de nós descrito em [12 pp.270].

B. Algoritmo Weighted Bit-Flipping

É possível introduzir valores de soft-decision no algoritmo descrito na seção anterior através da modificação do critério de inversão dos bits do vetor recebido [13, pp.52]. Assim para cada bit é considerado o valor real:

$$F_n = \sum_{m \in A_n} (2 \sum_{i \in B_m} M_i - 1) w_m$$

$$w_m = \left(\min_{i \in B_{j,i}} |y_i| \right)$$

onde B_m é definido em [12 pp.270] e y_i é a i -ésima saída real do demodulador. Agora o algoritmo inverterá o bit que tiver o maior valor real de F_n .

C. Algoritmo Modified Weighted Bit-Flipping

Neste algoritmo temos um valor real para inversão do bit dado por um valor real da seguinte função:

$$F_n = \sum_{m \in A_n} (2 \sum_{i \in B_m} M_i - 1) w_m - \alpha(y_i)$$

$$w_m = \left(\min_{i \in B_{j,i}} |y_i| \right)$$

sendo $\alpha > 0$ e simulado com o valor $\alpha = 0.1$, sugerida em [13, pp. 53].

D. Algoritmo Soma-Produto

O decodificador soma-produto [12, 280-288], [14, pp. 31-39] trabalha iterativamente para encontrar um valor mais próximo da máxima probabilidade a posteriori de cada bit. Isto funciona através de informações extrínsecas que cada bit recebe a cada iteração, reforçando o seu valor original. Essa informação extrínseca é trabalhada da seguinte maneira:

$$P_{j,i}^{ext} = \frac{1}{2} - \frac{1}{2} \prod_{i' \in B_{j,i'} \neq i} (1 - P_{i'}^{int})$$

onde $P_{i'}^{int}$ é a estimativa atual disponível no nó de verificação.

A mensagem do nó j de função para o nó de variável i será agora dada por:



$$E_{j,i} = LLR(P_{j,i}^{ext}) = \log\left(\frac{1 - P_{j,i}^{ext}}{P_{j,i}^{ext}}\right)$$

$$E_{j,i} = \log\left(\frac{\frac{1}{2} + \frac{1}{2} \prod_{i' \in B_{j,i'} \neq i} (1 - P_{i'}^{int})}{\frac{1}{2} - \frac{1}{2} \prod_{i' \in B_{j,i'} \neq i} (1 - P_{i'}^{int})}\right)$$

e se usarmos a seguinte relação,

$$\tanh\left(\frac{1}{2} \log\left(\frac{1-p}{p}\right)\right) = 1 - 2p$$

temos,

$$E_{j,i} = \log\left(\frac{1 + \prod_{i' \in B_{j,i'} \neq i} (\tanh(M_{j,i'}/2))}{1 - \prod_{i' \in B_{j,i'} \neq i} (\tanh(M_{j,i'}/2))}\right)$$

Como estamos considerando um canal AWGN com entrada binária, então, a saída do demodulador será trabalhada da seguinte maneira:

$$L_i = \frac{4y_i}{N_0}$$

onde a variância do ruído é $N_0/2$ e $M_{j,i} = L_i$.

V. RESULTADOS E DISCUSSÃO

Foram obtidos resultados de simulação do desempenho de um código com parâmetros $(N,K) = (240, 180)$ e uma matriz H com peso de linha igual a 4 e de coluna igual a 3. A Figura 1 compara o desempenho de um sistema sem código denominado de "UNCODED". O desempenho é medido pela taxa de erro de bit, BER (do inglês, Bit Error Rate). A razão sinal-ruído SNR (do inglês, Signal to Noise Ratio), é a razão da energia de bit, E_b , pela densidade de potência, N_0 .

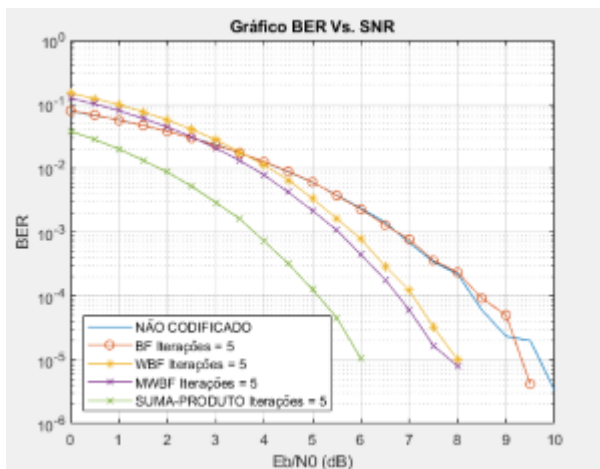


Fig. 1. Comparação das curvas BER Vs. SNR dos algoritmos de decodificação

Nesta Figura, BF denota o algoritmo Bit-Flipping, WBF o Weighted-Bit-Flipping e MWBF o Modified Weighted Bit-Flipping. Já SUM-PRODUCT denota o algoritmo soma-produto. Todos os resultados foram obtidos com 5 iterações. No algoritmo BF [12, pp.279], percebemos que só se torna viável a sua utilização com SNR acima de aproximadamente 9,3 dB, pois o sistema UNCODED começa a ter um BER superior.

O algoritmo WBF começa a ser bastante interessante com uma SNR maior que 3,5 dB. Já o algoritmo MWBF se torna interessante a partir de 3,0 dB.

O algoritmo soma-produto possui desempenho superior aos demais algoritmos para qualquer valor de SNR como era de se esperar..

REFERÊNCIAS

- [1] R.G. Gallager, "Low-density parity check-codes," IRE Trans. on Information Theory, pp. 21-28, Jan. 1962.
- [2] R.M.Tanner, "A recursive approach to low complexity codes," IEEE Trans. on Information Theory, vol. IT-27, pp. 533-547, Sept. 1981.
- [3] D.J.C.Mackay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. on Information Theory, vol. 45, n.2, pp. 399-431, March 1999.
- [4] T.J.Richardson and R.L.Urbanke, "Efficient encoding of low-density parity-check codes," IEEE Trans. on Information Theory, vol. 47, n.2, pp. 638-656, Feb. 2001.
- [5] T.J.Richardson, M.A.Shokrollahi and R.L.Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," IEEE Trans. on Information Theory, vol. 47, n.2, pp. 619-637, Feb. 2001.
- [6] Y.Kou, S.Lin and M.Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," IEEE Trans. on Information Theory, vol. 47, pp. 2711-2736, Nov. 2001.
- [7] M.Spiser and D.A.Spielman, "Expander Codes IEEE Trans. on Information Theory," vol. 42, pp. 1710-1722, Nov. 1996.
- [8] A.Barg and G.Zémor, "Error exponents of expander codes," IEEE Trans. on Information Theory, vol. 48, pp. 1725-1729, June 2002.
- [9] J.-F. Cheng and R. J. McEliece, "Frequency-Efficient Coding with Low-Density Generator Matrices," IEEE Int. Symp. on Information Theory, Ulm, Germany, Jul. 1997.]
- [10] W.E.Ryan, An Introduction to LDPC Codes, "unpublished manuscript", 2003.
- [11] T.J.Richardson and R.L.Urbanke, "Efficient encoding of low-density parity-check codes," IEEE Trans. on Information Theory, vol. 47, n.2, pp. 638-656, Feb. 2001.
- [12] K. Deerga Rao, "Channel Coding Techniques for Wireless Communications", Osmania University.
- [13] Fernando Pujaico Riveira, "Algoritmos de Decodificação Abrupta para Códigos LDGM", Tese de Mestrado, Universidade Estadual de Campinas, 2011.
- [14] Sarah J. Johnson, "Introducing Low-Density Parity-Check Codes", University of Newcastle, 2006.