LDPC Code Construction with Low Error Floor Based on the IPEG Algorithm

Sung-Ha Kim, Joon-Sung Kim, Dae-Son Kim, and Hong-Yeop Song, *Member, IEEE*

***Abstract*— We propose a modification on the improved progressive-edge-growth(IPEG) algorithm. Proposed modification increases the connectivity of variable nodes using extrinsic message degree of variable nodes, which results in reducing the small stopping sets. Through computer simulation, we confirm that the codes constructed by the proposed algorithm have lower error floor than those constructed by the original IPEG algorithm.**

***Index Terms*— Progressive-edge-growth (PEG) algorithm, stopping set, extrinsic message degree(EMD).**

I. Introduction

LDPC codes are known to achieve near capacity performance with infinite block length (cycle-free case) [1][2]. In practice, however, we cannot avoid cycles due to the restriction on the block length. The belief propagation (BP) algorithm is optimum decoding for cycle-free case [3]. If cycles exist, the independence of each updated message cannot be maintained. Hence, it is reasonable to make cycles as large as possible to reduce the dependency between messages. In this point of view, the progressive-edge-growth (PEG) algorithm makes each variable node have the maximum local girth so as to construct LDPC codes with better performance than randomly constructed codes [4].

Performance of LDPC codes is dependent on the connectivity of variable nodes over binary erasure channel (BEC), since the connectivity is related to stopping sets that make iterative decoding fail over BEC. The connectivity can be calculated by extrinsic message degree (EMD) or approximate cycle EMD (ACE)[6]. Improved PEG (IPEG) algorithm [7] introduces the concept of ACE to PEG algorithm so that it can improve the performance of irregular LDPC codes at high SNR.

In this paper, we propose a modification on the IPEG algorithm. Proposed modification increases the connectivity of variable nodes using extrinsic message degree of variable nodes, which reduces the small stopping sets. Section II provides a brief overview of stopping sets and describes the connectivity of variable nodes in terms of EMD and ACE. Section III outlines the IPEG algorithm and explains the proposed modification. Section IV presents the simulation results.

Manuscript received October 9, 2006. The associate editor coordinating the review of this letter and approving it for publication was Prof. Jing Li. Support for this work was provided in part by Samsung Electronics under the Project on 4G Wireless Communication Systems. Part of this work was presented at IEEE VTC 2007 Spring.

S.-H. Kim was with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea. He is now with Samsung Electronics Co. Suwon, Korea.

J.-S. Kim, D.-S. Kim, and H.-Y. Song are with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea (e-mail: js.kim@yonsei.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2007.061650.

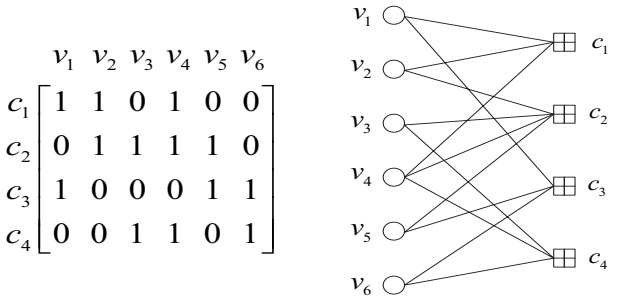

Fig. 1. Parity check matrix and bipartite graph.

II. The connectivity of variable nodes

The parity check matrix and bipartite graph have one-to-one correspondence as shown Fig. 1. A column(row, resp.) in the parity-check matrix corresponds to a variable(check, resp.) node in the bipartite graph. There exists an edge between variable node and check node if and only if $h_{ij} = 1$ where h_{ij} denotes the element of parity check matrix in the ith row and jth column.

Definition 1: (Stopping sets [5]) A subset S of variable nodes is said to form a stopping set if all its neighboring check nodes are connected to S at least twice.

For example, the subset of variable nodes $A = \{v_3, v_5, v_6\}$ form a stopping set in Fig. 1. However the subset $B = \{v_1, v_2, v_3\}$ is not a stopping set, since B has two neighboring check nodes, c_3 and c_4 which are connected to B only once. In parity check matrix, the stopping sets also can be determined by the ordinary (not mod 2) sum of the columns over S. S is stopping set if 1 does not appear in the sum [8]. The sum of columns over A is $[0\ 2\ 2\ 2]^T$ and there is no 1's in it.

It is proved that preventing small stopping sets prevents small minimum distance [6]. To avoid small stopping sets, we have to make the subset of variable nodes have many extrinsic edges. The quantities of these extrinsic edges are defined as follows.

Definition 2: (EMD [6]) An extrinsic check node of a variable node set is a check node that is singly connected to this set. The EMD of a variable node set is the number of extrinsic check nodes of this variable node set.

It can be easily shown that EMD of a stopping set is zero and the number of 1's in the sum of the columns over subset of variable nodes is equal to EMD. In the above example, the column sum over B is $[2\ 2\ 1\ 1]^T$, so the EMD of B is 2.

Definition 3: (ACE [6]) The ACE of a length $2d$ cycle is $\sum_i (d_i - 2)$, where d_i is the degree of the ith variable node in this cycle.

When variable nodes and their neighboring check nodes compose of a single cycle, the EMD of these variable nodes is equal to the ACE. However, if they compose of multi-cycles, the ACE becomes an upper bound of EMD. In Fig. 1, variable nodes $\{v_2, v_3, v_4\}$ and their neighboring check nodes compose of 6-cycle and ACE is one. However, EMD of these variable nodes is zero since this 6-cycle contains a 4-cycle. In next section, we propose the modified IPEG algorithm using these properties of EMD and ACE.

III. Modification on IPEG algortihm

Standard PEG algorithm with n variable nodes and m check nodes is as following [4]. (We use same notation in [4].)

for $j = 0$ *to* n - 1 *do* {
 for $k = 0$ *to* d_{s_j} - 1 *do* {
 if $k = 0$ {
 $E_{s_j}^0 \leftarrow$ *edge* (c_i, s_j), *where* $E_{s_j}^0$ *is the first edge incident to symbol node* s_j, *and* c_i *is one check node such that it has the lowest check node degree under the current graph setting* $E_{s_0} \bigcup E_{s_1} \bigcup \cdots \bigcup E_{s_{j-1}}$}
else {
 expand a subgraph from symbol node s_j *up to depth* l *under the current graph setting such that the cardinality of* $N_{s_j}^l$ *stops increasing but is less than* m, *or* $\overline{N}_{s_j}^l \neq \varnothing$ *then* $E_{s_j}^k \leftarrow$ *edge* (c_i, s_j), *where* $E_{s_j}^k$ *is the* k*th edge incident to* s_j, *and* c_i *is one check node picked from the set* $\overline{N}_{s_j}^l$ *having the lowest check node degree* }}}

In above algorithm, d_j is the degree of variable node s_j. $E_{s_j}^k$ presents the kth edge incident on s_j and E_{s_j} is the set of edges incident on s_j. Finally, $N_{s_j}^l$ means the subset of check nodes reached by a tree spreading from variable node s_j within depth l and $\bar{N}_{s_j}^l$ is complement of $N_{s_j}^l$.

Let the set of the candidate check nodes corresponding to the k^{th} edge of symbol node s_j be $\Omega_{s_j}^k$. When the cardinality of $\Omega_{s_j}^k$ is lager than one, PEG algorithm selects a check node in $\Omega_{s_j}^k$ at random. However IPEG algorithm introduces the idea of ACE and proposes the criterion for selection. IPEG algorithm chooses a check node in $\Omega_{s_j}^k$ that maximizes the minimum ACE for the new cycles. Similar to PEG algorithm, IPEG algorithm picks a check node at random if there are more than two check nodes that have same ACE condition. Instead of selecting at random in IPEG algorithm, we propose an algorithm that determines which check node is better to increase the connectivity of variable nodes.

Let $\Phi_{s_j}^k$ be the subset of check nodes classified by IPEG criterion. It is obvious that $\Phi_{s_j}^k$ is the subset of $\Omega_{s_j}^k$. When there are more than two elements in $\Phi_{s_j}^k$, we can extract subgraphs that are subsets of tree-expanded PEG graph. Each subgraph is constituted by back-tracking from a check node in $\Phi_{s_j}^k$ to root s_j. Hence, we know which variable nodes are contained in each subgraph and we can easily check which subgraphs have higher EMD by counting the number of 1's in the column sum over the variable nodes in each subgraph. In the end, we are able to choose a check node in $\Phi_{s_j}^k$ with higher EMD. Figure 2 is an example of selecting a check node among two candidate check nodes that have the same ACE but different EMD. These two check nodes are regarded as equivalent in IPEG algorithm criterion. However, we can make better selection by the proposed algorithm. In Fig. 2, left-hand side candidate check node is selected since it has the higher EMD than the right-hand side. It is possible there also exist more than two check nodes in $\Phi_{s_j}^k$ that have the same EMD condition. In this case, we select the check node whose corresponding subgraph has more variable nodes and if we cannot determine which check node is better using all these criteria (girth, ACE, EMD and more variables), then we pick a check node at random.

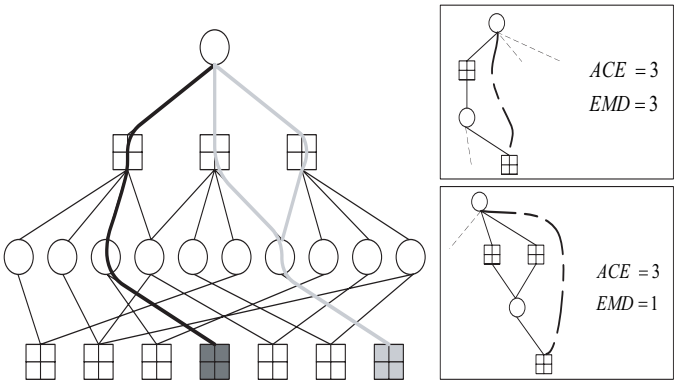

Fig. 2. Tree-expanded PEG graph and extracted subgraphs.

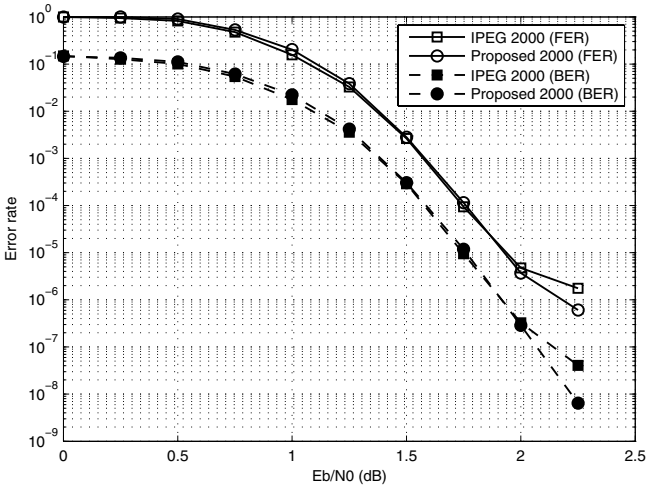

Fig. 3. FER/BER performance (length 2000).

IV. Simulation Results

We construct four irregular LDPC codes with block length 2000 and 3000. For each block length, one is constructed by IPEG algorithm and the other is constructed by proposed algorithm with good variable nodes degree distribution in [2], $\lambda_1(x)$ for block length 2000 and $\lambda_2(x)$ for block length 3000. We use BP algorithm with maximum number of iteration 200 in AWGN channel.

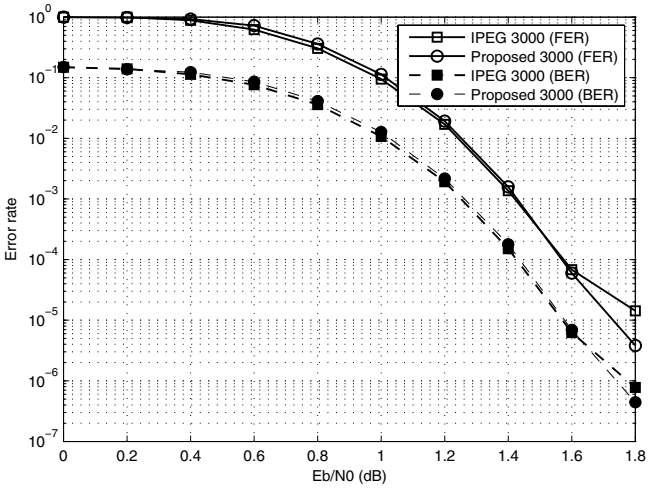

Fig. 4. FER/BER performance (length 3000)

$$\begin{aligned}\lambda_1(x) &= 0.21991x + 0.23328x^2 + 0.02058x^3 \\ &\quad +0.08543x^5 + 0.06540x^6 + 0.04767x^7 \\ &\quad +0.01912x^8 + 0.08064x^{18} + 0.22798^{19}\end{aligned}$$

$$\begin{aligned}\lambda_2(x) &= 0.19606x + 0.24039x^2 + 0.00228x^5 \\ &\quad +0.05516x^6 + 0.16602x^7 + 0.04088x^8 \\ &\quad +0.01064x^9 + 0.00221x^{27} + 0.28636x^{29}\end{aligned}$$

Figure 3 show the frame error rate (FER) and bit error rate (BER) curves for block length 2000. The code constructed by the proposed algorithm shows better performance at high SNR region than that constructed by IPEG algorithm. At low SNR region, proposed algorithm has no loss in performance compared to IPEG algorithm. In Fig. 4 with block length 3000, we confirmed the similar results. The codes constructed by IPEG algorithm show error floor at FER 10^{-5} with block length 2000 and at FER 10^{-4} with block length 3000. Note that the codes constructed by proposed algorithm does not at these FER region.

V. Conclusion

In this paper, we discuss the connectivity of variable nodes and propose more specific conditions to determine which check node is better to prevent small stopping sets in the subset classified by IPEG algorithm. From computer simulation, we confirm that proposed algorithm constructs LDPC codes to have lower error floor than IPEG algorithm.

References

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-B, pp. 21-28, Jan. 1962.

[2] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Trans. Inf. Theory*, vol. 47, pp. 619-637, Feb. 2001.

[3] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, pp. 533-547, Sept. 1981.

[4] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth Tanner graphs," in *Proc. IEEE GLOBECOM '01*, vol. 2, pp. 995-1001, Nov. 2001.

[5] C. Di, D. Proietti, I. E. Telater, T. J. Richardon, and R. L. Urbanke, "Finite-length analysis of low-density parity check codes on the binary erasure channel," *Proc. IEEE Tran. Inf. Theory*, vol. 48, no. 6, pp. 1570-1579, June 2002.

[6] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Construction of irregular LDPC codes with low error floors," in *Proc. IEEE Int. Conf. Comm.*, vol. 5, pp. 3125-3129, May 2003.

[7] H. Xiao and A. H. Banihashemi "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Commun. Lett.*, vol. 8, pp. 715-717, Dec. 2004.

[8] A. Ramamoorthy and R. Wesel, "Construction of Short block length irregular low-density parity-check codes," in *Proc. IEEE Int. Conf. Comm*, vol. 1, pp. 410-414, June 2004.