Construction of Column-Weight-Two Irregular LDPC Codes with a Predetermined High Girth

Gan Srirutchataboom[1], Nut Tantibut[2], Piya Kovintavewat[3], Lunchakorn Wuttisittikulkij[4], Kritsada Mamat[5] and Watid Phakphisut[6]
[1,2,4]Chulalongkorn University, Bangkok, Thailand 10300
[3]Nakhon Pathom Rajabhat University, Nakhon Pathom, Thailand 73000
[5] Kasetsart University, Bangkok, Thailand 10900
[6]Bangkok Suvarnabhumi University, Bangkok, Thailand 10520
[1]srirutchataboon.research@gmail.com, [2]nutplane@hotmail.com, [3]piya@npru.ac.th, [4]lunchakorn.ww@gmail.com, [5]mkritsada1@gmail.com and [6]phwatid@gmail.com

Abstract: A girth is one of the parameters that can determine the performance of low-density parity-check (LDPC) codes. Because a full Tanner graph of a parity-check matrix, $\boldsymbol{H}$, can be generated from a small graph, we propose a novel algorithm for constructing irregular column-weight-two LDPC codes with a predetermined large girth. The proposed algorithm carefully add edges and vertices in the graph until it fully expands to represent the required $\boldsymbol{H}$ matrix. Simulation results show that the proposed algorithm can provide the $\boldsymbol{H}$ matrix with a very high girth if compared to a well-known progressive-edge growth (PEG) algorithm, and also yields a good bit-error rate performance.

1. Introduction

Low-density parity-check (LDPC) codes were originally proposed by Gallager in 1962 [1], and rediscovered by Mackay and Neal in 1996 [2]. Since then, various different code construction methods have been proposed, such as finite geometry [3] and progressive-edge growth (PEG) [4] algorithms, with some good codes approaching the Shannon limit [2]. This paper focuses on the construction of column-weight-two LDPC codes. Although this class of codes is known to have the minimum distance increased logarithmically with code length, these codes offer other advantages, e.g., less computational complexity when compared to the codes with higher column weights, and good potential in partial response channels [5]. Moreover, it is used to design very sparse non-binary LDPC codes with small to moderate code length in high Galois field orders, which can provide high performance [6].

In literature, Malema and Liebelt [7] proposed to use known distance graphs or cages to construct column-weight-two or (j, k) LDPC codes, where $j = 2$ is a column weight and k is a row weight. By representing the vertices and edges of the cages as rows and columns of the parity-check matrix, $\boldsymbol{H}$, the column-weight-two LDPC code with girth twice that of the shortest cycles of cages can be derived. Venkiah *et al.* [8] presented a randomized PEG algorithm to design cages, given a target girth. Additionally, Tao *et al.* [9] introduced (k, k) quasi-cyclic LDPC codes instead of cages, that can achieve a large girth of 36 for $(2, 3)$ LDPC sample codes.

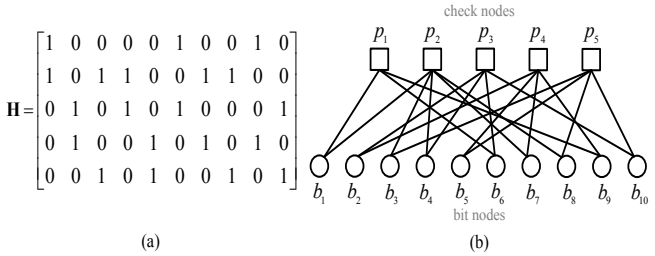

Figure 1. The $\boldsymbol{H}$ matrix of size 5×10 and its corresponding Tanner graph.

In this paper, we propose a novel method to design irregular column-weight-two LDPC codes with a given large girth. Our proposed method starts from a *ring* structure, where its vertices and edges correspond to rows and columns of the $\boldsymbol{H}$ matrix, respectively. Then, an edge or a series of alternating edges and vertices are recursively added to the initial ring until the code length is satisfied. At each round of adding process, a certain constraint is imposed to ensure that the girth of the $\boldsymbol{H}$ matrix will provide at least the target girth. Finally, the resulting $\boldsymbol{H}$ matrix satisfying the target girth, code length, and column-weight-two, using less parity-check bits, will be compared with the $\boldsymbol{H}$ matrix obtained from the PEG algorithm.

2. Definitions and Notations

Consider the $\boldsymbol{H}$ matrix of size $M \times N$, where M is the number of check nodes (or parity bits) and N is the number of bit nodes (or a code length). In general, the $\boldsymbol{H}$ matrix can be represented by a Tanner graph [10], which consists of two sets, namely, a set of check nodes $\{p_1, p_2, \ldots, p_M\}$ and a set of bit nodes $\{b_1, b_2, \ldots, b_N\}$. Let d_{p_i} be the degree of the i-th check node that corresponds to the number of non-zero elements in the i-th row of the $\boldsymbol{H}$ matrix, and d_{b_j} be the degree of the j-th bit node that corresponds to the number of non-zero elements in the j-th column of the $\boldsymbol{H}$ matrix. Fig. 1 illustrates an example of the $\boldsymbol{H}$ matrix of size 5×10 and its corresponding Tanner graph, where $d_{p_1} = 3, d_{p_2} = 5, d_{p_3} = d_{p_4} = d_{p_5} = 4$. and $d_{b_j} = 2$ for $j \in \{1, 2, \ldots, 10\}$.

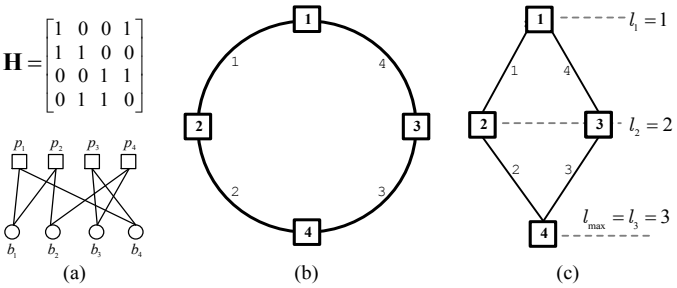

Figure 2. (a) The $\boldsymbol{H}$ matrix and its Tanner graph with $g = 8$, (b) its corresponding ring with 4 vertices and 4 edges, and (c) the 3 layers (l_1, l_2, l_3) in the ring.

Here, we map a check node p_i and a bit node b_j of the Tanner graph as a *vertex* and *edge* into a graph based on a ring structure denoted as G, respectively. Consider the $\boldsymbol{H}$ matrix with girth $g = 8$ and its corresponding Tanner graph in Fig. 2(a), in which it can be mapped into graph G with 4 vertices and 4 edges, as illustrated in Fig. 2(b). Note that each vertex is represented by a rectangular box $\square$, where the number i inside $\square$ is the i-th check node, whereas each edge is represented by a line, where the number j labeled close to the line is the j-th bit node. For example, the vertex 1 has two lines (labeled as 1 and 4) connected to it, which means the first row in the $\boldsymbol{H}$ matrix will have an element '1' on the 1-st and 4-th columns as shown in Fig. 2(a).

If we pick any vertex (e.g., vertex 1) in this graph G, and pull it up such that the remaining vertices are drawn closer. Hence, we can define the number of layers as the number of groups of the vertices that are of different heights. For instance, there are 3 layers denoted as l_i for $i \in \{1, 2, 3\}$, where the vertex 1 is in l_1, and the vertex 4 is in the last layer denoted as $l_{\max} = l_3$, as depicted in Fig. 2(c). With this ring structure, the girth of the $\mathbf{H}$ matrix is double of the number of vertices in ring. For example, the $\mathbf{H}$ matrix in Fig. 2(a) has $g = 8$, whereas the ring in Fig. 2(b) has 4 vertices and 4 edges.

3. Proposed algorithm

To construct a column-weight-two LDPC code with length N and target girth g_t, where g_t is an even number, we first create an initial ring of graph G with the number of vertices and edges equal to $g_t/2$. Hence, we obtain the layer

$$l_{\max} = \begin{cases} g_t/4 + 0.5, & \text{if } g_t/2 \text{ is odd} \\ g_t/4 + 1, & \text{if } g_t/2 \text{ is even} \end{cases}. \qquad (1)$$

Notice that the initial ring of graph G will represent the parity-check matrix of size $g_t/2 \times g_t/2$, which means we need to add $N - g_t/2$ bit nodes (or edges) in this graph to satisfy the code length of N.

The proposed algorithm recursively adds edge or a series of alternating edges and vertices in G until N edges (bit nodes) are obtained, while keeping every local girth equal to g_t, which can be explained as follows.

1) Choose any vertex with the lowest degree in G, which will be referred to as p_i.

2) Compute $l_{\max}$, where p_i is considered as in l_1. Then,

– If $g_t \leq 2l_{\max}$, we add one edge in G, connected between p_i and the vertex with the lowest degree in $l_{\max}$, where a new local girth will be equal to $2l_{\max}$.

– If $g_t > 2l_{\max}$, we add $g_t/2 - l_{\max}$ vertices and $g_t/2 - l_{\max} + 1$ edges on a single line connected between p_i and the vertex with the lowest degree in $l_{\max}$. It should be pointed out that if the number of additional edges in this step causes the total number of edges in G to exceed N, we will ignore this Step and go to Step 4. This will result in the actual code length less than N.

3) Go back to Step 1 until the number of edges is equal to N.

4) Transform the resulting G into the $\boldsymbol{H}$ matrix.

When completing this procedure, we obtain the $\boldsymbol{H}$ matrix with a girth of g_t and a code length approximately equal to N.

Example 1: Suppose we want to design the $\boldsymbol{H}$ matrix with $g_t = 8$ and $N = 9$. We first generate a ring with $g_t = 8$ as displayed in Fig. 2(b). At first, all vertices have $d_{p_i} = 2$ for $i \in \{1, 2, 3, 4\}$. Then, if we choose the vertex 2, we get $l_{\max} = 3$. Because $g_t > 2l_{\max}$, we add one vertex (i.e., vertex 5) and two edges (labeled as 5 and 6) between the vertex 2 (in l_1) and the vertex 3 (in $l_{\max}$), as shown in Fig. 3(a) together with its corresponding matrix form. Then, the vertices 1 and 4 now have the lowest degree. If we expand the graph from the vertex 1, which will be considered as in l_1, we get $l_{\max} = 3$. Again, because $g_t > 2l_{\max}$, we add one vertex (i.e., vertex 6) and two edges (labeled as 7 and 8) between the vertex 1 (in l_1) and the vertex 4 (in $l_{\max}$) as displayed in Fig. 3(b). Since the vertices 5 and 6 have the lowest degree, if we expand a graph from the vertex 6, we get $l_{\max} = 4$. Because $g_t = 2l_{\max}$ in this case, we add only one edge (labeled as 9) between the vertex 6 (in l_1) and the vertex 5 (in $l_{\max}$) as depicted in Fig. 3(c). Therefore, because the total number of edges in the graph is now equal to $N = 9$, we stop the algorithm and obtain the $\boldsymbol{H}$ matrix as given in Fig. 3(c).

4. Simulation Results

In this section, we first demonstrate that the proposed code construction algorithm can produce column-weight-two LDPC codes of various lengths with arbitrarily predefined target girths. Fig. 4 depicts several constructed sample codes of small to moderate block lengths (i.e., N = 256, 512, 1024, 2048, and 4096), and each of which has different target girths varying from 8 to 24. Note that each number inside the parenthesis in Fig. 4 denotes the number of check bits (M) used to generate each result. As can be seen, for each code length, a wide range of girths can be achieved where codes with higher girth are accomplished with the expense of reduced achievable

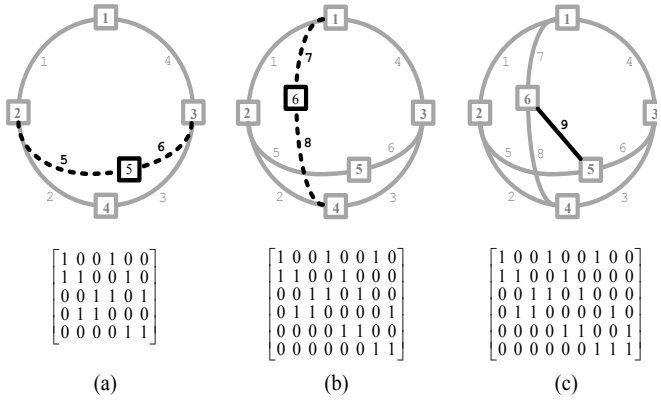

Figure 3. How the proposed algorithm expand an initial ring to obtain the $\boldsymbol{H}$ matrix with $g = 8$ and $N = 9$, and the corresponding matrix form for each case.

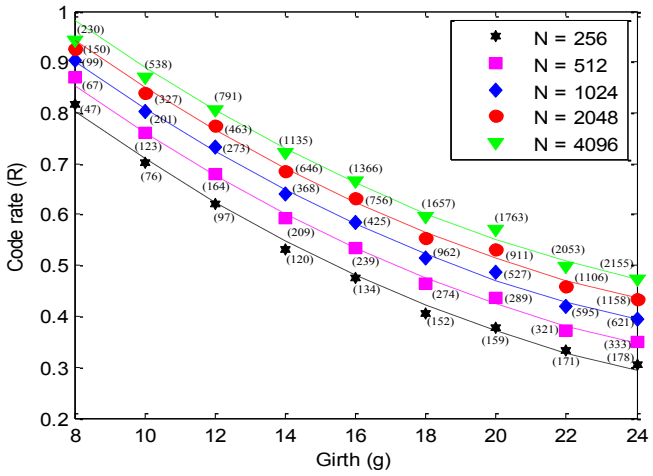

Figure 4. The code rate (R) versus the girth (g) for different N's and M's, where a solid line represents the estimated R from (2).

code rates ($R = 1 - M/N$), i.e., more check bits required. For the same target girth, codes with larger block lengths can be constructed at higher code rates than those with smaller block lengths.

From Fig. 4, it appears that there is a clear relation between the achievable code rates and the target girths for each code length. Hence, we derive a mathematical formulae to estimate R for given g_t and N by using a curve fitting technique with numerous constructed codes to obtain

$$R = 0.00103g_t^2 - 0.0647g_t + 1.3532 + 0.8\left(\frac{x-10}{x+10}\right), \quad (2)$$

where $x = \log_2(N)$. As shown in Fig. 4, the estimated code rate from (2) closely matches the actual one for N ranged from 256 to 4096, especially when g_t is large.

Next, we compare the minimum girth of codes obtained from the proposed algorithm with the PEG algorithm. Fig. 5 shows that, for different values of M and

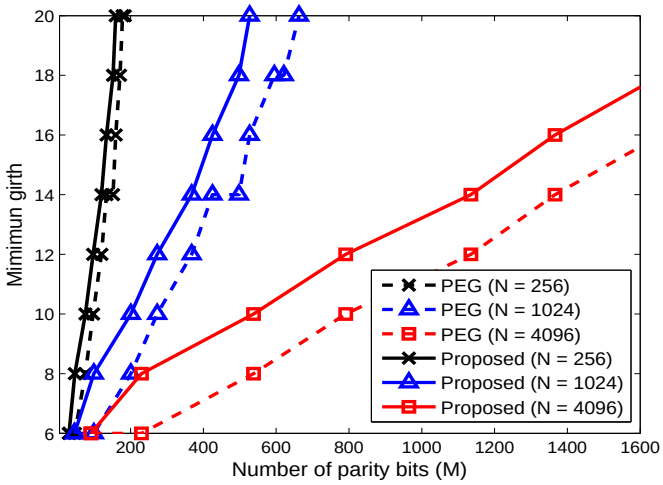

Figure 5. The number of girths as a function of M's for different N's.

N, the proposed codes can in most cases offer a minimum girth higher than that of PEG codes by two. For the same fixed target girth, the proposed codes require fewer check bits than the PEG codes, i.e., higher code rate. For example, when $g_t = 10$ and $N = 4096$, the proposed code provides a code rate of 0.8687, while the PEG code gives a code rate of 0.8069. In some other cases (not shown here), the proposed codes can provide higher girths than the PEG codes by four or even six.

Finally, to evaluate the bit-error rate (BER) performance of the proposed algorithm, we consider an additive white Gaussian noise (AWGN) channel model, where a binary input sequence $m_k \in \{0, 1\}$ of length $N-M$ bits is encoded by an LDPC encoder and is mapped to an N-bit coded sequence $c_k \in \{\pm 1\}$. Then, the received sequence is given by $y_k = c_k + w_k$, where w_k is AWGN with zero mean and variance σ^2. At the receiver, the received sequence y_k is decoded by an LDPC decoder implemented based on a message passing algorithm [1] with 30 iterations. In simulation, the signal-to-noise ratio is defined as

$$\text{SNR} = 10\log_{10}\left(\frac{1}{\text{R}\sigma^2}\right), \quad (3)$$

in decibel (dB). Moreover, each BER point is computed based on a minimum number of 50000 data packets and 1000 error bits.

Fig. 6 illustrates the BER performance of three different schemes, including the code from a random $\boldsymbol{H}$ matrix, where the parenthesis (M, N) denotes the number of parity bits (M) and coded bits (N), and each code has approximately the same code rate of 0.5. Apparently, the proposed algorithm performs better than other algorithms. It should also note that codes with larger block lengths have higher girths and also perform better than codes with smaller lengths.

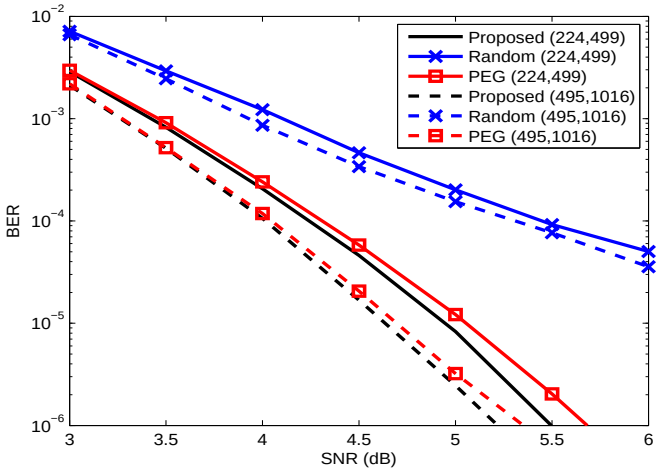

Figure 6. BER performance with different codes.

5. Conclusion

This paper proposes a novel approach to construct column-weight-two irregular LDPC codes with arbitrary high girth, whose key idea is to construct a Tanner graph from the a small graph based on a ring structure. Given this graph with high girth g_t and code length N, the proposed algorithm carefully adds the vertices represented by check nodes and the edges represented by bit nodes inside the graph, while maintaining every local girth equal to g_t, until it contains N edges. Simulation results indicate that the proposed algorithm can yield a higher girth and a lower BER than the PEG algorithm for all code rates and lengths.

References

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.

[3] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[4] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[5] H. Song, J. Liu, and B. V. K. Vijaya Kumar, "Large girth cycle codes for partial response channels," *IEEE Trans. Magn.*, vol. 40, no. 4, pp. 3084–3086, July 2004.

[6] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular $(2, d_c)$-LDPC codes over GF (q) using their binary images," *IEEE Trans. Comm.*, vol. 56, no. 10, pp. 1626–1635, Oct. 2008.

[7] G. Malema and M. Liebelt, "High girth column-weight-two LDPC codes based on distance graphs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, ID 48158, 2007.

[8] A. Venkiah, D. Declercq, and C. Poulliat, "Design of cages with a randomized progressive edge-growth algorithm," *IEEE Comm. letters*, vol. 12, no. 4, pp. 301–303, Apr. 2008.

[9] X. Tao, L. Zheng, W. Liu, and D. Liu, "Recursive design of high girth $(2, k)$ LDPC codes from (k, k) LDPC codes," *IEEE Comm. letters*, vol. 15, no. 1, pp. 70–72, Jan. 2011.

[10] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 5, pp. 533–547, Sept. 1981.