Asian Journal of Information Technology 15 (13): 2162-2166, 2016
ISSN: 1682-3915

Scheduled Progressive Edge Growth LDPC Encoder with Minimum Trapping Set

[1]A. Anand and [2]P. Senthilkumar
[1]RVS College of Engineering and Technology, 05 Dindigul, Tamil Nadu, India
[2]Department of Information and Technology, SKR Engineering College,
20 Chennai, Tamil Nadu, India

Abstract: Outstanding bit error rate LDPC design in waterfall region and error floor region is one of the challenging tasks for the past decade. In this study, we focus the design of LDPC encoder with low the error floor and waterfall region of BER with minimum trapping set Scheduled Progressive Edge Growth (SPEG) LDPC encoder innovatively, the simulation result of density evolution and exit chart give the better convergence of LDPC encoder, BER performance in error floor can controlled by minimum trapping set and waterfall region controlled by scheduled PEG LDPC encoder with (1000, 500) with code length (n) is <600. The girth of the SPEG encoder is 8. SPEG with minimum trapping set will perform well for short length code also and it converges faster than the other PEG encoder.

Key words: Scheduled PEG, density evolution, minimum trapping set, error floor, India

INTRODUCTION

LDPC codes are celebrated for its Shannon capacity (Gallager, 1962) and easy implementation in ASIC and FPGA. Former, the BER performance of LDPC code depends only on decoding mechanism. But, now implementation of proper encoder will increase the performance of BER in both error floor and waterfall region. Most of the studies (Richardson, 2003; Tian *et al*., 2004; Zheng *et al*., 2010) are focus only on error floor performance. LDPC ensembles design has lot techniques such as Quasi Cyclic (QC) LPDC, prototype, PEG with regular and irregular construction. QCLPDC construction with easy implementation with shift register and its permutation matrix (Z) plays vital role in design of QCLDPC. Prototype and lifting prototype mechanism yields good ensemble construction with fast convergence even better in irregular LDPC. But PEG technique has lot of flexibility of designing of ensembles in regular and irregular with high girth. If the girth of the LDPC is maximum then its stopping set size will decrease. Hence PEG with large girth will eliminate the problem of stopping set in the tanner graph. In this study, will enrich the ensemble design of PEG LDPC by two ways. Scheduled PEG directed towards fast convergence PEG mechanism growth in the fashion of minimum trapping set.

MATERIALS AND METHODS

PEG LDPC ensemble construction: LDPC ensembles can be represented by tanner graph edges and nodes as (V, C, E) where $V = \{v_1, v_2, ..., v_n\}$ is the set of Variable Nodes (VN), $C = \{v_1, v_2, ..., v_n\}$ is the set of Check Nodes (CN) and E is the set of edges. The edges are placed in the graph one by one, by processing one VN socket at a time. At the end of the process, a bisection is established between the VN sockets and CN (Check Node) sockets. This class of algorithms is known as the class of Progressive Edge-Growth (PEG) algorithms. The PEG algorithm is suited to construct the unstructured finite length LDPC code with large girth.

The motivation behind the PEG algorithm is to tackle the problem of increasing the girth of a Tanner graph by maximizing the local girth of a VN whenever a new edge is drawn from this VN toward the CN set. The PEG algorithm works for any number of VNs and CNs and for any VN degree distribution. Therefore, it is extremely flexible. For an irregular VN degree profile, ordering the VNs according to their degrees from the smallest to the largest and processing the VNs according to this ordering is in general beneficial. PEG algorithm node by node manner summarized as follows.

Algorithm A; PEG Algorithm

```
For j = 1 to n do
  For k = 1 to d_vj do
Determine C_vj ∈E
C_i←{C_vj| mindeg}
Add edge (V_j C_j,) to E
End for
  End for
```

$$D_s = \{d_{v1}, d_{v2}, ..., d_{vn} | d_{v1} \leq d_{v1} \leq ... \leq d_{sn}\}$$

Corresponding Author: A. Anand, RVS College of Engineering and Technology, 05 Dindigul, Tamil Nadu, India

where, D_s the target sequence of the variable node degrees sorted in non-decreasing order. We denote C_{vj} the set of check node whose distance V_j is maximum. If $E_{vj} \neq \phi$, C_{vj} can be determined by expanding a sub graph from variable node V_j up to maximal length. Finally we observe that check node degree distribution of the constructed Tanner graph is almost uniform. Finite length LDPC codes are characterized by a good compromise between waterfall and error floor performances. But finite length codes are not providing good error floor. Hence some modification is made in PEG algorithm for good error floor without sacrificing waterfall region.

Hence, the improvisation of PEG done by degree-by-degree manner minimizing the number cycles created (Ramamoorthy and Wesel, 2004) minimizing the Aapproximate Cycle Extrinsic (ACE) message degree PEG produce LDPC code graphs with significantly larger minimal stopping set compared with random construction algorithm. Comparing the method and the minimal trapping set, performs good in error floor region. But, finding trapping set from Tanner graph is NP hard problem. We will discuss the modified PEG by ACE and degree-by-degree with scheduling method.

Modified PEG algorithm: One of the key metrics that have been successfully adopted to improve the original PEG is referred as the Approximated Cycle Extrinsic message degree (ACE) of cycles of Tanner graph (Hu *et al*., 2005). The edges of VN are indexed from 0 to d_{vj}-1 and the Kthe edge of VN V_j is denoted by e_{vj}^{k} where $k \in \{0, \ldots, D_{vj}-1\}$. Moreover, the neighborhood of VNV_j within the depth l is denoted by N_{vj}^{l}. Denoting by $p_{vj,c}^{l}$ is the set of paths of length 2 l+1 from V_j to $c \in CA_{vj}^{l}$

Algorithm B; Modified PEG with ACE

If $\left|N_{vj}^{lmax+1}\right| = \left|N_{vj}^{lmax+1}\right| < m$ then
Set $e_{vj}^{k} = (C_1, V_j)$
End
Else
Determine the ACE of $p_{vj,c}^{lmax+1}$
Do: 2 until lowest degree of $p_{vj,c}^{lmax+1}$
End

The above PEG with ACE algorithm gives better error floor performance by giving penalty of waterfall region. Therefore scheduled PEG degree-by-degree (Sharon and Litsyn, 2008) is formed to overcome the tradeoff between waterfall region and error floor region.

Scheduled progressive edge growth algorithm proposed to improve the average inefficiency ($\bar{u}$) of irregular LDPC. The ensemble of irregular LDPC can be represented by fraction on node and edge. Let δ_d and γ_d are the fraction on variable node and check node of degree d. Let also λ_d and ρ_d are the fraction of edges connected to variable and check node of degree d. Where π is the random permutation matrix:

$$\lambda(x) = \sum_d \lambda_d x^{d-1} \rho(x) = \sum_d \rho_d x^{d-1} \quad (1)$$

$$\delta(x) = \sum_d \delta_d x^{d} \lambda(x) = \sum_d \gamma_d x^{d} \quad (2)$$

$$\text{Code rate (r)} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} \quad (3)$$

$$\text{And coding inefficiency } \mu(\pi) = \frac{{}^{k}\pi}{K} \quad (4)$$

It is assumed that girth of the graph goes infinity with the codeword n which actually happen for almost all the codes of irregular ensemble E(λ, ρ). It follows the decoding efficiency which can be expressed as $1\text{-}p_e/r$ also goes to threshold value: $\mu th = 1\text{-}p_{th}/r$ which will referred as inefficiency threshold. The density evolution of the irregular LDPC can be found by tracking the threshold with the ensemble E (λ, ρ). We consider the collection of discrete variable node subset $v_d^{(t)} c$ V indexed by $t \in \{1, 2, \ldots, T\}$ and $d \in \{1, 2, \ldots, d_{max}\}$:

$$v_d^{(t)} \subseteq V_d \quad (5)$$

$$\bigcup_{V=d=1}^{d_{max}} \bigcup_{t=1}^{T} v_d^{(t)} \quad (6)$$

and $v_d^{(t)}$ the number of variable node in:

$$n = \sum_{d=1}^{d_{max}} \sum_{t=0}^{T} n_d^{(t)} \quad (7)$$

Algorithm C; Scheduled PEG

For t =1 to T do
  For d = 1 to do
    For k = 1 to d do
For $Vj \in v_d^{(t)}$ do
Determine $C_{vj} \in E$
$C_i \leftarrow \{C_{vj} | \text{minding}\}$
Add edge (V_j, C_j) to E
    End for
    End for
  End for
End for

Different choice of scheduling subset $\{v_d^{(t)}\}$ might lead to codes with different performance. Even though the set optimized the there should be the penalty for waterfall region at some extent. SPEG providing good performance with error floor and waterfall region compared with ACE but exact calculation of scheduling subset is random distribution so we focus the scheduled PEG with avoiding minimal trapping set will yields good result in both domain.

Minimal trapping set of irregular code: A trapping ser for an iterative decoding algorithm is defined as a non-empty set of variable nodes, that are not eventually correct by the decoder []. A trapping set T is called an (a, b) trapping set if it contains a variable nodes and the sub graph induced by these nodes has b odd degree check nodes. T(a, b) is the subset of V, the set of variable nodes in T are connected to T atleast twice.

The size of stopping set T is defined as the cardinality of T. From the Fig. 1 set $\{v_2, v_6, v_9\}$ is a stoping set. It shown in [] that the set of erasures which remains when the iterarative erasure decoding algorithm stops is equal to the unique maximum stopping set:

$$d_c(i,j)\begin{cases}|i-j| \text{ for } i,j \le n_2 \\ \infty \text{ Otherwise}\end{cases} \qquad (8)$$

where, n_2 is the number of degree-2 variable nodes. In order to identify the non-selectable CNs a sub graph from

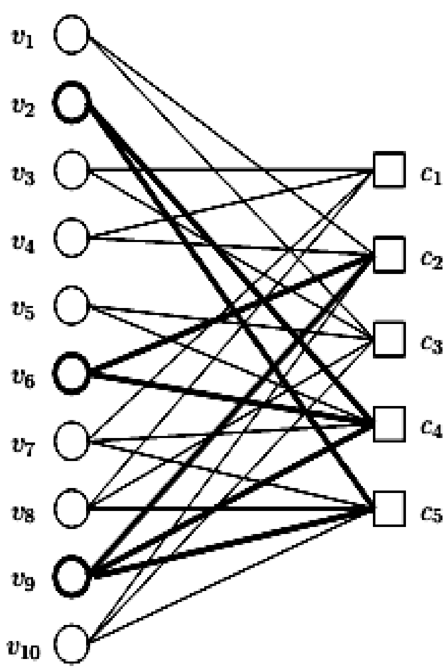

Fig. 1: An irregular LDPC code

VN V_j should be spread up to depth 2 []. The modification on algorithm 1 with minimal trapping set is reinforced by further condition of non-select ability on the surviving CNs in N_{vj}^{lmax}. By satisfying the Eq. 8 we can build PEG avoidance of even small trapping set. This will provide great performance in error floor region.

V proposed work speg with avoidance of small trapping set: This study gives the idea of SPEG and small trapping set condition. In SPEG the scheduled parameter $v_d^{(t)}$ calculation is trial and error problem. then it is optimized by diffrential evolution method. But, still calculation of scheduling parameter is exhaustive search, due to that the performance of error floor falls with some extent, hence this can be overcome by adapting the idea of avoidance of minimal strapping set (Tian *et al*., 2004) with SPEG will result outstanding performance in error floor region without sacrificing of waterfall region. This notion can be used for regular and irregular PEG LPDC construction. PEG with minimal trapping set algorithm applicable for BSC and AWGN also. Hence, the proposed work gives the universal use of algorithm with various family of PEG.

Algorithm D; SPEG with avoidance of minimal trapping set:

For t = 1 to T do
For d = 1 to d_{max} do
For k = 1 to d do
For $V_j \in v_d^{(t)}$ do
Determine $C_{vj} \in E$
If
{
Det.distance: d_c (i, j) = |i-j| for i, j≤n_2+1
Along the selectable survival path N_{vj}^{lmax}
$C_i \leftarrow \{C_{vj}|\text{minding}\}$
Add edge (V_j, C_j) to E
}
Else
}
Reject the survival path N_{vj}^{lmax}
$N_{vj}^{lmax} = N_{vj}^{lmax} + 1$
}
End for
End for
End for
End for

Algortihm 4 shows the degree-by-degree manner scheduled progressive check node and variable node but the elimination of trapping set from the small subset level will reduced the untraceable erasure when decoding process. Hence, the combination of these two techniques together yields good error floor performance.

RESULTS AND DISCUSSION

Figure 2 shows that performance of random LDPC ensemble (1024, 500) and decoded by Normalized

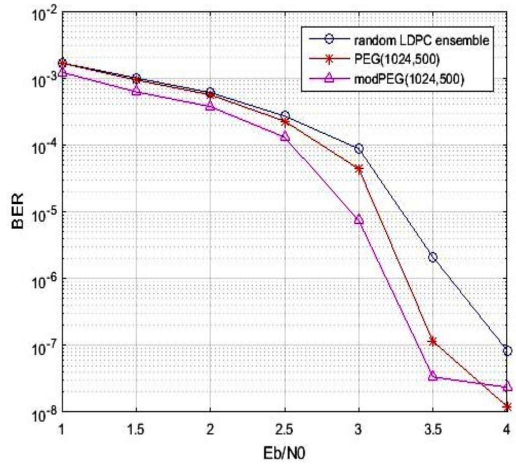

Fig. 2: BER vs Eb/N0 for random LDPCs ensemble and PEG with modPEG

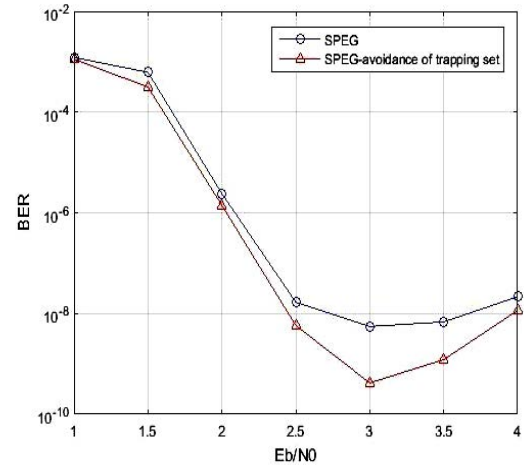

Fig. 3: BER vs Eb/No for SPEG (1000, 500) r = ½ N = 500. With avoidance of minimal of trapping set

Min-Sum algorithm (NMS) with 50 iteration under AWGN channel. Irregular (1024, 500) PEG ensemble generated with the girth = 9 and check node fractional edge polynomial is:

$$(x) = 0.32660\lambda^2+0.11960\lambda^3+0.18393\lambda^4+0.36988\lambda^5$$

Modified PEG with ACE is also plotted. This performance can also be analyzed in binary symmetric channel also (Richter, 2006).

It is obvious to infer from the BER curve at 3.5 dB the error region of mod-PEG falls and the error rate not reducing more than that the expected one. Comparing PEG and mod-PEG are performing very similar at water fall region (i.e., 2-3.2 dB).

SPEG scheduling factor calculated by mutating the parameter for 50 generation and is obtained in three level of scheduling. SPEG irregular (1000, 500) generated with the optimized scheduling factor and with fraction edge of check node $\lambda(x)$ and girth also 9 using.

Density evolution Mutual information threshold is 0.918. From Fig. 3 SPEG provide good performance in water fall region and error floor region. But, error floor get saturated at 3 dB onward. Because the trapping set lead to propagate the error in this level.

SPEG avoidance of small trapping set use the Eq. 8 to open the socket connection between check node to variable node. From the beginning itself the process of rejection node starts so the minimum trapping set is avoided. Figure 3 comparing to SPEG and avoidance trapping set good improvement of in the error floor up to 3.2 dB. So, the proposed work improvises the SPEG to some extent.

CONCLUSION

In this study, we have proposed the SPEG with minimal trapping set. The proposed algorithm exploits the work in two ways SPEG construction on ensemble with scheduling factor which enhance the waterfall region of BER Rejection path of PEG under the scheduling by avoidance of minimal trapping set. Option will reduce the error of NMS decoding in distributed way. So, the LDPC irregular ensemble (1000, 500) constructed with girth of 9 for the finite length N = 500; the simulation result show that this method behaves very well in waterfall region while also maintain error floor. This proposed work improvises the SPEG with 12%. Note that the error floor of after 3.5-4 dB is increasing further error rate. This is due to complete avoidance of trapping set is hardest task. Hence, we focus still to improvise the error floor level as our future research work.

REFERENCES

Gallager, R., 1962. Low-density parity-check codes. IRE. Trans. Inf. Theory, 8: 21-28.

Hu, X.Y., E. Eleftheriou and D.M. Arnold, 2005. Regular and irregular progressive edge-growth tanner graphs. IEEE. Trans. Inf. Theory, 51: 386-398.

Ramamoorthy, A. and R. Wesel, 2004. Construction of short block length irregular low-density parity-check codes. Proceedings of the 2004 IEEE International Conference on Communications, June 20-24, 2004, IEEE, Paris, France, ISBN: 0-7803-8533-0, pp: 410-414.

Richardson, T., 2003. Error floors of LDPC codes. Proceedings of the Annual Allerton Conference on Communication Control and Computing, October 1-3, 2003, Qualcomm Flarion Technologies, Bedminster, New Jersey, pp: 1426-1435.

Richter, G., 2006. Finding small stopping sets in the Tanner graphs of LDPC codes. Proceedings of the Turbo Codes and Related Topics 6th International ITG-Conference and 4th International Symposium on Source and Channel Coding (TURBOCODING), April 3-7, 2006, IEEE, Munich, Germany, ISBN: 978-3-8007-2947-0, pp: 1-5.

Sharon, E. and S. Litsyn, 2008. Constructing LDPC codes by error minimization progressive edge growth. IEEE. Trans. Commun., 56: 359-368.

Tian, T., C.R. Jones, J.D. Villasenor and R.D. Wesel, 2004. Selective avoidance of cycles in irregular LDPC code construction. IEEE. Trans. Commun., 52: 1242-1247.

Zheng, X., F.C. Lau and K.T. Chi, 2010. Constructing short-length irregular LDPC codes with low error floor. IEEE. Trans. Commun., 58: 2823-2834.