Journal of Theoretical and Applied Information Technology

www.jatit.org

NOVEL CONSTRUCTION OF SHORT LENGTH LDPC CODES FOR SIMPLE DECODING

Fatma A. Newagy, Yasmine A. Fahmy, and Magdi M. S. El-Soudani
Department of Electronics and Communications, Faculty of Engineering, Cairo University, Giza, Egypt.
Email: fnewagy@ieee.org

ABSTRACT

This paper introduces different construction techniques of parity-check matrix H for irregular Low-Density Parity-Check (LDPC) codes. The first one is the proposed Accurate Random Construction Technique (ARCT) which is an improvement of the Random Construction Technique (RCT) to satisfy an accurate profile. The second technique, Speed Up Technique (SUT), improves the performance of irregular LDPC codes by growing H from proposed initial construction but not from empty matrix as usual. The third and fourth techniques are further improvements of the SUT that insure simpler decoding. In Double Speed Up Technique (DSUT), the decoder size of SUT matrix is fixed and the size of H is doubled. In Partitioned Speed Up Technique (PSUT), the H size is fixed and the decoder size decreases by using small size of SUT matrices to grow H. Simulations show that the performance of LDPC codes formed using SUT outperforms ARCT at block length N = 1000 with 0.342dB at BER = 10^{-5} and LDPC codes created by DSUT outperforms SUT with 0.194dB at BER = 10^{-5}. Simulations illustrate that the partitioning of H to small SUT submatrices not only simplifies the decoding process, it also simplifies the implementation and improves the performance. The improvement, in case of half, is 0.139dB at BER=10^{-5} however as partitioning increases the performance degrades. It is about 0.322dB at BER=10^{-5} in case of one-fourth.

Keywords: Low Density Parity-Check (LDPC) codes, Random Construction Technique (RCT), Accurate Random Construction Technique (ARCT), Speed Up Technique (SUT).

1. INTRODUCTION

Low-Density Parity-Check codes (LDPC) have been the subject of intense research lately because of their capacity-achieving performance and linear decoding complexity by using an iterative decoding algorithm, the so-called belief propagation or sum-product algorithm [1]. They were originally proposed in 1962 by Robert Gallager [2]. In the late 90's LDPC codes were rediscovered by Mackay and Neal [3-4] and also by Wiberg [5]. Current hardware speeds make them a very attractive option for wired and wireless systems. Gallager considered only regular LDPC, i.e., codes that are represented by a sparse parity-check matrix with a constant number of ones (weight) in each column and in each row. Later it was shown that the performance of LDPC codes can be improved by using irregular LDPC codes, i.e., both nonuniform weight per column and nonuniform weight per row [6-7].

We can define an irregular Gallager code in two steps. First, we select a profile that describes the desired number of columns of each weight and the desired number of rows of each weight. Second, we construct the parity-check matrix that achieves the given profile.

The parity-check matrix of a code can be viewed as defining a bipartite graph [8] with "variable" vertices corresponding to the columns and "check" vertices corresponding to the rows. Each non-zero entry in the matrix corresponds to an edge connecting a variable to a check.

Most of the current techniques in the design of irregular codes are based on random sampling from appropriate degree distribution ensemble along with some simple constrains such as the avoidance of length-4 cycles. It is shown in [8] that in the limit of large block length, the neighborhood of a node in LDPC code tends to be tree-like and hence the belief propagation algorithm is exact. It is also shown that the performance of a randomly chosen code is very good with high probability. However at short block lengths, there is considerable variation among codes from a given ensemble and the neighborhood tends to be inevitably non-tree-like. From practical perspective it is desirable to have an algorithm that is able to efficiently find good codes from a degree distribution at a particular block length.

This paper is organized as following, definitions and notations are given in section 2. In section 3, we present the Random Construction Technique (RCT) and the proposed Accurate Random Construction Technique (ARCT). The Speed Up Technique (SUT) is introduced in section 4. Two other Simple decoding techniques based on SUT are described in section 5. In section 6, simulation results are presented. Finally, section 7 concludes the paper.

2. DEFINITIONS

An M×N LDPC code parity-check matrix H is usually represented by a Tanner graph *g*. Let *g*={(*V*,*E*)} be a graph, where *V* is a set of vertices or nodes V and *E* is the set of edges E connecting the vertices. The degree of node V is the number of edges incident on V.

Definition 1: (Cycle) A cycle of length 2d in Tanner graph representation of a code, is a set of d variable nodes and d check nodes connected by edges such that a path exists that travels through every node in the set and connects each node to itself without traversing an edge twice.

Definition 2: (Girth) The girth of a graph is the length of its shortest cycle.

The graph *g* is bipartite if the set of vertices *V* can be decomposed into two disjoint sets *V1* and *V2* such that no two vertices within either *V1* or *V2* are connected by an edge. It is well known from graph theory that a graph with at least two nodes is bipartite if and only if all its cycles are even length.

The relation between the parity check matrix and its Tanner graph is illustrated in Figure 1. the figure shows a 5×10 matrix, one column in the parity-check matrix corresponds to one variable node (v_1 , v_2 , ... , v_{10}) in Tanner graph, and one row corresponds to one check node (c_1 , c_2 , c_3,, c_4). The bold solid lines (c_4, v_2), (v_2, c_5), (c_5, v_9), and (v_9, c_4) depict a cycle in the Tanner graph; this turns out to be the shortest cycle in this graph, so that its girth equals 4.

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

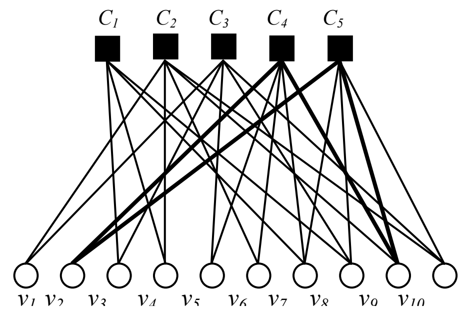


Figure 1 parity-check matrix and Tanner graph of an irregular LDPC code

In irregular LDPC codes, profile which specifies the degrees of the vertices is often presented in polynomial form [9]. The variable node degree distribution is denoted by λ(*x*) and it can be expressed as:

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \qquad (1)$$

Where λ_i is the fraction of edges emanating from variable nodes of degree *i* and d_v is the maximum variable degree of the irregular LDPC codes. Similarly, the check node degree distribution denoted by $\rho(x)$ and can be expressed as:

$$\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1} \qquad (2)$$

Where ρ_i is the fraction of edges emanating from check nodes of degree *i* and d_c is the maximum variable degree of the irregular LDPC codes. The given degree distributions of (λ, ρ) in [9] is used in this paper.

3. CONSTRUCTION TECHNIQUES

The construction technique objective is to construct the parity-check matrix H of irregular LDPC code from a given profile for a block length

N and M=N(1-R), where R is the rate $R = 1 - \frac{\sum_i \rho_i / i}{\sum_j \lambda_j / j}$.

3.1 RANDOM CONSTRUCTION TECHNIQUE (RCT)

Random construction technique (RCT) for a given profile by adding columns can be described as follow:

1- Start with a sparse empty matrix with size M×N.
2- According to a given degree distribution of (λ, ρ), calculate the number of ones in each row (from 1 to M) and in each column (from 1 to N).
3- For each column, select a random row and assign ones to that (row, column).
4- Check
   A. If the degree constraint for the corresponding row is violated.
   B. If any cycles of length four will be formed.

If any of the two constrains is violated, select another random row and check again.

5- Repeat the selection and check till the number of ones of that column equals the required number.

The above technique leads to profile as near as possible to the desired; the column weights is as desired, but the row weights can not be controlled in short block lengths.

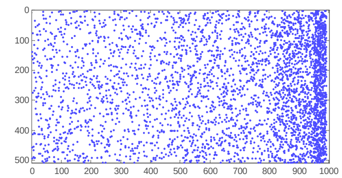
3.2 ACCURATE RANDOM CONSTRUCTION TECHNIQUE (ARCT)

The ARCT is proposed to force the matrix to have specific row weights, the technique can be improved as follow: construct a vector *u* of size *e* (number of ones in H matrix), that acts as the supply. The elements of *u* contain the row number where the corresponding one can be placed. Instead of selecting a row with the random generator directly, we select an element of *u* randomly after placing the one in *H*, we delete the selected element from the vector *u* and then the length of *u* is *e*-1 and so on till the *u* vector is empty. This method guarantees the desired profile. For example, in the case of maximum variable node degree distribution equals to 20, the degree distribution pair (profile) [9] is as follow:

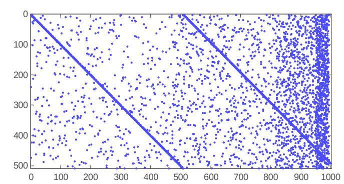
$$\lambda(x) = 0.2415x + 0.2663x^2 + 0.0514x^5 + 0.1849x^6 + 0.2559x^{19}$$

$$\rho(x) = 0.3392x^6 + 0.6388x^7 + 0.0220x^8$$

The parity check matrix H using ARCT for block length N=1000 is shown in figure 2-a



a)



b)

Figure 2 H using a) ARCT b) SUT

4. SPEED UP TECHNIQUE (SUT)

The proposed modification on ARCT is presented in the following:
For half rate as in the above degree distribution pair (profile) i.e. N=2M

1- Start with a sparse empty matrix with size M×N.
2- Split H to two submatrices [M×M | M×M]
3- Start with two eye matrices (ensure no cycles) with size M×M in H submatrices.
4- According to a given degree distribution of (λ, ρ), calculate the number of ones in each row and column.
5- Subtract one from the number of ones of each column and two from the number of ones of each row.
6- Calculate u vector.
7- For each column, select randomly a row from u and assign ones to that (row, column)

8- check:
   A. If the degree constraint for the corresponding row is violated.
   B. If any cycles of length four will be formed.

   If any of the two constrains is violated, select another random row from vector *u* and check again.
9- Continue till a row is found which together with that column satisfies the above two constrains.

The parity check matrix H for the above example using SUT for block length N=1000 is shown in figure 2-b.

As we will explain later, simulation shows that there is an improvement of LDPC code formed by SUT over ARCT. So, we will use SUT matrix in the reminder of this paper.

5. SIMPLE DECODING USING SUT

The decoding process and implementation are simplified by the following two proposed techniques

5-1 Double Speed Up Technique (DSUT)

In this technique the decoder size is fixed and the H size is increased to improve the performance as follow

1- Start with SUT matrix with size M×N as in Fig. 2-b.
2- Create an empty sparse large matrix H_L with size 2M×2N.
3- Construct H_L as follow

$$H_L = \begin{bmatrix} H & [0] \\ [0] & H \end{bmatrix} \qquad (3)$$

Where [0] is a zero matrix of size M×N.

This construction will use decoder of H size without any more complexity. An example for N=1000 is shown in figure 3

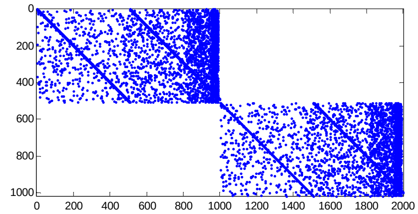


Figure 3 DSUT for N=1000

5-2 Partitioned Speed Up Technique (PSUT)

In this technique H size is fixed and the decoder size is decreased to simplify the implementation. This is accomplished as follow

1- Start with an empty sparse matrix with M×N.
2- Divide the parity-check matrix H to submatrices Hi each of size equals Mi×Ni, where $M_i = \frac{M}{i}$ And $N_i = \frac{N}{i}$ i is the degree of portioning.
3- Construct the submatrix Hi using SUT for the given degree distribution (λ, ρ).
4- Construct the parity-check matrix H as follow

$$H_{M\times N} = \begin{bmatrix} H_i & [0]_i & [0]_i \ldots\ldots\ldots\ldots [0]_i \\ [0]_i & H_i & [0]_i \ldots\ldots\ldots\ldots [0]_i \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ [0]_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots H_i \end{bmatrix}_{M\times N} \qquad (4)$$

where $[0]_i$ is a zero matrix of size Mi×Ni. Example for N=1000 and His partitioned to 2, 3, 4, and 10 is shown in figure 4.

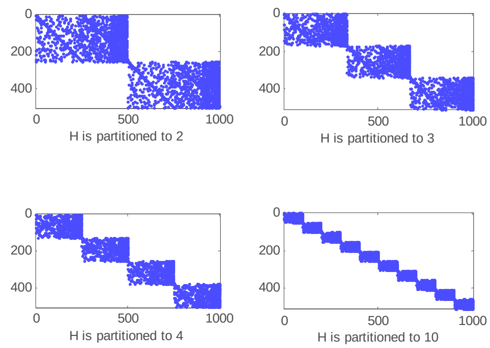


Figure 4 The partitioned parity-check matrix H i=2, 3, 4, 10 using SUT

6. SIMULATION RESULTS

Simulations were conducted to compare the performance of the two random construction algorithms, ARCT and SUT. The case of irregular LDPC code with rate 0.5 and block length N=1000 for the given degree distribution pair (profile) was considered. We always assume Binary-Input AWGN channel. The SUT has initially two eye matrices which satisfy no cycles (not only no cycle 4) and the adding ones in H satisfies the above mentioned constrains for good performance. The simulation shows that SUT outperforms ARCT with 0.342dB at BER = 10^{-5} as shown in figure 5.

The performance of LDPC codes formed by using the DSUT is shown in figure 6. There is an improvement of 0.19dB at BER = 10^{-5} using the same small decoder size.

The performance of the PSUT which simplify the decoding process and implementation is shown in figure 7. Simulation shows that partitioning H with i=2 does not only simplify the decoding process (where decoder for codeword length $\frac{N}{2}$ can be used), it also improves the performance by 0.139dB at BER=10^{-5}. For i=3, the performance is nearly the same as without partitioning but this is achieved with even simpler decoder (for codeword length $\frac{N}{3}$).

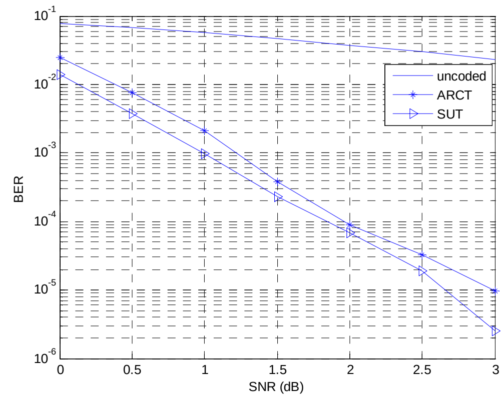


Figure 5 the performance of LDPC code formed by ARCT and SUT

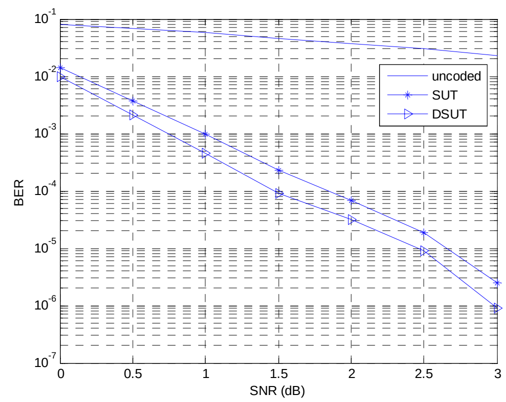


Figure 6 the same decoder for N=1000 and N=2000

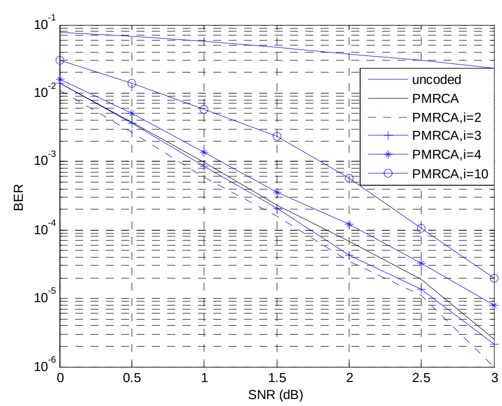


Figure 7 the performance of partitioned parity-check matrix i=2, 3, 4, 10 using SUT

To conclude, as the size of the submatrices in H gets smaller the performance degrades as the performance of each submatrix degrades and H matrix has more regular form. On the other side, the decoder is simpler and the degree of simplicity is the degree of partitioning. For further partitioning, e.g., i= 4, or 10, the performance degrades by 0.322dB at BER=10^{-5} and 0.5dB at BER = $2*10^{-5}$ respectively.

7. CONCLUSIONS

Different construction techniques of parity-check matrix H for irregular Low-Density Parity-Check codes (LDPC) have been introduced. ARCT aims to modify the traditionally RCT to satisfy an accurate profile. Speed up technique SUT, growing H from proposed initial construction ensures no cycles not from empty matrix as usual, improves the performance of irregular LDPC at block length N = 1000 with 0.342dB at BER = 10^{-5}. Finally, two techniques having simple decoding process are proposed based on SUT. Double speed up technique (DSUT) is used to fix the decoder size of SUT matrix and double the size of H lengthening the block length. This technique improves the performance of DSUT over SUT by 0.194dB at BER = 10^{-5}. Partitioned speed up technique (PSUT), fix H size and decrease the decoder size by using small size of SUT matrices to grow H. This improves the performance in case of half by 0.139dB at BER=10^{-5} but as partitioning increases the performance degrades by 0.322dB at BER=10^{-5} in the case of one-fourth. This can be explained by the fact that the performance of the submatrices themselves degrades and H matrix has more regular form.

8. REFERENCES

[1] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and sum-product algorithm," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 498-519, Feb. 2001.

[2] R. G. Gallager, Low-Density Parity-Check Codes, Cambridge, MA:MIT Press, 1963.

[3] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," IEE Electron Lett., vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[4] D. J. C. Mackay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. Inform. Theory, vol. IT-45, no. 2, pp. 399–431, March 1999.

[5] N. Wiberg, "codes and decoding on general graphs," Linkoeping studies in science and technology, No. 440, 1996.

[6] T. J. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching low-density parity-check codes," IEEE Trans. Inform. Theory, vol. 47, pp. 619-637, Feb.2001.

[7] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in Proc. 30th Annu. ACM Symp. Theory of computing, 1998, pp. 249-258.

[8] F. R. Kschischang, "Codes defined of graphs," IEEE Commun. Mag. Vol. 41, no. 8, pp. 118-125, Aug. 2003.

[9] F. Newagy, Y. Fahmy, and M. El-Soudani, "Designing near Shannon limit LDPC codes using particle swarm optimization algorithm," in Proc. IEEE ICT-MICC, 14-17 May 2007, Penang, Malaysia.