International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017(O)
Volume 1 Issue 3 (September 2014) ISSN: 2349-7009(P)
www.ijiris.com

Parity check matrix and construction methods of Low Density Parity Check code

Prof. Isha Gautam
Electronics and communication Department,
Vadodara Institute of Engineering

Abstract--Low-density parity-check (LDPC) codes are linear block codes with sparse parity-check matrices. In this paper, a brief description of some construction methods used to generate LDPC codes is given. These methods generally fall into two categories: random and analytical. The randomly constructed codes include those from Gallager and Mackay. The analytical construction methods described are the codes from finite geometries. One type is randomly constructed based on the original prescription of Gallager. The other is built analytically from Euclidean and projective geometry. Their performance can be examined using the decoding algorithm based on likelihood difference.

I. INTRODUCTION

LDPC Code Overview

In information theory, the Shannon's channel coding theorem is considered to have been stimulating the development of error control codes. It states that all the data rates r_b less than the channel capacity C can be achieved with an arbitrarily small probability of error P_e, where C is given by the Shannon-Hartley formula [1]:

$$C = B \log_2[S/N] \text{ (Bits per second)} \quad (1.1)$$

Here, B is the channel bandwidth in Hz and S/N is the signal to noise ratio (SNR). The SNR is related to the "bit energy to one sided noise power spectral density ratio"

$$\frac{S}{N} = \frac{r_b E_b}{B N_0} = \frac{r_b}{B} \times \frac{E_b}{N_0} \quad (1.2)$$

Using Equation (1.2), Equation (1.1) can be written as:

$$\frac{C}{B} = \log_2\left(1 + \left(\eta_{max} \frac{E_b}{N_0}\right)\right) \quad (1.3)$$

Or

$$\frac{E_b}{N_0} = \frac{2^{\eta max} - 1}{\eta_{max}} \quad (1.4)$$

Equation (1.4) is known as the Shannon limit. It gives the required E_b/N_0 to transmit data at a rate close to the channel capacity. This limit is always used as a benchmark to evaluate a coding-modulation scheme. Turbo codes and LDPC codes have been reported to have performance very close to the Shannon limit [2].

LDPC codes, also known as Gallager codes, were devised by Gallager in early 60's [3]. As a class of linear block codes, they are distinguished by sparse binary parity- check matrices. In each matrix, every row has a fixed number (j) of 1's and every column also has a fixed number (k) of 1's. However, at that time, computing power was not enough to show their effectiveness, therefore LDPC codes have been forgotten until recently [2]. Mackay and Neal are said to have "rediscovered" Gallager codes by pointing out their excellent functioning using the decoding algorithm based on sum-product algorithm [4]. From the original prescription of Gallager, Luby et al. marked an important progress of LDPC codes by introducing irregular codes [5]. Another advance of LDPC codes was the introduction of irregular codes over GF(q) (q>2), by Davey and Mackay. In [6], this class of LDPC codes was shown to have remarkably improved performance over the codes in GF(2).

II. CONSTRUCTION OF LDPC CODE

In this part we describe some constructions of regular and irregular LDPC codes. The methods used to construct the parity-check matrices of LDPC codes have fallen into two main categories: random and analytical methods. However, according to

Johnson and Weller [10], random construction methods are still dominant, as the LDPC codes created analytically only make up a small part. It has been shown that, the iterative sum-product decoding algorithm can converge at the optimal solution if the Tanner graph has no cycle [8]. Shorter cycles will worse degrade the algorithm.

Review of some useful concepts in dealing with constructions of LDPC codes is presented below.

Tanner graph: Tanner graph is used to represent the relationship between the codeword bits and parity check bits of a linear block code. Tanner graphs have been generalized to become factor graphs [7].
Cycle: A cycle in a Tanner graph is a sequence of connected codeword nodes and parity check nodes that begin and conclude at the same node and no other nodes can appear in the sequence more than once.
Length: The length of a cycle is the number of edges in the cycle.
Girth: The girth of a Tanner graph is the length of its shortest cycle.
Degree: The degree of a node in the Tanner graph is the number of edges connected to it.
Figures 1 and 2 illustrate two cycles of length-4 and length-6 respectively, and their corresponding parity check matrices.

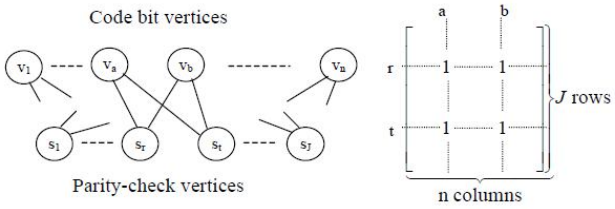

Figure 1 A length 4 cycle in Tanner graph and corresponding parity check matrix.

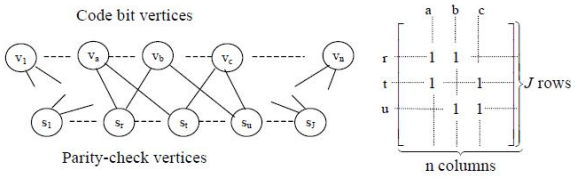

Figure 2 A length 6 cycle in Tanner graph and corresponding parity check matrix.

The roles of codeword vertices and parity-check vertices are equal in forming cycles. Therefore, using I or its transpose, I^T as the parity-check matrix will create the same set of cycles.

In particular, for length-4 cycle case, the three statements below are equivalent for a LDPC code:

- There is no length-4 cycle in the Tanner graph.
- The number of overlapping 1's between any two rows is smaller than 2.
- The number of overlapping 1's between any two columns is smaller than 2.
- These remarks will be used to simplify the proofs related to cycle lengths and girth of LDPC codes and in checking the overlap in the simulation programs as well.

III. RANDOMLY CONSTRUCTED CODES

A. Gallager's Prescription

In the original paper of Gallager [3], LDPC code is defined to be a type of linear block code whose parity check matrix H is a sparse matrix, i.e. a matrix that mainly contains 0's and a few 1's. Additionally, in this matrix, every row has the same number of 1's and every column has the same number of 1's. In particular, Gallager defines a (n, j, i) LDPC code as a code that has a block length of n, number of 1's in each column of the parity check matrix of j and in each row of i. A parity check matrix of this type is presented in Figure 3

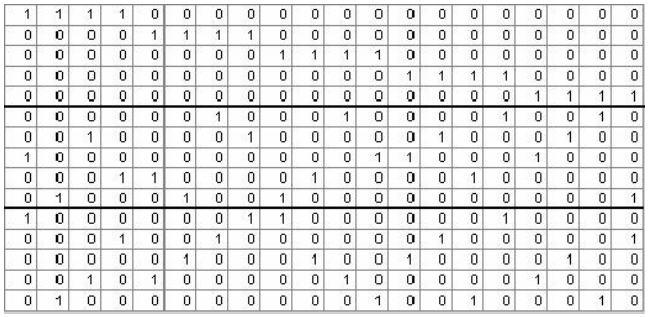

Figure 3 A (20, 3, 4) LDPC parity check matrix.

This matrix can be divided into three equal sub-matrices, each having the column weight of 1, as shown in Figure 3. The first sub-matrix is a special one: The 1's are placed in a downward manner and the whole sub-matrix looks like an identity matrix with each column being repeated four times. In fact, the sub-matrix just has to be a random column permutation of this special structure. It can also be readily seen that a random column permutation of the first sub-matrix (or any of subsequent sub-matrices) will have exactly four 1's in each row and a single 1 in each column. Vice versa, any (5×20) sub-matrix that has four 1's in each row and a single 1 in each column is just a column permutation of the first sub- matrix. A representation of this type of parity check matrices is given in Figure 4 [2].

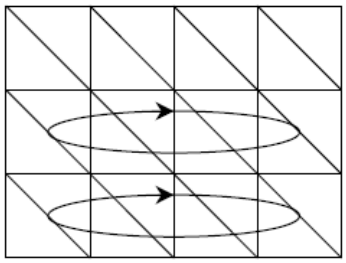

Figure 4 A representation of Gallager's construction with row weight 4 and column weight 3. A square with a diagonal line represents an identity matrix. An ellipse with an arrow represents a random column permutation.

In general, a (n, j, i) matrix can be divided into j sub-matrices that have column weight of 1. The number of rows of a (n, j, i) matrix is given by nj/i (j is the number of 1's in each column, therefore the total number of 1's is nj, and i is the number of 1's in each row). Thus, each sub-matrix will have n/i rows.

Considering LDPC codes as a type of linear block code, we can calculate the code rate R of a (n, j, i) LDPC code as follows: Firstly, assume that all the rows of H are linearly independent (or H is full rank), the rank of H (calculated in modulo-2 arithmetic) will be the same as the number of rows J, which is nj/i (J < n as J=n-k). Then the generator matrix G will have the dimension of(n-nj/i)×n (j < i as J < n) and the code rate R will be

$$R = (n-nj/i)/n = 1 - j/i.$$

When the rows of H are not linearly independent, the rank of H will be smaller than nj/i, and the number of rows of G will be larger than n-nj/i, which results in R > 1 – j/i. So, in general, we have:

$$R \geq \frac{n-n\frac{j}{i}}{n} = 1 - \frac{j}{i} \qquad (1.5)$$

B. Mackay's Construction

The presence of short cycles in the Tanner graph of an LDPC code will degrade the iterative decoding algorithms. Consider the portion of a Tanner graph given in Figure 5.

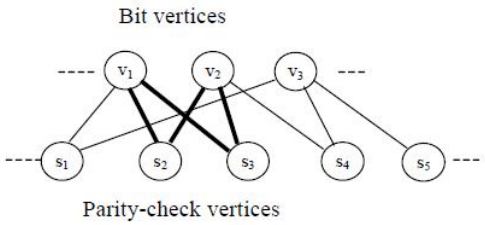

Figure 5 A length-4 cycle and its effect on the iterative decoding algorithms.

Three noise symbols are connected to five checks as shown in Figure 5. The cycle of length 4 is drawn by the bold lines. If all these three noise symbols are erroneous (changed from 1 to 0 or vice versa), only the right-most check (s_5) may be warned. According to Davey, this situation can degrade the performance of the decoding process [2].

Therefore, Mackay's constructions are aimed at decreasing the short cycles in the Tanner graphs of LDPC codes. Length-4 cycles can be avoided by constructing the parity check matrix with the number of overlapping 1's between any two rows less than 2. Some constructions by Mackay, which are free of length-4 cycles, are presented in this section.

Construction 1A: The parity-check matrix is created randomly with fixed column weight j and the row weights are kept to be as regular as possible. The overlap between any two rows is less than 2 (Figure 6 a).

Construction 2A: Similar to construction 1A, except there are a number (up to J/2, where J is the number of rows of the parity check matrix) of weight-2 columns. There is no overlapping 1 between any pair of these weight-2 columns. They can be constructed by stacking two identity matrices of order J/2 as shown in Figure 6 b. The remaining columns are built randomly with weight 3 and the row weight as regular as possible.

Construction 1B and 2B: To avoid the cycles of length less than l (e.g. l=6), a small number of columns of a parity check matrix resulted from constructions 1A and 2A, respectively, are deleted.

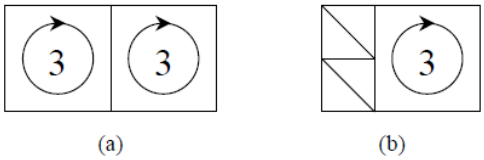

Figure 6 Mackay's construction (a) Construction 1A with column weight 3 and R=1/2. (b) Construction 2A with R=1/3. A circle with an integer inside a square shows the number of permutation matrices superposed on the square. A square with a diagonal represents an identity matrix.

IV. ANALYTICAL CONSTRUCTION METHODS

A. Regular Ldpc Codes Based On Finite Geometries

There are four classes of LDPC codes constructed using Euclidean and projective geometries over finite fields. These codes are shown to have good distance properties and Tanner graphs with girth 6. Their performance is reported to be very good with iterative decoding. Moreover, their distinguishing characteristic is that they can have cyclic or "quasi-cyclic" form, therefore they can be encoded using simple feedback shift registers. Some techniques to extend or shorten these codes are also presented in this paper and the long extended codes are shown to be just several tenths of a dB below the Shannon limit. The description of linear block codes based on finite geometries can also be found in [11].

The four classes of LDPC codes based on Euclidean and Projective Geometry are:

- Type-I Euclidean geometry (EG) LDPC codes
- Type-II Euclidean geometry (EG) LDPC codes
- Type-I projective geometry (PG) LDPC codes
- Type-II projective geometry (PG) LDPC codes

Type - I Euclidean Geometry LDPC Codes

Consider an m-dimensional Euclidean geometry with components from the Galois field GF(2^s) where m and s are two positive integers. This geometry, denoted EG(m, 2^s), is made up of all the m-tuples (a_0, a_1, …, a_{m-1}), where ai's come from GF(2^s). These tuples are also known as points and it can be seen that there is a total of 2ms points in EG(m, 2^s). The tuple with all the components equal to zero is called the origin. An EG(m, 2^s) can be considered as a vector space of the 2^{ms} tuples over GF(2^s). In this EG, a line is either a one-dimensional subspace or a coset of a one- dimensional subspace, and therefore, composed of 2^s points. All the 2^s points of a line in EG(m, 2^s) can be represented by the formula:

$\{a_0 + \beta_a\}$

Where $\beta \in GF(2^s)$, a_0 and a are two linearly independent points. Each value of β will give a point in the line.

When $a_0=0$, the line $\{\beta_a\}$ contains the origin (at $\beta=0$). $\{a_0+\beta_a\}$ and $\{b_0+\beta_a\}$ with $a_0 \neq b_0$ and $a \neq 0$ are two parallel lines. Two lines $\{a_0+\beta_{a1}\}$ and $\{a_0+\beta_{a2}\}$ with $a_1 \neq a_2$ intersect at the point a_0. It has been shown that there are

$(2^{ms}-1)/(2^s-1)$

lines in EG(m, 2^s) containing a given point a_0. The number of lines in EG(m, 2^s) is given by:

$2^{(m-1)s}(2^{ms}-1)/(2^s-1)$

The Euclidean geometry EG(m, 2^s) can be considered as the GF(2^{ms}), the extension field of GF(2^s). The reason is that GF(2^{ms}) also has 2ms elements and each element can also be regarded as an m-tuple over GF(2^s). The 2ms points of the EG(m, 2^s) therefore can be represented by $0,1,\alpha,K,\alpha^{2ms-2}$, as in the GF($2^{ms}$).

Now let $H_{EG}^{(1)}$ (m, s) denote a matrix with the elements from GF(2), which satisfies:

- The rows are the incident vectors of all the lines in EG(m, 2^s) that do not contain the origin. The matrix therefore has a total of J= $[2^{(m-1)s}-1](2^{ms}- 1)/(2^s-1)$ rows. (The total number of lines subtracted by the number of lines crossing the origin).
- The columns correspond to n=2^{ms}-1 points different from the origin and are placed in the order $1,\alpha,\ldots..,\alpha^{2ms-2}$.
- The value of the i^{th} element in a row is the coefficient of α^i in the represented point.

Type-II Euclidean Geometry LDPC Codes

Consider the matrix $H_{EG}^{(2)}$ (m, s) = $[H_{EG}^{(1)}$ (m, s)$]^T$. This matrix has 2^{ms}-1 rows corresponding to the non origin points of EG(m, 2^s) and J=$[2^{(m-1)s}-1](2^{ms}-1)/(2^s-1)$columns corresponding to the lines that do not contain the origin. The column weight is $\gamma=2^s$ and the row weight is $\rho=(2^{ms}-1)/(2^s-1)-1$. This matrix also has the overlap between any two rows or columns of less than 2.

If we choose $H_{EG}^{(2)}$ (m, s) to be the parity check matrix, we also have an LDPC code. This code is named the type-II m-dimensional EG-LDPC code with block length of J=$[2^{(m-1)s}-1](2^{ms}-1)/(2^s-1)$

And the minimum distance is at least 2^s+1. Notice that $H_{EG}^{(1)}$ (m, s) and $H_{EG}^{(2)}$ (m, s) have the same rank, therefore the two corresponding codes have the same number of parity check nodes. Also, it can be easily seen that they have the same cycle distribution.

The code corresponding to the parity check matrix $H_{EG}^{(2)}$ (m, s) , $C_{EG}^{(2)}$ (m, s) is said to be not cyclic but quasi-cyclic [9]. Basically, this means that the parity check matrix $H_{EG}^{(2)}$ (m, s) , whose dimension is $(2^{ms}-1) \times \{[2^{(m-1)s}-1](2^{ms}-1)/(2^s-1)\}$, can be rearranged (by column exchange) into (2^{ms}-1) sub matrices:

$H_{EG}^{(2)}(m, s) = [H0, H1, \ldots\ldots, H_{2ms-2}]$

where the dimension of each sub matrix is $(2^{ms}-1) \times \{[2^{(m-1)s}-1]/(2^s-1)\}$ and H_i $(1 \leq i \leq 2^{ms}-2)$ is the i^{th} vertical downward shift of H_0.

Type-I Projective Geometry (PG) LDPC Codes

Suppose that α is a primitive element of $GF(2^{(m+1)s})$, where $GF(2^{(m+1)s})$ is an extension field of $GF(2^s)$.

Let $n=[2^{(m+1)s}-1]/(2^s-1)$ and $\beta=\alpha^n$. Subsequently, the order of β is (2^s-1). Then 0,1, β, $\beta^2, \ldots., \beta^{2s-2}$ the 2^s elements of $GF(2^s)$. The $[2^{(m+1)s}-1]$ non zero elements of $GF(2^{(m+1)s})$ can be arranged into n separate subsets:

$\{\alpha^i, \beta\alpha^i, \beta^2\alpha^i, \ldots., \beta^{2s-2}\alpha^i\}$ where $0 \leq i < n$.

Each subset can be represented by the first element $(\alpha^i) = \{\alpha^i, \beta\alpha^i, \beta^2\alpha^i, \ldots., \beta^{2s-2}\alpha^i\}$ The n elements $(\alpha^0), (\alpha^1), (\alpha^2), \ldots (\alpha^{n-1})$ form an m-dimensional projective geometry over $GF(2^s)$, denoted $PG(m, 2^s)$. The main difference between a projective geometry and a Euclidean geometry is that in PG, 2^s-1 tuples in $(\alpha^i) = \{\alpha^i, \beta\alpha^i, \beta^2\alpha^i, \ldots., \beta^{2s-2}\alpha^i\}$ are regarded as the same point while in EG, each tuple corresponds to a different point.

The parity check matrix constructed from the lines and points of the $PG(m, 2^s)$ in the same way as in Type-I EG-LDPC codes, $H_{PG}^{(1)}(m, s)$, has the following features:

- It has $J=(2^{ms} + \ldots + 2^s + 1)(2^{(m-1)s} + \ldots + 2^s + 1)/(2^s + 1)$ rows and $n=(2^{(m+1)s} - 1)/(2^s - 1)$ columns
- The row weight is $\rho=2^s+1$.
- The column weight is $\gamma=(2^{ms}-1)/(2^s-1)$.
- The number of overlapping 1's between any two rows (therefore, any two columns) is at most 1.
- Thedensity $r=(2^{2s}-1)/(2^{(m+1)s}-1)$, which is guaranteed to be small with $m \geq 2$.

The code specified by this parity check matrix is therefore a regular LDPC code. According to [9], this code is cyclic and can be encoded using linear feedback shift registers.

Type-II Projective Geometry (PG) LDPC Codes

Similar to the derivation of the type-II EG-LDPC codes, type-II PG-LDPC codes are also constructed based on $H_{PG}^{(2)}(m, s) = [H_{PG}^{(1)}(m, s)]^T$. This subclass of codes based on finite geometries is also quasi-cyclic [9]

V. CONCLUSION

Random construction methods are dominant, as the LDPC codes created analytically only make up a small part. Shorter cycles will worse degrade the algorithm. Until now, to prevent short cycles, random construction methods have mainly been relying on the sparsity of the parity-check matrix. Therefore, random construction methods become less effective for the codes with small block lengths.

REFERENCES

[1]B. Vucetic, J. Yuan, Turbo Codes – Principles and Applications, Kluwer Academic, 2000.

[2]M. C. Davey, "Error-correction using low-density parity-check codes," Ph.D. dissertation, University of Cambridge.

[3]R. G. Gallager, "Low density parity check codes," IRE Transactions on Information Theory, IT-8, pp. 21-28, January 1962.

[4]D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," July 2002, http://www.inference.phy.cam.ac.uk/mackay/CodesGallager.html

[5] M. G. Luby, M. Mitzenmacher and M. A. Shokrollahi and D. A. Spielman, “Analysis of low density codes and improved designs using irregular graphs,” July 2002, http://www-math.mit.edu/~spielman/Research/irreg.html
[6]M. C. Davey and D. J. C. Mackay, “Low density parity check codes over GF(q),” IEEE Communication Letters, Volume 2, June 1998.
[7]F. R. Kschischang, B. J. Frey and H. Loeliger, “Factor graphs and the sum product algorithm,” IEEE Transactions on Information Theory, vol. 47, pp.498-519, Feb 2001.
[8]R. J. McEliece, D. J. C. Mackay and J. F. Cheng, “Turbo decoding as an instance of Pearl’s belief propagation algorithm,” IEEE Journal on Selected Areas in Communications, Vol.16, No.2, February 1998.
[9]Y. Kou, S. Lin and M. Fossorier, “Low density parity check codes based on finite geometries: A rediscovery and new results”, IEEE Transactions on Information Theory, Oct 1999.
[10] S. J. Johnson and S. R. Weller, “Regular low-density parity-check codes from combinatorial designs,” Proc. IEEE Information Theory Workshop, pages 90–92, Cairns, Australia, September 2001.
[11]S. Lin, D. J. Costello, Error Control Coding: Fundamentals and Applications, Prentice Hall, Englewood Cliffs, N.J., 1983.

