World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:2, 2012

Low Complexity, High Performance LDPC Codes Based on Defected Fullerene Graphs

Ashish Goswami, Rakesh Sharma

Abstract—In this paper, LDPC Codes based on defected fullerene graphs have been generated. And it is found that the codes generated are fast in encoding and better in terms of error performance on AWGN Channel.

Keywords—LDPC Codes, Fullerene Graphs, Defected Fullerene Graphs.

I. Introduction

Low density parity-check (LDPC) codes are the Linear Block Codes which perform close to the Shannon limit. It was proved in [1] that LDPC codes with column weight (number of 1's in code matrix column) $j \geq 3$ have a minimum distance that grows linearly with the block length (number of columns) n for given j and row-weight (number of 1's in matrix row) k and that the minimum distance for codes with $j = 2$ grows logarithmically with n. However, compared with $j \geq 3$ codes, codes with $j = 2$ are easier to implement and require less storage making them best for low complexity applications [2]. Column weight 2 codes can be generated from regular graphs called cages. LDPC codes based on cages were discussed in [3], and it was shown that the codes are suitable for magnetic storage devices [4].

One of the limitations of the codes generated by cages have limited code size which depends on the vertices and edges of these cages. Large size cages are very complex to design. In this paper we have used family of fullerene graphs to generate LDPC codes. Fullerene graphs have a property that it can be extended up to large number of vertices [5]. Another approach is taken in this paper, to generate LDPC codes from defected fullerene graphs. In this approach a defect has been created in the fullerene graph by removing a node while its edges are kept unaltered. We have analyzed the time taken in encoding of LDPC Codes and its Bit Error Rate (BER) performance over AWGN Channel. A typical configuration of the code generated from defected fullerene graph has given better BER performance and also took less time for encoding. We have compared the results with existing LDPC codes based on other known cages. The rest of the paper is organized as follows, in section II a short introduction to LDPC Codes is giving, in section III parity check matrix from defected fullerene graph is generated. Performance evaluation is giving in section IV followed by conclusion in section V.

Ashish Goswami is Lecturer at The Department of E & CE, National Institute of Technology, Hamirpur, INDIA (e-mail: goswami.ashish@rediffmail.com).

Rakesh Sharma is an Assistant Professor at The Department of E & CE, National Institute of Technology, Hamirpur, INDIA (e-mail: rakesh.sharma@nith.ac.in).

II. LDPC Codes

LDPC codes are the Linear Block Codes having low density of $1's$ in their parity check matrix H. For a typical LDPC Code of size $m \times n$, if W_c and W_r are the column and row weights respectively then $W_c << n$ and $W_r << m$. LDPC codes are said to be Regular if W_c and W_r are constant throughout the columns and rows. Otherwise it is said to be Irregular.

For encoding, we have to generate the generator matrix G. To generate the G matrix, H matrix should be converted into systematic form $[P_{m\times k} \quad I_{n-k}]$. This can be done by first converting H matrix into reduced row echelon form (rref) and then with some column permutations it is converted into systematic form. Generator matrix then can be easily generated as $[I_k \ P^T]$. Another way to convert H matrix from unsystematic form $[A \quad B]$ to systematic form is multiplication by B^{-1}, i.e. $B^{-1}[A \quad B] = [B^{-1}A \quad I]$. But if B is singular matrix then we have to go for the previous one. In term of complexity, the main step is the conversion of H into rref which takes maximum time in encoding of the codes. An efficient encoding method was developed by Richardson and Urbanke [6]. But this method still requires some rows and columns permutation which is further a time taking task.

For decoding there are several methods namely, Believe Propagation (BP), Sum-Product (SP), and Message Passing (MP). In this paper Log-domain Sum-Product algorithm [7] was used for decoding.

III. Design of Parity Check Matrix from Fullerene Graphs

The role of Parity Check Matrix is very important in the performance analysis of LDPC Codes. After the design of parity check matrix its respective generator matrix can be constructed. In this section the parity check matrix of the LDPC codes is generated with the help of fullerene graphs. A Fullerene graph is a cubic planar graph with all faces 5-cycles or 6-cycles. If the number of 5-cycles (pentagons) in a given Fullerene is p and number of 6-cycles (hexagons) is h. And if v be the number of vertices, and e be the number of edges then for Fullerene graph $p = 12$, $v = 2h + 20$ and $e = 3h+30$. Hence the size of the fullerene graph depends on the number of hexagons h. The extension of fullerene graphs were explained in [5].

Once fullerene graph is constructed its incidence matrix can be use as parity check matrix. After this corrosponding generator matrix has to be generated. Here an approach is taken to generate code with the help of defected fullerene graph. The defect is created in fullerene graph by removing a node while its edges are kept unaltered. The graph looks

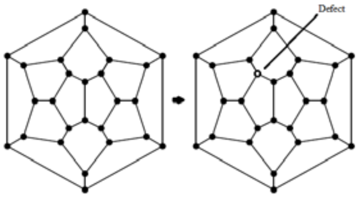


Fig. 1: Fullerene graph before and after removing node (showing defect)

similar to a hole formation in a semiconductor and for a typical graph this situation is shown in Fig. 1. The incidence matrix of the modified graph has been taken as parity check matrix for LDPC Codes. This type of configuration has an advantage that a typical column permutation of its H matrix make the B matrix non-singular and hence can be easily converted into systematic form without converting into rref form and hence saves time. Since the number of rows of the incidence matrix is reduces 1 due to a node removed and the number of columns remains same the code is strictly irregular and there is a slight change in code rate but for large code lengths it remains approximately 1/3. The fullerene graph and the defected fullerene graph used in simulation are of form $h = 3 + 6k$; where $k = 0, 1, 2, 3, ...$ [5].

IV. Performance Evaluation

The LDPC Codes generated from fullerene graphs and defected fullerene graphs has been simulated in MATLAB for BPSK modulated signals passed through AWGN channel for different SNR. The LDPC encoder and decoder objects, modulator and demodulator objects, and the channel have been generated by inbuilt standard MATLAB commands. The variance (σ^2) of additive noise can be calculated from the expression [8] of SNR given by:

$$SNR(dB) = 10 \log_{10} \frac{a^2}{2R\sigma^2} \qquad (1)$$

The codes based on fullerene and defected fullerene graphs have been generated and compared with the existing similar codes i.e. codes generated from cubic cages (cages having vertex degree 3). Some cage graphs result in too small codes for practical use. An expansion method [3] is therefore needed to get larger codes. Our main focus is on time elapsed during encoding and the BER performance on AWGN Channel.

We have simulated different codes having message code-lengths in the range of ≈ 125 and ≈ 250. Table I gives the comparison of time elapsed during encoding process between different codes. $[\times N]$ in Table I shows the N^{th} extension [3] so as to approximate the length of the codes. Figure 2 shows the BER performance on AWGN Channel. From the Table I, it is clear that the codes based on defected fullerene graphs takes less time for encoding. Codes based on other graphs in range of message length ≈ 125 takes approximately $3 - 6$ $seconds$ for encoding while codes based on defected fullerene graphs takes only 0.05 sec. Similarly, they take only 0.06 sec in comparison with the other codes in the range of message length $\approx$ 250 which take about $15 - 25$ $seconds$ for encoding. The reason behind it is that these codes are not required to be converted to rref form conversion during the process of conversion of H matrix into systematic form. We have also calculated encoding time for some higher message length codes. The parameters are shown in Table I.

We have simulated these codes for AWGN Channel and the BER performance were shown in Figures 2a & 2b for message lengths of $\approx$ 125 & $\approx$ 250 respectively. We can see from both the figures that as the size of graph increases the BER performance of the codes generated from that graph degrades but a revolutionary inprovement in BER performance occur when a defect is introduced. The codes which are generated from defected fullerene graphs gives BER under acceptable range for E_b/N_0 in between $2 - 2.5$ dB and $1.5 - 2$ dB for message lengths of ≈ 125 & ≈ 250 respectively. Hence it is clear that the codes generated from defected fullerene graphs outperforms from the codes generated from other similar graphs having similar message lengths.

V. Conclusion

In this paper we have generated the codes based on fullerene and defected fullerene graphs and the BER performance and encoding time is compared with the codes based on other cubic cages. We have simulated these codes to calculate the encoding time and its performance on AWGN Channel. The codes based on defected fullerene graphs shows less encoding time and hence having low encoding complexity. Also from the BER performance curves it has been seen that as the size of graph increases, the performance of the codes generated from that graph degrades but a revolutionary increment in performance occur when a defect is introduced. From both the graphs it is clear that the codes from defected fullerene graphs outperforms from the codes generated from other graphs and having similar message lengths. Hence it is clear that due to its low encoding complexity and good performance these codes can be used in applications where low complexity, high performance codes are required.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*, IRE Trans. Information Theory, vol. 8, no. 1, pp. 21-28, January 1962.

[2] Hongwei Song, Jingfeng Liu, B.V.K.V. Kumar, *Low Complexity LDPC Codes for Partial Response Channels*, IEEE Globecom 2002, vol. 2, pp. 1294-1299, November 2002.

[3] Gabofetswe Malema and Michael Liebelt, *High Girth Column-Weight-Two LDPC Codes Based on Distance Graphs*, EURASIP Journal on Wireless Communications and Networking Volume 2007, Article ID 48158, 5 pages.

[4] G. Malema, and M. Liebelt, *Low Complexity Regular LDPC codes for Magnetic Storage Devices*, Proc. of World Academy of Science, Engineering and Technology, Issue 7, 2005.

[5] Rashid Zaare-Nahandi, *Some Infinite Classes of Fullerene Graphs*, International Mathematical Forum, 1, 2006, no. 40, 1997-2002.

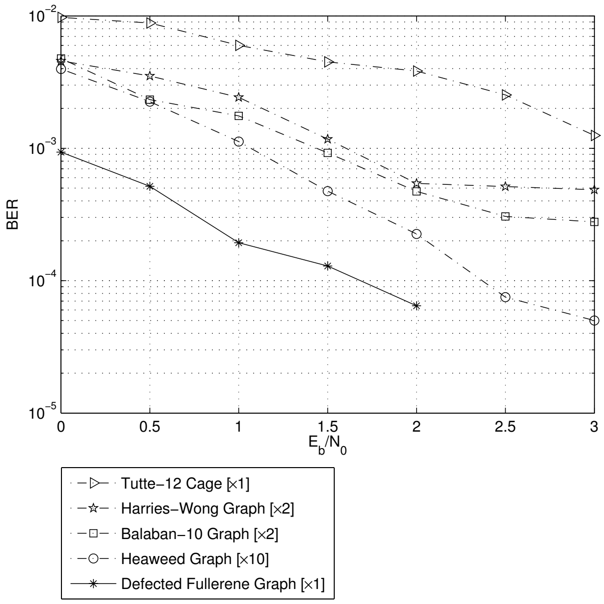
[6] T. J. Richardson and R. L. Urbanke, *Efficient Encoding of Low-Density Parity-Check Codes*, IEEE Trans. on Information Theory, vol. 47, no. 2, pp. 638-656, February 2001

[7] C Prasartkaew, S Choomchuay, *A design of parity check matrix for Irregular LDPC codes*, IEEE 9th International Symposium on Communications and Information Technology, ISCIT 2009, pp. 239-242, 2009.

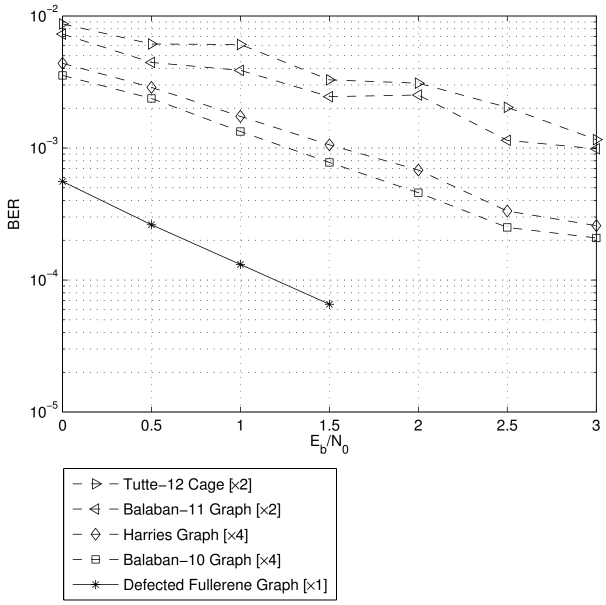
[8] D. J. C. MacKay and R. M. Neal, *Near Shannon Limit Perfoemance of Low Density Parity Check Codes*, IEEE Electronics Letters Vol. 32 No. 18, pp. 1645-1646, 29th August 1996.

TABLE I: Time elapsed in encoding of different codes generated from Cubic Cages

Codes generated from Cubic Cages		Code Rate	Size of H matrix	Required rref conversion	Time elapsed (sec) in Encoding
Petersen Graph (girth=5)	[$\times$12]	$1/3$	120×180	yes	4.074901
Heawood Graph (girth=6)	[$\times$10]	$1/3$	130×210	yes	5.482909
McGee (girth=7)	[$\times$5]	$1/3$	120×180	yes	4.067087
Levi Graph (girth=8)	[$\times$4]	$\approx 1/3$	116×180	yes	4.063403
Balaban-10 Graph (girth=10)	[$\times$2]	$\approx 1/3$	138×210	yes	5.785651
Harries Graph (girth=10)	[$\times$2]	$\approx 1/3$	138×210	yes	5.832270
Harries-Wong Graph (girth=10)	[$\times$2]	$\approx 1/3$	138×208	yes	5.656878
Balaban-11 Graph (girth=11)	[$\times$1]	$1/3$	112×168	yes	3.926893
Tutte-12 Graph (girth=11)	[$\times$1]	$\approx 1/3$	125×189	yes	4.888213
Fullerene Graph (girth=5)	[$\times$1]	$1/3$	122×183	yes	4.338537
Defected Fullerene Graph (girth=5)	**[$\times$1]**	$\approx 1/3$	121×183	no	$\mathbf{0.054425}$
Balaban-10 Graph (girth=10)	[$\times$4]	$\approx 1/3$	276×420	yes	24.479025
Harries Graph (girth=10)	[$\times$4]	$\approx 1/3$	276×420	yes	24.670780
Harries-Wong Graph (girth=10)	[$\times$4]	$\approx 1/3$	276×216	yes	24.166946
Balaban-11 Graph (girth=11)	[$\times$2]	$1/3$	224×336	yes	16.407000
Tutte-12 Graph (girth=11)	[$\times$2]	$\approx 1/3$	250×378	yes	20.362313
Fullerene Graph (girth=5)	[$\times$1]	$1/3$	242×363	yes	18.403568
Defected Fullerene Graph (girth=5)	**[$\times$1]**	$\approx 1/3$	241×363	no	$\mathbf{0.064095}$
Defected Fullerene Graph (girth=5)		$\approx 1/3$	1225×1839	no	$\mathbf{0.179094}$
(Higher Configurations)	[$\times$1]	$\approx 1/3$	6025×6039	no	$\mathbf{1.828735}$
		$\approx 1/3$	12025×18039	no	$\mathbf{6.072098}$



(a)



(b)

Fig. 2: Performance of codes based on Cubic Cages on AWGN Channel, (a) with message length ≈ 125, (b) with message length ≈ 250