The Effect of PEG-Lifting Order on the Performance of Protograph GLDPC Codes

Dae-Young Yun
Electrical and Computer Engineering
Seoul National University
Seoul, Republic of Korea
dyyun@ccl.snu.ac.kr

Jae-Won Kim
Electronic Engineering
Gyeongsang National University
Jinju, Republic of Korea
jaewon07.kim@gnu.ac.kr

Jong-Seon No
Electrical and Computer Engineering
Seoul National University
Seoul, Republic of Korea
jsno@snu.ac.kr

2022 27th Asia Pacific Conference on Communications (APCC) | 978-1-6654-9927-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/APCC55198.2022.9943671

***Abstract*—Generalized low density parity check (GLDPC) codes can be constructed by replacing some single parity check (SPC) nodes in LDPC codes with generalized constraint (GC) nodes. GC nodes are defined by component codes whose minimum distance is larger than that of SPC nodes. Therefore, the variable nodes (VNs) connected to GC nodes, which are called *doped* VNs, are more protected than the *undoped* VNs. Due to this effect, we observe that the doped VNs are more robust to local cycles. The distribution of local cycles is affected by the processing VN order of the progressive edge growth (PEG) algorithm, where the latter processed (lifted) VNs tend to have more local cycles. Based on the property of doped VNs and the PEG algorithm, we show that a tangible performance gain is achieved by placing the doped VNs in the latter order of the PEG algorithm compared to the former order. The performance gain is shown with a well known GLDPC code in the literature and over both the binary erasure channel and addictive white Gaussian noise channel.**

***Index Terms*—Generalized low density parity check (GLDPC) codes, low density parity check (LDPC) codes, protograph, progressive edge growth (PEG) algorithm.**

I. Introduction

Low density parity check (LDPC) codes, which were first introduced in [1] and rediscovered in [2], have been widely used due to their capacity-approaching performance under low-complexity iterative decoding. Generalized LDPC (GLDPC) codes [3] are generalized versions of LDPC codes, which incorporate not only single parity check (SPC) nodes but also generalized constraint (GC) nodes as their check nodes (CNs). GC nodes represent more complex linear component codes than single parity check codes and thus they give stronger protection to neighbor variable nodes (VNs). One method to construct GLDPC codes is replacing some SPC nodes in LDPC codes with GC nodes. This replacement is called *doping* [4], and the VNs connected with GC nodes are called *doped* VNs. The *undoped* VNs are only connected with SPC nodes, so they are less protected than the doped VNs.

The progressive edge growth (PEG) algorithm [5] is the most popular algorithm for lifting protograph-based LDPC codes. In an edge-by-edge manner, PEG connects a VN with a CN while maximizing the local girth in a greedy way. Due to the greedy behavior, local cycles tend to be more generated in the latter processed VNs. In other words, the backward VNs in terms of the PEG schedule may have much more local cycles than the front VNs. Since PEG-lifted codes have this inevitably unbalanced distribution of local cycles, we need to consider the processing order of the PEG algorithm carefully.

In this paper, we investigate the effect of the PEG processing order for constructing protograph GLDPC codes. While doped VNs are protected by strong GC nodes, undoped VNs are relatively more vulnerable to channel noise. To balance the robustness of each VNs, we propose the PEG scheduling that lifts undoped VNs first to make less local cycles with them and then lifts doped VNs later. Experimental results show that the latter doped codes which lifted by the proposed PEG ordering outperform the former doped codes under the same doped positions.

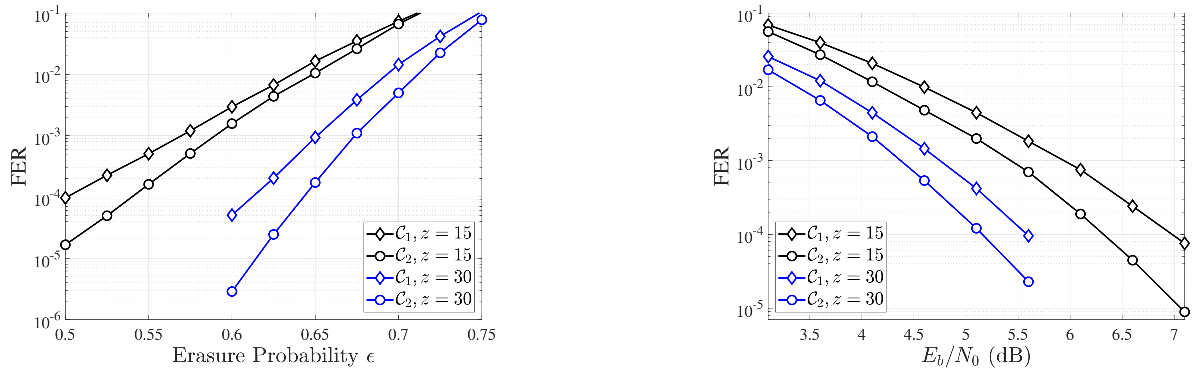
II. The Structure of Protograph GLDPC Codes

A protograph can be represented by a base matrix. Let $\mathbf{B}$ denote an $m_p \times n_p$ base matrix whose element $b_{i,j}$ represents the number of connection between the ith CN and jth VN in the protograph, where n_p, m_p are the numbers of protograph VNs and CNs, respectively. For quasi-cyclic (QC) LDPC codes, parity check matrices (PCMs) $\mathbf{H}$ are obtained by replacing elements of $\mathbf{B}$ with $z \times z$ circulant permutation matrices (CPMs) with the lifting size z.

Different from LDPC codes, we need one more procedure to obtain PCMs of GLDPC codes. Since GC nodes correspond to multiple SPC nodes representing their component codes, we have to replace GC nodes with their component PCMs to make full PCMs of GLDPC codes. Instead of using PCMs straight for iterative decoding like LDPC codes, GLDPC codes employ a posteriori probability (APP) decoding for GC nodes in general.

One of well-known protograph GLDPC codes with near-capacity performance is described as the base matrix in (1). The first row with the bold numbers represents the GC node whose component code is the (7,4) Hamming code. One can see the sum of the first row equals to the length of the component code.

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{4} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{1}$$



(a) (b)

Fig. 1. Performance results of GLDPC codes lifted from protograph (1) with different PEG-lifting orders over (a): the BEC and (b): the AWGN channel.

III. Experiment Settings

In this paper, we investigate the effect of the PEG-lifting order by comparing two different orders with the same base matrix (1). We suppose that the PEG algorithm operates in order of the base matrix and then the different orders can be described in terms of two base matrices as follows.

$$\begin{bmatrix} \mathbf{1} & \mathbf{4} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 3 & 1 \end{bmatrix}, \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{4} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 3 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Let $\mathcal{C}_1$ denote the code sets lifted in the order of the left matrix in (2), which we call the former doped codes, and $\mathcal{C}_2$ denote the latter doped codes. The PEG algorithm is performed for each case from left to the right direction in an edge-by-edge manner, which means the right nodes have worse cycle characteristics than the left ones. Note that the left matrix in (2) places doped VNs in front, while the right matrix in (2) places the doped VNs in back. Therefore, the former doped codes will generate more cycles in the vulnerable undoped positions, while latter doped codes will make relatively sparse cycle distribution in the undoped position.

We use lifting factor $z = 15, 30$, and the PEG selects a CN randomly if there are multiple CN candidates making the tie local girth. Due to this randomness, we make 50 codes for each case and compare the average performance like the previous PEG works [7]. We transmitted all-zero codewords over binary erasure channel (BEC), and additive white Gaussian noise (AWGN) channel. Received codewords were decoded by iterative decoder performing maximum likelihood (ML) decoding in GC nodes.

IV. Results and Conclusion

The decoding results are presented in Fig. 1. Both results show significant frame error rate (FER) gaps between two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ although they share the same protograph structure, and accordingly the same asymptotic performance such as the BP threshold [8]. Over the BEC, $\mathcal{C}_2$ with $z = 30$ acheives 15x lower FER than $\mathcal{C}_1$ at $\epsilon = 0.6$. Over the AWGN channel, $\mathcal{C}_2$ with $z = 15$ attains about 0.5dB performance gain compared to $\mathcal{C}_1$ at FER 10^{-4}.

By further looking into the performance of each code instance of 50 randomly generated codes, we confirm that the latter doped codes $\mathcal{C}_2$ can suppress the bad effect of local cycles. The difference between the best and worst FER performance among 50 codes in $\mathcal{C}_2$ is within in 3x while the difference for $\mathcal{C}_1$ is 400x over the BEC with z=30, which implies that the lifting order of $\mathcal{C}_2$ results in much more stability against the greedy PEG algorithm. This result demonstrate the lifting order of $\mathcal{C}_2$ operates much more robustly against unstable local-greedy behavior of the PEG algorithm.

In this paper, we demonstrates that the undoped VNs which are not protected by GC nodes are vulnerable to local cycles, therefore the undoped VNs should be lifted first in the PEG procedure to balance the local correction capability.

Acknowledgment

This work was supported in part by Samsung Electronics Co., Ltd(IO201208-07823-01), and in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1G1A1091583).

References

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
[2] D. J. C. Mackay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *IEEE Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, 1996.
[3] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.
[4] G. Liva, W. E. Ryan, and M. Chiani, "Quasi-cyclic generalized LDPC codes with low error floors," *IEEE Trans. Commun.*, vol. 56, no. 1, pp. 49–57, Jan. 2008.
[5] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IRE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386-398, Jan. 2005.
[6] A. K. Pradhan and A. Thangaraj, "Near-capacity protograph doubly-generalized LDPC codes with block thresholds," in *Proc. IEEE ISIT*, 2016.
[7] G. Richter and A. Hof, "On a construction method of irregular LDPC codes without small stopping sets," in *Proc. IEEE int. Conf. Commun.*, vol. 3, pp. 1119–1124, Jun. 2006.
[8] G. Liva and M. Chiani, "Protograph LDPC codes designed based on EXIT analysis," in *Proc. IEEE GLOBECOM*, 2007.