DOI: 10.21917/ijct.2019.0298

A REVIEW ON CONSTRUCTION METHODS FOR REGULAR AND NON-QUASI CYCLIC LDPC CODES

S. Vengatesh Kumar[1] and R. Dhanasekaran[2]

[1]Department of Electronics and Communication Engineering, Mohamed Sathak Engineering College, India

[2]Department of Electrical and Electronics Engineering, Syed Ammal Engineering College, India

Abstract

Low Density Parity Check (LDPC) codes are one of the most powerful error correction codes available today. Its Shannon capability that closely matches performance and lower decoding complexity has made them the best choice for many wired and wireless applications. This Paper provides an overview of the LDPC codes and compares the Gallager method, the Reed-Solomon-based algebraic method, and the combinatorial Progressive Growth (PEG) method for constructing regular LDPC codes and also Overlapped and Modified overlapped message passing algorithm for Non-Quasi Cyclic(NQC) LDPC codes.

Keywords:

Low-Density Parity-Check (LDPC) Codes, Reed-Solomon (RS) Codes, SPA, Tanner Graph, Progressive Edge Growth (PEG), Message Passing, Non-Quasi Cyclic (NQC)

1. INTRODUCTION

Low-Density parity-check (LDPC) codes were discovered by Gallager in early 1960s [1]. After being overlooked for almost 35 years, this class of codes were recently rediscovered by Mackay and Neal and Wiberg [8] [14] and shown to form a class of Shannon limit approaching codes [2], [6]-[8]. This class of codes decoded with iterative decoding, such as the sum-product algorithm (SPA) [1, 9], performs amazingly well for a lot of different channels. Since their rediscovery, LDPC codes have become a focal point of research for a variety of applications such as distributed source coding [10] and Forward error correction (FEC) [5].

The paper is organized as follows: section 2 introduces the necessary concepts about LDPC codes and their representation. Section 3 describes the pseudorandom construction method proposed by Gallager [1] [2]. We summarize the construction methods based on the Reed-Solomon (RS) Codes [4] and the Progressive Edge Growth (PEG) Algorithm in section 4, section 5 and section 6, respectively. Finally, section 7 concludes this paper.

2. OVERVIEW

The LDPC codes are a class of linear block codes. The name comes from the characteristic of their parity-check matrix which contains only a few 1's in comparison to the amount of 0's. Such a structure guarantees both: a lower decoding complexity and good distance properties [2]. We define two numbers describing these matrices: ρ for the number of 1's in each row and γ for the columns. For an $m\times n$ matrix to be called low-density the two conditions $\gamma \ll m$ and $\rho \ll n$ must be satisfied [3].

A parity-check matrix is said to be regular when γ is same for all the columns and ρ is constant for all the rows. If an LDPC code is described by a regular parity-check matrix, it is called a (γ,ρ)-regular LDPC code otherwise it is an irregular LDPC code [12]. Generally, there are two different methods to represent LDPC codes. Like all linear block codes, they can be described via matrices. The second method is a graphical representation [13].

2.1 MATRIX REPRESENTATION

Let's look at an example for a regular LDPC code. The matrix defined in Eq.(1) is a 4×8 parity check matrix for the (2, 4) regular code. This matrix cannot really be called low-density, since the size of H should be large enough for the condition given above to be satisfied.

$$H=\begin{bmatrix}0&1&0&1&1&0&0&1\\1&1&1&0&0&1&0&0\\0&0&1&0&0&1&1&1\\1&0&0&1&1&0&1&0\end{bmatrix}$$

2.2 GRAPHICAL REPRESENTATION

Tanner considered LDPC codes and showed how they may be represented effectively by a so-called bipartite graph, also known as Tanner graph [2].

It provides a complete representation of the code and it aids in the description of the decoding algorithm [9]. The tanner graph for Eq.(1) is given in Fig.1.

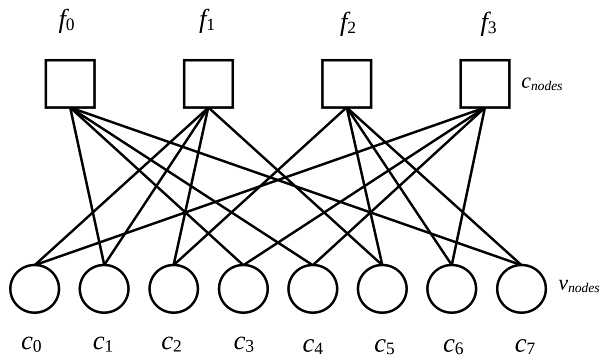


Fig.1. Tanner graph corresponding to the parity check matrix in matrix Eq.(1).The marked path c_2-f_1-c_5-f_2-c_2 is an example for a short cycle of length 4

A bipartite or tanner graph consists of two types of nodes which may be connected by edges. The two types of nodes are 'variable' nodes and 'check' nodes [16]. The Tanner graph of a code is drawn according to the following rule: check node j is connected to variable node i whenever element h_{ji} in H is '1'. There are $m = (n–k)$ check nodes, one for each check equation and n variable nodes, one for each code bit c_i, where n is the block length and k denotes the number of information bits. The m rows of H specify the m c-node connections and the n columns of H specify n v-nodes.

A cycle in a Tanner graph is a sequence of connected vertices which start and end at the same vertex in the graph, and which contains other vertices no more than once [15]. The length of a cycle is the number of edges it contains, and the girth of a graph is the size of its smallest cycle. For optimum decoding performance the Tanner graph should free of short cycles of length 4 [2].

3. GALLAGER'S CONSTRUCTION TECHNIQUE

For a given choice of ρ and y, Gallager [1] [2] gave the following construction method for a class of linear codes specified by their parity-check matrices. Form a $k_y \times k_\rho$ matrix H that consists of $y(k_y \times k_\rho)$ submatrices, $H_1, H_2, \ldots, H_y$. Each row of a sub matrix has ρ 1's and each column of a sub matrix contains a single 1 [17]. Thus, each sub matrix has a total of k_ρ 1's. For $1 \le i \le k$, the i^{th} row of H_1 contains all its ρ 1's in columns $(i\text{-}1)\rho+1$ to i_ρ. The other sub matrices are merely column permutations of H_1.

Random permutations of columns of H_1 to form the other sub matrices result in a class of LDPC codes with the properties given in section 2. There is no known method for finding these permutations to guarantee that no short cycles (especially of length 4) exist in the resultant code. Computer searches [2] are required to find good permutations and hence good LDPC codes. From this construction, it is clear that (1) no two rows in a sub matrix of H have any 1-component in common; and (2) no two columns of sub matrix of H have more than one 1 in common. The density of H is $1/k$. For H to be sparse, k is chosen much greater than 1.

For Example, given the regular (Gallager) LDPC code parameters n=20, k=5, q=4 and y=3, the resultant H is given by the following [3],

$$
\begin{array}{cccccccccccccccccccc}
\hline
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1\\
\hline
\end{array}
$$

The feature of LDPC codes to perform near the Shannon limit of a channel exists mostly for large block lengths. For example, there have been simulations that perform within 0.0045dB of the Shannon limit at a bit error rate of 10^{-6} with a block length of 10^7 [7]. The large block length results in large parity-check and generator matrices. The complexity of multiplying a code-word with a matrix depends on the amount of 1's in the matrix. If we put the sparse matrix H in the systematic form $[P^T I]$ then the generator matrix G can be calculated the Gauss Elimination method [3] as $G=[IP]$.

The sub-matrix P is generally not sparse so the encoding complexity will be quite high. Since the complexity grows in $O(n^2)$ even sparse matrices does not result in a good performance if the block length gets very high.

3.1 RS-BASED REGULAR-LDPC CODES

Djurdjevic et al. [4] proposed an algebraic method for constructing regular LDPC codes is presented. This construction method is based on the simple structure of Reed–Solomon (RS) codes with two information symbols. It guarantees that the Tanner graphs [4] of constructed LDPC codes are free of cycles of length 4 and hence have girth at least 6. The construction results in a class of LDPC codes in Gallager's original form [1]. These codes are simple in structure and have good minimum distances. They perform well with iterative decoding or SPA. Such parity check matrix can be masked to generate new and better LDPC codes [2].

3.2 RS CODES WITH TWO INFORMATION SYMBOLS

Consider the Galois field GF (q) with q elements, where q is a positive integer power of a prime number. Let q be a positive integer such that $2 \le q < q$. The generator polynomial of cyclic (n, k,d_{min}) RS code C is given by [2]:

$$g(X) = (X\text{-}\alpha)(X\text{-}\alpha^2),\ldots,(X\text{-}\alpha_{q\text{-}2}) = g_0 + g_1 X +,\ldots,+X_{q-2}$$

Notice that $n = q\text{-}1$, $k = q\text{-}q+1$, $g_i \in \mathrm{GF}(q)$ and α is a primitive element of a field. The parity check matrix RK for a Reed-Solomon code has size $(q\text{-}2)\times n$. The rank of matrix RK can be utmost $(q\text{-}2)$. Thus minimum distance is $d_{min} = (q\text{-}1)$.

Now consider the $(q\text{-}1)$ tuple vector

$$g^{(O)} = (g_0, g_1, \ldots, g_{q-2}, 0, 0, \ldots, 0)$$

Note that $g_{q-2} = 1$. By cyclically shifting $g^{(O)}$, we get generator matrix G of size $k \times \underline{n}$ for code C.

$$
G = \begin{bmatrix}
g_0 & g_1 & g_2 & . & . & 1 & 0 & . & . & 0 \\
0 & g_0 & g_1 & g_2 & . & . & 1 & 0 & . & . \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & . & . & g_0 & g_1 & g_2 & . & . & . & 1
\end{bmatrix}
$$

C is shortened by deleting the first $q\text{-}q+1$ information symbols from each code word of C [2]. The generator matrix for shortened RS code C_b is a sub matrix of size $2\times q$ and it is shown below:

$$
G_b = \begin{bmatrix}
g_0 & g_1 & g_2 & . & . & . & 1 & 0 \\
0 & g_0 & g_1 & g_2 & . & . & . & 1
\end{bmatrix}
$$

4. PROPERTIES OF SHORTENED CODE C_b

1. Since the length of code words in C_b is q and the minimum distance between two code words of C_b is $(q\text{-}1)$, two code words in C_b only agree at most at one location.

2. Let c be a code word of weight q. If we multiply c by $\forall p \in GF(q)$, we get set $C_b^{(1)}$ of (q-1) code words of weight q. Now, length of every $C_b^{(1)}$ is also q. So $C_b^{(1)}$ is a MDS (Maximum Distance Separable) code.
3. Let us partition C_b into a q co-sets $C_b^{(1)}, C_b^{(2)},\ldots, C_b^{(q)}$ based on $C_b^{(1)}$. Notice that $C_b^{(i)}$ is a MDS code. Therefore two code words in any co-set $C_b^{(i)}$ must differ in all the locations.

5. CONSTRUCTION OF LDPC CODE CHECK MATRIX

Let us now explain the explicit construction procedure for a LDPC code check matrix H.

1. All q elements of GF(q) can be expressed as some power of a primitive element α. Let us define the location vector of α_i as a q-tuple over GF(2) is given by:

   $Z(\alpha^i) = (0,0,\ldots,1,0,\ldots,0)$, where i^{th} element of $Z(\alpha^i)$ is 1 and all other elements are 0.

   Choose one code word $b = (b_1, b_2,\ldots, b_q) \in C_b(i)$. If we replace each $b_i(1\leq i\leq q)$ by its location vector $Z(b_i)$, we get $Z(b) = (Z(b_1), Z(b_2),\ldots, Z(b_q))$, which is a q q-tuple of weight q over GF(2).
2. Arrange all q q-tuple of $C_b^{(i)}$ as rows of a matrix and call this matrix as A_i. The weight of each column of A_i is 1.
3. Choose a positive integer y, such that $1\leq y\leq q$. Then the parity check matrix H of size $y_q\times q_q$ is defined as:

$$H \triangleq \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_y \end{bmatrix}$$

Since each column of A_i has weight 1, weight of an each column of H is y. So, H is a (y,q)-regular matrix. Each row in A_i is a co set member, so each row in A_i is different.

Two rows in A_i do not have single common element and two code words in C_b agree at most one symbol location ($d_{min} = q$-1). Hence it can be said that no two rows from A_i, A_j, G_j agree at more than a single element. This will imply that the Tanner graph corresponding to H is free of length 4.

The Gallager-LDPC code constructed is a (6,48)-regular (12288,10845) code with rate 0.8825 and minimum distance at least 8. The error performance is shown in Table.1. At the BER of the code performs 1.1dB from the Shannon limit. For comparison, the performance of the MacKay's computer generated code of the same length and rate is also given in Table.1 We see that in this case MacKay's code is 0.2 dB better than RS-based Gallager-LDPC code. However, the error performance of the RS-based Gallager-LDPC code has larger dropping rate. The performance curves of the two codes may cross each other at lower BER.

For Example: In Table.1, error performance with iterative decoding of (6,32)-regular LDPC code over GF(26) using the SPA is given. This (6,32) regular (2048,1723) RS-LDPC code has been adopted as the FEC in the IEEE 802.3an 10GBase-T standard [5].

Table.1. Error performance of the (2048, 1723) RS-based Gallager (6,32)-regular LDPC code with construction field GF(26)

E_b/N_0 (dB)	Uncoded BPSK	RS FER	RS BER	MacKay FER	MacKay BER	Shannon Limit
0	$10^{-1.1}$	-	-	-	-	-
1	$10^{-1.4}$	-	-	-	-	-
2	$10^{-1.6}$	-	-	-	-	10^{-3}
3	$10^{-1.8}$	$10^{-0.45}$	$10^{-0.449}$	$10^{-1.84}$	$10^{-1.84}$	-
4	$10^{-1.9}$	10^{-4}	$10^{-3.98}$	$10^{-7.78}$	$10^{-7.78}$	-
5	$10^{-2.4}$	-	-	-	-	-
6	$10^{-2.7}$	-	-	-	-	-
7	$10^{-3.3}$	-	-	-	-	-
8	$10^{-3.8}$	-	-	-	-	-
9	$10^{-4.6}$	-	-	-	-	-

6. PROGRESSIVE EDGE GROWTH ALGORITHM

A bipartite graph can be described using variable nodes, check nodes and set of edges E. The Progressive Edge Growth (PEG) algorithm proposed by Hu et al. [10] is a general method for the construction of finite length regular and irregular Tanner graphs having large girth by establishing-edges or connections between the variable and check nodes in an edge-by-edge or progressive manner. For given variable node, an edge connects it to one of the check node such that girth is maximum. Thus, PEG algorithm yields large girth when compared to codes constructed using random methods [8]. Hence, code constructed using PEG algorithm has low error floor in comparison with code constructed using random methods [11].

The PEG algorithm generates good codes for any given block length and rate, provided a density-evolution optimized degree sequence is supplied. Its low complexity makes it suitable for constructing codes of very large lengths and, with a slight modification to avoid the Gauss Elimination step, they can generate linear- time-encoding LDPC codes The Gallager's construction, does not have this degree of flexibility [2] [11].

Given the graph parameters, i.e. the number of variable nodes n, the number of check nodes m=n-k, and the symbol/variable node degree sequence D_v, an edge selection procedure is started such that the placement of a new edge on the graph has very less impact on the girth.

The variable node degree sequence can be described as follow:

$$D_v = \{d_{v1}, d_{v2}, d_{v3},\ldots, d_{vn}\}$$

where d_{vi} represents the degree of i^{th} variable node.

Input: sequence D_v

Output: parity check matrix H

Initialize all check nodes with degree 0

for i = 1 to n do

  for j = 1 to d_{vi} do

    if j = 1 then

      Find minimum degree check nodes set

$C = \{c_1, c_2, c_3,\ldots, c_m\}$ $1 \le m \le n\text{-}k$

s.t. $\deg(c_1) \le \deg(c_2) \le \ldots \le \deg(c_m)$

Choose check node $c_1 \in C$

Put an edge between i^{th} variable node and check node c_1

Increase the degree of c_1 by 1

else

For i^{th} variable node find check nodes set

$C = \{c_1, c_2, c_3,\ldots, c_m\}$ $1 \le m \le n\text{-}k$

s.t. girth is maximum and $\deg(c_1) \le \deg(c_2) \le \ldots \le \deg(c_m)$

Put an edge between i^{th} variable node and check node c_1

Increase the degree of c_1 by 1

end if

end for

end for

The check node degree distribution obtained using the above algorithm is almost uniform. Whenever multiple choices are available to pick check node from set C, we can either pick first check node in the set C or randomly pick any check node from set C. In algorithm, we always choose the first member of set C.

The Fig.3 is an example of symbol/variable node degree D_v = {2,2,2,2,3,3,3,3} [11]. The dashed lines represents an edges incident on variable of degree 2 and dark lines represents edges corresponding to variable nodes of degree 3.

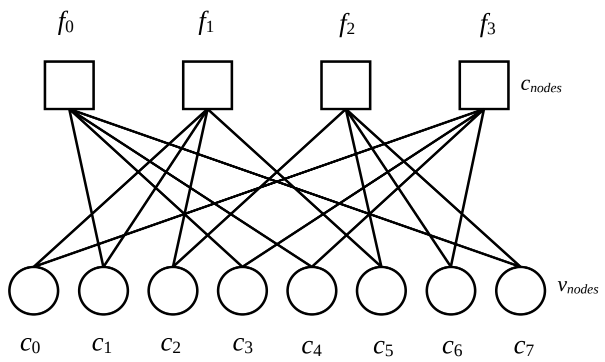


Fig.2. Tanner graph corresponding to D_v = {2, 2, 2, 2, 3, 3, 3, 3}

7. OMP TECHNIQUE AND THE MODIFIED OMP ALGORITHM

7.1 OVERLAPPED MESSAGE PASSING SCHEDULE

To adopt the OMP algorithm for NQC semi-random LDPC codes, let us first review this technique. As is well known, the BP algorithm consists of two decoding processes: the VNP and the CNP. In general, these two processes may not overlap because they offer updated data to each other. However, studies have found that the effect of this data dependency could be reduced if the row and column operations followed proper sequences [11]-[13]. If these operation sequences are taken as a kind of matrix permutation, the schedule finds a permutation that could transform the square sub-matrix into an H-matrix or the H-matrix into a standard matrix (refer Table.1), in which the bottom-left and top-right corners are the zero regions. Moreover, the VNP and CNP could be completely overlapped if the H-matrix is reconstructed in a specific mathematical pattern.

However, these proposed methods are meaningless when applied to NQC semi-random LDPC codes. As the $0_{i;j}$ parameters are random in each sub-matrix, there are several problems, as follows:

First, a compatible permutation may be hard to describe mathematically, so ROM is needed to store the permutation for implementation, which would increase hardware complexity.

Second, even if a permutation matrix could be found, the OMP technique may not result in any significant hardware improvement because the zero region may be too small for the VNP and CNP to overlap to any great extent.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		1		1	1				1		1	1		
2			1				1	1	1		1		1	
3	1	1				1	1						1	1
4	1		1			1			1	1		1		
5	1	1			1					1		1		1
6			1	1			1	1					1	1
7				1	1	1		1		1	1			

	8	3	4	9	5	7	10	13	11	6	14	1	12	2
7	1		1		1		1		1	1				
2	1	1		1		1		1	1					
6	1	1	1			1		1			1			
4		1		1			1			1		1	1	
1			1	1	1				1				1	1
5					1		1				1	1	1	1
3						1		1		1	1	1		1

Zero region

Fig.3. OMP technique for a parity-check matrix H: matrix H (top) and permuted standard matrix (bottom).

7.2 OPTIMAL BELIEF PROPAGATION (OBP)-PROGRESSIVE EDGE GROWTH (PEG) CONSTRUCTION METHOD

7.2.1 System Model and LBP Decoding Algorithm:

A (n,k) LDPC code is described by a $m \times n$ parity check matrix H or the Tanner graph [1], where n denotes the number of variable nodes; m denotes the number of check nodes; k denotes the original information bits ($k=n-m$). The check nodes in the Tanner graph correspond to the parity check functions, and the variable nodes correspond to the coded bits including information bits and parity bits. $\{c_i\}(1 \le i \le m)$ denotes the check nodes and $\{v_j\}(1 \le j \le n)$ denotes the variable nodes. The posteriori probability (APP) message of v_j is denoted by Q_j. r_{ij} denotes the check-to-variable (CTV) extrinsic message from c_i to v_j. q_{ji} denotes the variable-to-check (VTC) extrinsic message from v_j to c_i. In this paper, all the messages are in the shape of the Log Likelihood Ratio (LLR) denoted by $L[\bullet]$.

In efficient LBP algorithms, all the extrinsic messages are initialized to zeros, i.e., $L[r_{ij}] = 0$ and $L[q_{ji}] = 0$. Afterward, the CNP are operated node-by-node until all the check functions are satisfied or the number of iterations reaches the pre-given limit. CNP is a serial of operations that update the values of $L[r_{ij}]$ and $L[Q_j]$ for the v_j in $N(i)$, where $N(i)$ is the set of variable nodes neighboring c_i. The decoded result is decided to be zero and one when $L[Q_j] > 0$ and $L[Q_j] \leq 0$, respectively.

7.3 BELIEF PROROGATION CONDITION (BPC)

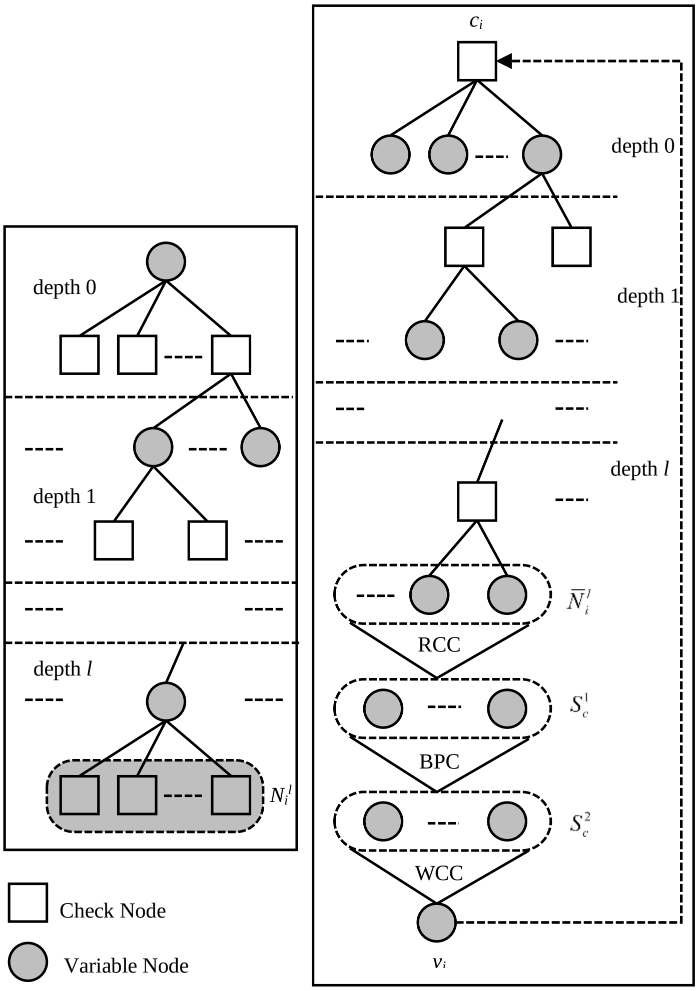
Based on the regulations of LBP algorithms we can approximately have that

$$E(L[q_{ji}]) = E(L[Q'_j]) - E(L[r'_{ji}]) \quad (1)$$

$$1\text{-}\varphi\, E(L[r_{ij}]) = \prod_{k \in N(i)\setminus j} [1 - \varphi E\left(L\left[q_{ki}\right]\right) \quad (2)$$

$$E(L[Q_j]) = E(L[q_{ji}]) + E(L[r_{ij}]) \quad (3)$$

The proposed OBP-PEG method not only select the variable nodes having the small-value degree ratio γ_j (as in Eq.(6)), but also enhances the belief propagation of the variable nodes having minimum $E(L[Q_j])$.



(a) PEG Method (b) OBP – PEG Method

Fig.4. Comparison of the procedure of PEG method and the proposed OBP-PEG method, where the latter is illustrated in QC-LDPC applications

7.4 MODIFIED OVERLAPPED MESSAGE PASSING ALGORITHM

To solve the above problems, we propose a modified OMP algorithm. We suppose that old message data will be used for decoding when the VNP and CNP overlap. C_j denotes the Log-Likelihood Ratio (LLR) channel information of the j^{th} variable node. $N.v_k$ denotes a set of check nodes connected to the k^{th} variable nodes, while $N.c_k$ denotes a set of variable nodes connected to the k^{th} check nodes. P denotes a parallelism parameter, and $PD2m$, $m2N$, $P<L$ denotes the check-to-variable message whereas the variable-to-check message. The modified OMP algorithm is shown in Algorithm.

In this algorithm, compute with the same formulation as the offset min-sum algorithm or the BP algorithm, but with a different message-passing schedule. The message passing of each square sub-matrix can be divided into three regions (Fig.3). In the first region, the message passing is the same as that of the BP algorithm. In the second region, check-to-variable messages updated in the n^{th} iteration are used to calculate the variable-to check messages in the n^{th} iteration, whereas variable to-check messages updated in the n^{1th} iteration are used to calculate check-to-variable messages in the n^{th} iteration.

In the third region, variable-to-check messages in the n^{th} iteration are calculated using the check-to-variable messages updated in the n^{2th} iteration, whereas check-to-variable messages in the n^{th} iteration are calculated using the variable-to-check messages updated in the $n^{C_1 th}$ iteration. Compared to the OMP technique, the modified OMP algorithm completely overlaps the VNP and CNP without introducing any constraints on code construction. Furthermore, the parameter P can adjust the parallelism of the single-core decoder architecture.

However, these message-passing tasks have different degrees of efficiency in their corresponding Tanner graphs. Moreover, when the variable-to-check messages are updated, old variable-to-check messages are used to calculate the check-to-variable messages, which implies that the variable-to-check and check-to-variable messages must be stored separately in simple two-port RAMs. In contrast to the BP vector partially parallel decoder, double memories are needed for intra-message storage.

8. CONCLUSIONS

Good regular LDPC codes with large block lengths constructed using Gallager's pseudorandom technique are largely computer generated. This leads to the following limitations:

- Do not ensure absence of short cycles.
- Due to lack structure, the encoding complexity is very high for large column weights and code lengths.

The above drawbacks are overcome using the algebraic method studied in section 4 which exploits the structural advantages of the Reed Solomon codes and results in a class of Gallager's LDPC codes having simple structure, good minimum distances and a girth at least 6. So they work well with the SPA decoding. Furthermore, because of the cyclic nature of the RS-LDPC code their encoding is simple and can be implemented using linear shift registers.

The PEG algorithm also avoids the occurrence of short cycles by providing larger girths even better than RS-based codes. It can be easily tailored to construct LDPC codes having triangular structure which makes them linear-time encodable. Moreover computation and storage requirements in the encoder are also reduced because of sparsity of the parity check matrix.

Thus compared with Gallager's explicit construction, the RS-LDPC and PEG construction in general achieves a better girth and minimum distance properties with much less complexity. However, the PEG algorithm can also be applied to generate irregular graphs whereas the Gallager's and the RS-based construction only apply to regular codes.

REFERENCES

[1] R. Gallager, "Low-Density Parity-Check Codes", *IRE Transactions on Information Theory*, Vol. 8, No. 1, pp. 21-28, 1962.

[2] Shu Lin and Costello, "*Error Control Coding*", Pearson Prentice Hall, 2004.

[3] Othman O. Khalifa, Sheroz khan, Mohamad Zaid and Muhamad Nawawi, "Performance Evaluation of Low Density Parity Check Codes", *International Journal of Computer Science and Engineering*, Vol. 1, No. 4, pp. 67-70, 2013.

[4] Ivana Djurdjevic, Jun Xu, Khaled Abdel-Ghaffar and Shu Lin, "A Class of Low-Density Parity-Check Codes Constructed Based on Reed-Solomon Codes with Two Information Symbols", *IEEE Communication Letters*, Vol. 7, No. 7, pp. 317-319, 2003.

[5] C.E. Shannon, "A Mathematical Theory of Communication", *The Bell System Technical Journal*, Vol. 27, No. 3, pp. 623-656, 1948.

[6] S.Y. Chung, G. D. Forney,T.J. Richardson and R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit", *IEEE Communication Letters.*, Vol. 5, No. 3, pp. 58-60, 2001.

[7] J. C. MacKay, "Good Error-Correcting Codes Based on very Sparse Matrices", *IEEE Transactions on Information Theory*, Vol. 45, No. 2, pp. 399-432, 1999.

[8] T. Richardson and R. Urbanke, "*Modern Coding Theory*", Cambridge Press, 2003.

[9] Angelos D. Liveris, Zixiang Xiong and Costas N. Georghiades, "Compression of Binary Sources with Side Information at the Decoder Using LDPC Codes", *IEEE Communications Letters*, Vol. 6, No. 10, pp. 440-442, 2002.

[10] Xiao Yu Hu, Evangelos Eleftheriou and Dieter M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs", *IEEE Transactions on Information Theory*, Vol. 51, No. 1, pp. 386-398, 2005.

[11] R.C. Baumann, "Radiation-Induced Soft Errors in Advanced Semiconductor Technologies", *IEEE Transactions on Device and Materials Reliability*, Vol. 5, No. 3, pp. 301-316, 2005.

[12] M.A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A.F.Witulski, J. Sondeen, S.D. Stansberry, J. Draper, L.W. Massengill and J.N. Damoulakis, "Models and Algorithmic Limits for an ECC-Based Approach to Hardening Sub-100-nm SRAMs", IEEE *Transactions on Nuclear Science*, Vol. 54, No. 4, pp. 935-945, 2007.

[13] H. Naeimi and A. Dehon, "Fault-Secure Encoder and Decoder for Memory Applications", *Proceedings of IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems*, pp. 409-417, 2007.

[14] T.Y. Chen, K. Vakilinia, D. Divsalar and R.D. Wesel, "Protograph based Raptor-like LDPC Codes", *IEEE Transactions on Communications*, Vol. 63, No. 5, pp. 1522-1532, 2015.

[15] S. Myung, S.I. Park, K.J. Kim, J.Y. Lee, S. Kwon and J. Kim, "Offset and Normalized Min-Sum Algorithms for ATSC 3.0 LDPC Decoder", *IEEE Transactions on Broadcasting*, Vol. 63, No. 4, pp. 734-739, 2017.

[16] Y. Xu, L. Szczecinski, B. Rong, F. Labeau, D. He, Y. Wu and W. Zhang, "Variable LLR Scaling in Min-Sum decoding for Irregular LDPC Codes", *IEEE Transactions on Broadcasting*, Vol. 60, No. 4, pp. 606-613, 2014.

[17] F. Alberge, "Min-Sum Decoding of Irregular LDPC Codes with Adaptive Scaling based on Mutual Information", *Proceedings of IEEE International Symposium on Turbo Codes and Iterative Information Processing*, pp. 71-75, 2016.