International Journal of Engineering & Technology, 7 (1.3) (2018) 198-202

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Progressive edge growth LDPC Encoder with spiral search algorithm

Anand Anbalgan [1]*,Senthil Kumar.P [2],

[1]*Lecturer/Department of Electrical Engineering Institute of Technology,, University of Gondar, Ethiopia*
[2]*Professor/Department of Computer science Engineering, S.K.R College of Engineering, Chennai*
**Email: anandtce@gmail.com*

Abstract

Trapping set causes the drop of performance in error floor region. Identification of TS is done by graphical method and enumerators. The lowest odd degree (minimal) TS is increasing the formation of more unsaturated nodes in iterative decoding. Progressive edge growth (PEG) Low-density parity check code (LDPC) [2] avoidance of trapping sets are mainly based on the distance and degree calculation of successive CN. This simple tool is used to eliminate TS when the encoder ensemble designs itself. Non-zero neighborhood search also made an influence on error floor. The spiral search method is used for Non-zero codeword search (NZCW) search for the first time in this research, at the decoder part. So, Non-zero codeword spiral search (NZCSS) converge fast with less number iteration, and this reduces the iteration of the decoder.

Keywords: *PEG LDPC; Trapping set; Neighbourhoods search; Spiral search*

1. Introduction

LDPC encoder design the Trapping set (TS) analysis is the main function to determine the error floor and waterfall region [3] on LDPC [1]. In this research, TS elimination process is done in encoder design as well as decoder part. Many researches were eliminating the TS [4] by graph theory approach, which is tedious in the process. But this research is on eliminating the TS by neighborhood analysis and distance evaluation process. Finding of non-zero code neighborhood [5] is one of the important processes in the iterative decoder. This research work has innovatively developed an algorithm for non-zero codeword search by spiral manner for LDPC. This chapter explains the trapping set analysis and non-zero codeword search algorithms. The section 2 gives the idea of Trapping set analysis by line and node analysis and identification of lowest degree and section 3 provide how that Trapping set is identified in PEG. Section 4 list out different types of searching algorithm and section 5 describes that the proposing algorithm section 6 illustrates the results and discussed the performance and section 7 give the conclusion of the PEG with proposed search algorithms

2. Trapping set

A drop of error floor performance at high SNR level, caused by the harmful structure of LDPC is known as trapping set. Trapping set is the subset of tanner graph, which has unsaturated nodes. Especially, iterative decoding process fails to perform error floor, by trapping set in the structure. By simply improving the local girth of the ensemble will reduce the formation of the unsaturated subset in LPDC ensembles. Already PEG encoder and improved PEG encoder ensemble maintain large local girth in the construction of LDPC ensemble.

Trapping set is always independent of input message codeword, hence it will not take the recent codeword to next iteration. It follows always constant value to the check node irrespective of input codeword. It propagates error and the error is propagated to all check nodes and to the variable nodes. Due to fixed error propagation from one node to all nodes, after the iteration all the data are erroneous. So, the error rate increased for particular SNR of AWGN channel or crossover probability of BEC channel.Now the removal of dominant trapping set is a much focused research area. Elimination of all trapping set is not possible. But it can be eliminated almost lower. To remove the TS, it is mandatory to have some basic idea of trapping set. The following section elaborately explains the identification of trapping set and elimination procedure.

2.1 Identifying Trapping set

Initially, the tanner graph is converted as vertex and edges. These edges and vertex are shown in Figure 3.1. This graph is modified as a line graph. Check nodes are separated by the odd node and even node [6]

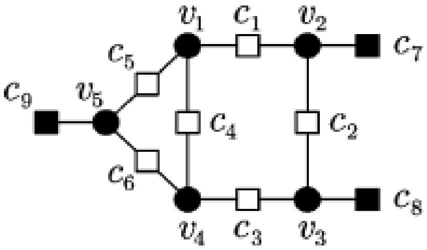

Fig.1 {5,3} edges and vertex representation of Tanner graph

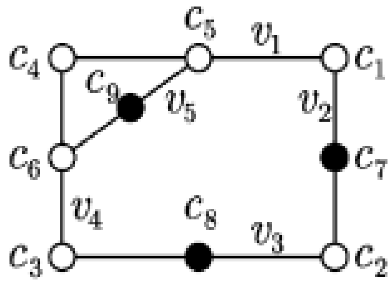

Fig.2 Line graph of {5,3}

Dark circles are referred to available nodes and Dark Squares represent odd-degree check nodes and the empty squares represent even degree check node. The line graph is decomposed as adding and deleting the line. This process of addition and deletion will lead to new subset graph. Decomposition of graph ends until the inseparable subsets is formed. In most of the cases, odd degree check nodes are leading to trapping set behaviors in the iterative decoding process. This dominant trapping set reduces the error floor performance of LDPC. By adding a line the following subset is identified as a graphical representation [7]

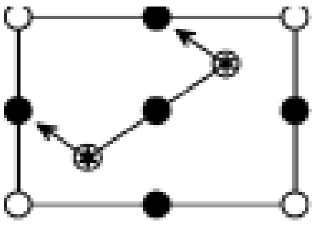

Fig.3 (a) (5,3){1} trapping set

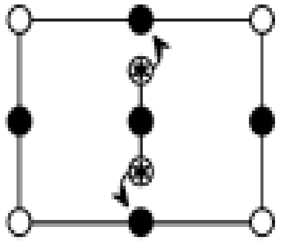

Fig. 3 (b) (5,3){2} trapping set

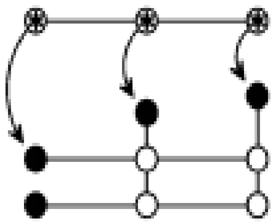

Fig.3.(c) (5,1) trapping set by adding line

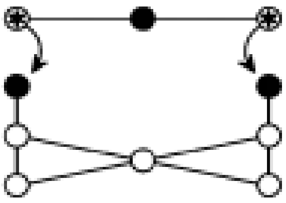

Fig3.(d) (5,1) Trapping set

So the trapping set identification and finding dominant trapping set by adding the line to the edges of check node with the lowest degree are shown. Figure (3.3a) shows that the degree of check node as two, and removing the lines from the subset graph will reduce the degrees as odd. Figure (3d) illustrates that adding the line to the lowest degree check node with a merged node which leads to trapping set. Trapping set elimination is categorized as follows:

- Modify the decoder to decode with the limit of cycle structure optimally
- Modify the LDPC parity check ensembles for the specific girth

2.2 Trapping set Enumerator

By using impulse algorithm on LDPC H matrix, the following modification is done to get trapping set enumerator

(a.b) enumerator calculation

a - Hamming weight of original VN

b - Hamming weight of Flag VN

Log-likelihood ratio (LLR) initialized for flag VN to get large positive value Maximum codeword length as a constraint of a+b for enumerating

Maximum ratio b/a returned by the algorithm Trapping set enumeration with high failure rate and small values of a and b, generally

$$b/a \ll 1 \tag{1}$$

Trapping set enumerator for LDPC code with the constraints is as,

$$a+b < 10 \tag{2}$$

$$b/a \leq 0.6 \tag{3}$$

The above enumerator is listed out the unsaturated set for the constraint of $a+b < |max|$ and $b/a \leq |min|$. Most of the enumerators listed out trapping set, but the max and min constraints are trial and error method. So, the complexity of the system is increased by choosing unoptimized constraint of a and b.

Trapping set control mechanism is one of the mandatory ways to get good error floor performance. In this research construction of PEG LDPC encoder is designed with the controlling of small trapping set. Mean while, the layered decoder part also has control of trapping set and elimination. Hence, the elimination of trapping set in encoder and decoder is one of the main reasons for an efficient algorithm for short length LDPC. The following simple procedure for eliminating the small trapping set in PEG LDPC encoder reduces the error floor at lowest level.

3. PEG LDPC Encoder ensemble design with avoidance of small trapping set

In this design, the elimination of trapping set is done by avoiding the connection between the CN and the progressive VN. If any findings of a harmful generating set of small size trapping set are there, then there is no connection between VN to CN[10]. Here, the elimination of TS is a very simple mechanism of distance between two CNs.

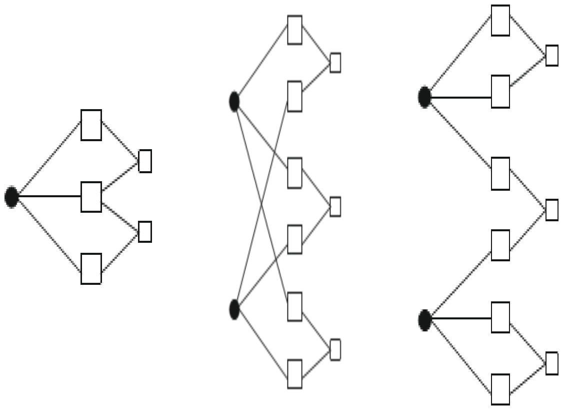

Figure 4 PEG with avoidance of trapping set

The condition of avoiding small trapping set in terms of the distance between two CNs is presented here. The distance between two CN with indices of i and j is:

$$d_c(i,j) = |i-j| \tag{4}$$

If both i and j are not larger than $n_{2+1,}$ and is $d_c(i,j) = \infty$ otherwise.

Figure 3(a) is the presentation of basic seed matrix and it is progressive as a depth l as 2 in Figure 3(b). The connection path is changed to a short distance variable node in Figure 3(c).

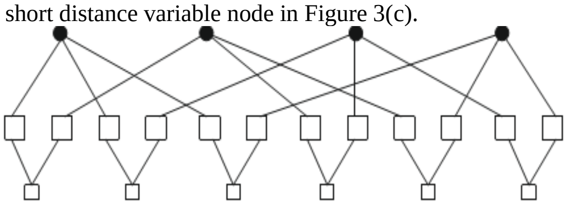

Figure 5 PEG avoidance of trapping set

Figure 5 shows the variable node degree as 2 and the check node degree as 3. If the VN degree as γ and CN degree as $\gamma + 1$ it will remove the minimal value to check nodes. Because the small odd node is check node which leads to the trapping set. Once the sub graph has been spread to depth l_{max} those CNs in complement of neighborhood exists as in chapter 2. PEG $\overline{N_{vj}^{l_{max}}}$whose connection to vj might generate a stopping set (S_t) of size is smaller than the subset formed by the PEG algorithm [8]

$$\sum_{n=1}^{k} d_c\left(c_{jn}, c_{jn+1}\right) \leq S_t - k/3 \tag{5}$$

The above condition should satisfy to avoid the small set of trapping set [13][14]. This distance calculation of trapping set is the easiest way to eliminate the trapping set, whenever the encoder is progressive. But in the research the PEG avoidance with trapping set again attributed as scheduling process which is explained in chapter 6. So, this combination will result great performance in error floor region without the sacrificing of waterfall region.The second innovative approach of the research is non-zero neighborhood search in a spiral way. To know the spiral search, we should get the basic idea of different non-zero codeword search (NZCS) algorithms.

4. Non-Zero code word search

The minimum distance calculation of the codeword determines the error correction capability of the system. In iterative coding, the minimum distance determined by neighborhood search, is as a non-zero codeword to anticipate the message from noisy channel

Methods of NZCS

(i) Heuristic search

(ii) Branch-cut algorithm

(iii) Pseudo code word search

4.1 Heuristic Search

In this approach the non-zero codeword starts with a maximum step and the step size is reduced as the gradient of non-zero (NZ) element in neighborhood analysis. In this search method, the convergence of the search is not optimized. In LDPC irregular the gradient of NZ varies as random. So, this can be applied only for regular code. But irregular codes are indeed necessary to calculate NZCS.

4.2 Branch cut algorithm

Branch cut algorithm determines the neighborhood analysis by connecting and deleting the node. To branch, from one node (zero code word) to another node (non-zero codeword) is done by metric to maintain path. It needs memory in maintaining to branch and cut the path. Even though branch cut algorithm has a fast convergence of step, it required memory storage.

4.3 Pseudo code algorithm

This code word search as random as heuristic search, the algorithm contains 3 step to follows,

Step 1 : Initiate starting configuration of noise. If noise measure is high then the search is incremented as linearly and can find the closest pseudo code word $(\sigma^{(k)})$[11]

Step 2 : After that, comparison of median of noise $(x^{(k)})$ between pseudo code word(σ^{k})

Step 3 : If $(x^{(k)})$=$(x^{(k-1)})$ then the algorithm terminates. Whenever the current search median and next search medians are same then the searching process ends, otherwise step2 is repeated until Step 3 is obtained.

This pseudo search is linear search and it searches step depends upon the median noise. The median of Gaussian distribution random noise falls in the tail region. So, the non-zero codeword in that tail region observation is very hard to analyze in this search mechanism.

The above-discussed methods are de-merited by their own hardness and convergences. So this research proposes a new way to calculate neighborhood NZCS in a spiral way using the density function of NZ element.

5. Proposed Non-zero codeword search by spiral (NZCSS)

In this approach, the searching starts, from the center of codeword space set and the search path is in a spiral way. The path step is incremented and decremented by the density function on NZ.

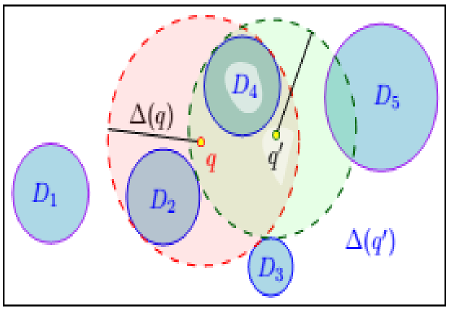

Fig.6 Spiral search with the query point (q) with different density of non-zero codeword

The above Figure 6 D_1,D_2…D_n values are representing density of non-zero codeword with noise. The step size $\Delta(q)$ calculated as minimum density of the function, and the minimum density can be a spiral always, so the search step path becomes spiral in nature [9]

$$\Delta(q)=min_{1<i<n}\Delta_i(q) \tag{6}$$

$$\text{NN}\neq 0\{\text{q,p}\} = \{p_i \,/\, \delta_i(q) < \Delta_i(q)\} \tag{7}$$

The neighborhood search is not as null vector when it falls in the above condition
where

$$\delta_i(q) = \ min_{p\in D1} d(x,q) = \max\{d(q,c_i\}\text{-}r_{i,1}\ \}$$

q- minimum distance query point
p_i- maximum distance
c_i-center of the plane
$r_{i,o}$ – radius of the non-zero density plane

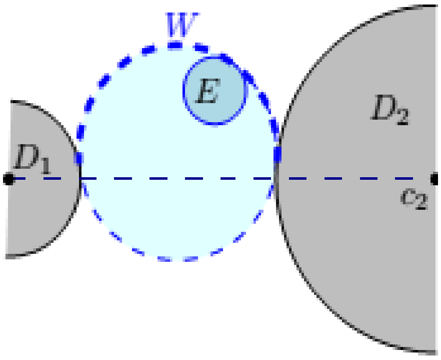

Fig. 7 Spiral search of NZCS in different density function with error

w- is the weight distance of the two noisy codeword.
The spread location probability is:

$$p_i=\{Pi1, Pi2, \ldots \ldots \ldots Pik\} \tag{8}$$

Pi1 is non-zero element distribution in i to kth direction.

$$wij = Pr[Pr = Pij] \tag{9}$$

wij - weight vector of the non-zero element in H matrix. Set S=$\bigcup_{i=1}^{m} Pi$ the quantity function referred as $and\ \rho = \frac{\max w_{ij}}{\min w_{ij}}$.

The nearest neighbor point from GF (q) is $m(\rho,\varepsilon)$ and ε error impulse pattern holds the uniform distribution up to upper bound P (e).

$$m(\rho,\varepsilon) = \rho\left(\frac{k}{n}\right)\left(\frac{\rho}{\epsilon}\right) + k - 1 \tag{10}$$

k is the nearest neighborhood element. Nearest non-zero codeword search is modified by the spiral and nonlinear search algorithm.[12]

The nearest neighborhood is calculated as:

$$\Pi_i(q) = \sum_{Pi,a\in P1} \Pi_{i,a}(q) \tag{11}$$

For i≤ n

$$\widehat{\Pi_\iota(q)}/[p_i \leftarrow \widehat{p_\jmath}] \text{ for j} \leq n \tag{12}$$

$$\forall\ i \leq n$$

$$|\Pi_i(q) - \widehat{\Pi_\iota(q)}| \leq\ \varepsilon \tag{13}$$

6. Results and discussion

The above condition maintained the search continuous and the search converges in a spiral way.

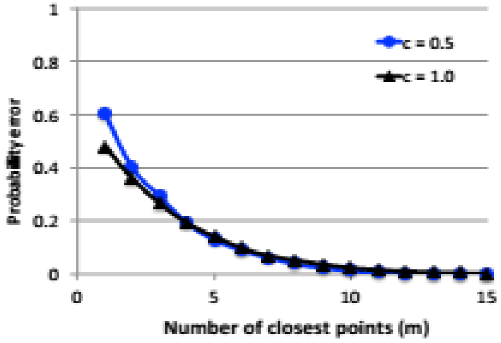

Fig.8 Spiral search for center (0.5&1) to the closest point

The result showed that if the number of point is more, then the probability of error is reduced, for a number of point 10, approximated probability of error 0.1 which is very low. So, the spiral search can give better result under the searching point in probability space by increasing center of searching values. If the non-zero closest points are increased about 10, the center values are not making an impact on both converging in the same level, as show in Figure 8

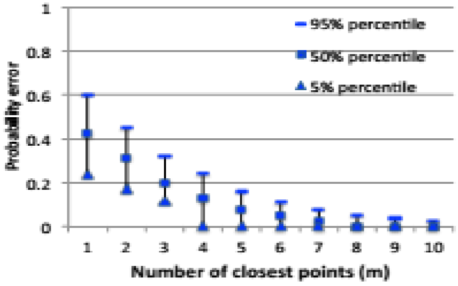

Fig. 9 NZCSS probability error for all points

Figure 9 shows the probability error for all points in a clustered way. This can be obtained as only 50 iteration number density function and the closest point for 4 is 0.35 probability error as 95 percentile and 0.15 probability of error as 50 percentile. So, all neighborhood points are codeword density calculated with 50 iterations. NZCSS method is fast convergence with proper weight updating of Non-zero codeword (NZCW)

The NZCSS algorithm converges as fast for irregular ensemble and density function and it strongly supports to search neighborhood instead of the median. Therefore, the stability of NZCSS algorithm is high compared with heuristic and pseudo codeword search. Spiral search inspects the neighborhood as a clear and fast way with less iteration. So, NZCSS method is adapted innovatively to LDPC decoder and it achieves low error floor region.

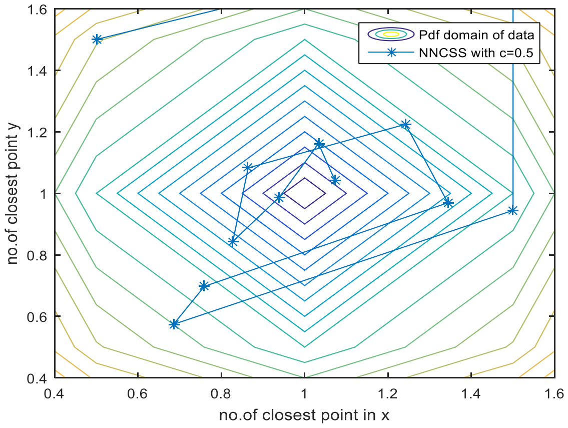

Fig. 10 Searching path convergence of NNCSS with c=0.5

The NNCSS algorithm can be modified as a first search by choosing radius as in different values; many values of center is simulated and plotted. For analysis purpose the sample space is taken as 1.6X1.6 and the pdf domain as contours with different PDF. When the searching started, the path travelled as spiral with Equation (11). The path searching is being divergence for low values. It can be converging by increasing the starting center values of search. The step size is increased depending up on the Probability density of non-zero values points.

The different values of starting search are simulated below: Figure 10 with c=1.2 shows that the search is converging with respect to the non-zero element. The candidate and the closest points are differentiated weight vector wij and $\widehat{\Pi_\iota(q)}$ estimated neighborhood function. Figure 3.11 shows that if the starting vector is increased then the searching step is varying and convergence as fast.

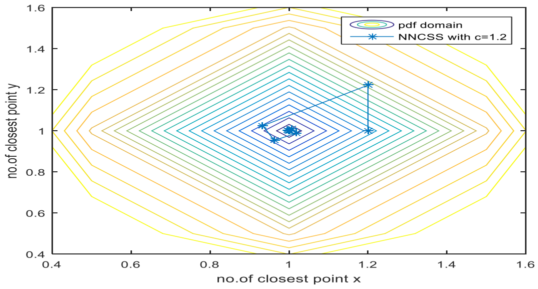

Fig. 11 NNCSS with c=1.2

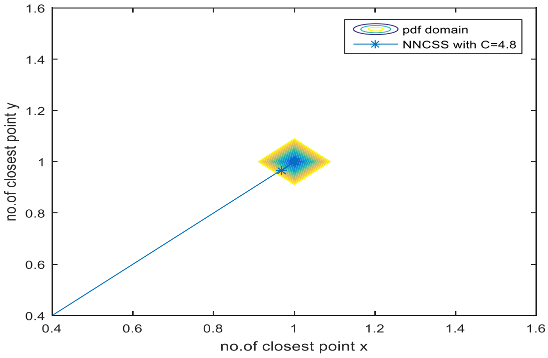

Fig. 12 NNCSS with c=4.8

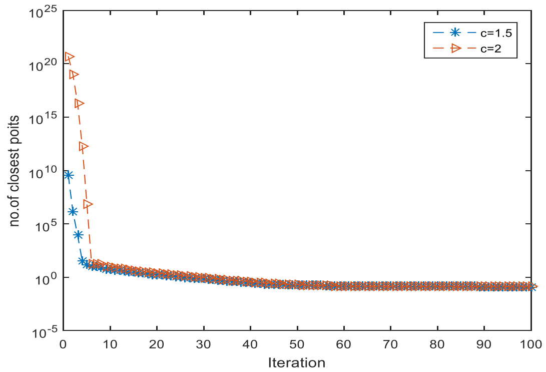

Fig.13 Convergence performance of NNCSS

7. Conclusion

The optimized point of starting vector is within 0.5 to 3 which will be the best fit values found in the NNCSS analysis. For any neighbor search algorithm mainly depends upon the convergence with iteration. Figure 13 shows the complete convergence of NNCSS. It is so obvious that it can converge its search by with in 10 iteration. Irrespective of any center values the convergence is obtained as much as fast as in 10 iteration with the matrix size of (600,930). This is the strength of this algorithm which has fast convergence than the Monte-Carlo simulation

References

[1] Gallager, RG 1962, ‘Low density parity check codes’, IRE Trans.inf.Theory IT-8, pp. 21-28.

[2] Sharon, E &Litsyn, S 2008, ‘Constructing LDPC codes by error minimization progressive edge growth’, IEEE Trans. Commun.,vol. 56, no. 3, pp. 359-368.

[3] Landner, S & Milenkovic, O 2005,‘Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes’, in: IEEE International Conference on Wireless Networks, Communications and Mobile, Computing, vol. 1, pp. 630-635.

[4] Vasic, B, Chilappagari, SK, Nguyen, DV & Planjery, SK 2009, ‘Trapping set ontology’, in: Proceedings of the 47th IEEE Annual Allerton Conference on Communication, Control, and Computing, pp. 1–7.

[5] Savin, V 2007, ‘Iterative LDPC decoding using neighborhood reliabilities’, in: IEEE International Symposium on Information Theory (ISIT), pp. 221-225.

[6] Chilappagari, SK, Nguyen, DV,Vasi´c, B & Marcellin, MW 2010,‘On trapping sets and guaranteed error correction capability of LDPC codes and GLDPC codes, IEEE Trans. Inf., Theory, vol. 56, no. 4, pp. 1600-1611.

[7] McGregor, A & Milenkovic, O 2010,‘On the hardness of approximating stopping and trapping sets’, IEEE Trans. Inf. Theory vol. 56, no. 4, pp. 1640-165

[8] Richter, G & Hof, A 2006, ‘On a construction method of irregular LDPC codes without small stopping sets’, in: Proceedings of the 2006 IEEE International Conference on Communications, Istanbul, Turkey,pp. 1119-1124.

[9] Michael Chertkov& Mikhail Stepanov, G 2007, ‘An efficient pseudocodeword search algorithm for linear programming decoding of LDPC Code’, IEEE Transactions on Information Theory.

[10] Zheng, X, Lau, F &Tse, C 2010,‘Constructing short-length irregular LDPC codes with low error floor’, IEEE Trans. Commun. vol. 58, no. 10, pp. 2823-2834.

[11] Pankaj Agarwal, K 2012, ‘Duke University, ‘Nearest Neighbor Searching Under Uncertainty II’. Duke University,

[12] Anand, A&Senthil Kumar, P 2015, ‘An efficient non-binary LDPC decoder with layer dynamic scheduling’,J. Comput.Theor. Nanosci,vol. 12, no. 12, pp. 5066-5070

[13] Tian, T, Jones, C, Villasenor, J & Wesel, R 2004, ‘Selective avoidance of cycles in irregular LDPC code constructions’, IEEE Trans. Commun. vol. 52, no. 8, pp. 1242-1247.

[14] Mitchell, DGM &Pusane, AE 2013, ‘Costello, D.J.Minimum distance and trapping set analysis of protograph-based LDPC convolutional codes’, IEEE Trans. Inf. Theory vol. 59, no. 1, pp. 254-281.