2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Constrained Scale-Free LDPC Codes

Xia Zheng†, Francis C.M. Lau†, Yejun He‡ and Chi K. Tse†

†Department of Electronic and Information Engg., Hong Kong Polytechnic University, Hong Kong
‡College of Information Engineering, Shenzhen University, Shenzhen
Email: †{xia.zheng,encmlau,encktse}@polyu.edu.hk; ‡yjhe@szu.edu.cn

Abstract—Short-length scale-free low-density-parity-check (SF-LDPC) codes have been found to outperform codes optimized by the density-evolution technique. However, a large fraction of degree-2 variable nodes will exist in SF-LDPC codes when the code rate is high. Consequently, small-size cycles with no external connections can easily be created by these degree-2 nodes, producing a large error rate. In this paper, we propose constraining the proportion of degree-2 variable nodes in the design SF-LDPC codes. We will evaluate the error performance of the high-rate SF-LDPC codes under the new constraint.

1. Introduction

Low-density-partiy-check (LDPC) codes are well-known for their superb error-correction capability. Moreover, LDPC codes optimized by the density-evolution (DE) technique have been widely studied [1, 2]. Recently, we have applied complex-network theories to the design of LDPC codes. Inspired by the shortest-average-path-length property of scale-free networks, we have proposed scale-free LDPC (SF-LDPC) codes in which the variable-node degrees follow power-law distributions [3, 4]. We have also constructed several short-length SF-LDPC codes with code rate 0.5 and have evaluated their performance. The results have shown that short-length SF-LDPC codes outperform other DE-optimized codes in terms of block/bit error rates and convergence time.

Yet, when the code rate is higher, say 0.75, there will be a large proportion of degree-2 variable nodes existing in the SF-LDPC codes. Under such circumstances, there is a high chance that short cycles, with no externally-connected check nodes, will be formed by the degree-2 variable nodes. Consider the short cycle in Fig. 1. In the iterative decoding process, the information generated by each variable node will spread to all other variable nodes via the check nodes. Suppose an all-zero codeword is sent by the transmitter. When most, even if not all, of the variable nodes in the cycle in Fig. 1 are erroneous (decoded as “1”), the (erroneous) message starting from any of the variable nodes will be enhanced as it passes through each of the erroneous variable nodes in the cycle. After a number of iterations (equal to the number of variable nodes in the cycle), the enhanced message will return to the same variable node, further reinforcing the incorrect belief of the variable node. The end result is that all variable nodes in

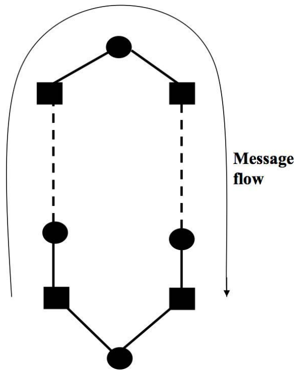

Figure 1: A short cycle formed by degree-2 nodes and without any externally-connected check nodes. Filled circles and filled squares represent, respectively, variables nodes and check nodes. A solid line represents a direct connection between a variable node and a check node. A dash line represents that there is a path between the check node and variable node.

the cycle will become erroneous. Moreover, if all the variable nodes residing outside the cycle have been decoded correctly as bits “0”, all the check nodes will become satisfied and the decoder converges — to an incorrect codeword though. In consequence, such a short cycle is very likely to give rise to errors, even at the high SNR region [5, 6].

In this paper, we will overcome the aforementioned issue by introducing a new constraint on SF-LDPC codes. We will impose an upper bound on the proportion of degree-2 variable nodes in the SF-LDPC codes. In Sect. 2, we briefly review the construction method of SF-LDPC codes and in Sect 3, we will introduce the proposed “constrained SF-LDPC (CSF-LDPC) codes”. Finally, we will present the performance of the CSF-LDPC codes in Sect. 4.

2. Review of scale-free LDPC codes

Low-density-parity-check codes are in fact linear block codes [7] which can be represented by bipartite graphs consisting of two sets of nodes, namely *variable nodes* and *check nodes*. The variable nodes represent the elements of the codeword and the check nodes represent the sets of parity-check constraints satisfied by the codewords of the code. The block length of the code, denoted by N, is the number of variable nodes; while the check length of the code, denoted by M, is the number of check nodes. The connections between the two different types of nodes are called *edges*. The number of edges emanated from a node is referred to as the *degree* of the node.

Given a distribution pair (λ, ρ) of an LDPC ensemble,

$$\lambda(x) := \sum_{k=2}^{d_v} \lambda_k x^{k-1} \quad \text{and} \quad \rho(x) := \sum_{k=2}^{d_c} \rho_k x^{k-1} \qquad (1)$$

specify the variable-node degree distribution and the check-node degree distribution, respectively, where λ_k denotes the fraction of edges connected to degree-k variable nodes and ρ_k denotes the fraction of edges connected to degree-k check nodes. Moreover, d_v and d_c denote the maximum variable-node degree and maximum check-node degree, respectively.

In the following, we review the main steps in constructing a SF-LDPC code. For details, please refer to [4].

1. Denote the probability that a variable node has k connections by $\Pr_\lambda(k)$.
2. Assign the fraction of variable nodes with degree k according to a power-law function, i.e., $\Pr_\lambda(k) \propto k^{-\gamma}$, where γ is the characteristic exponent.
3. Since $\sum_k \Pr_\lambda(k) = 1$, the fraction of edges connecting to variable nodes with degree k equals $\lambda_k = \frac{k^{1-\gamma}}{\sum_{i=2}^{d_v} i^{1-\gamma}}$.
4. Variable-node degree distribution in (1) is re-written as $\lambda(x) = \sum_{k=2}^{d_v} \frac{k^{1-\gamma}}{\sum_{i=2}^{d_v} i^{1-\gamma}} x^{k-1}$.
5. Average variable-node degree equals $< k_v >= \frac{\sum_{k=2}^{d_v} k^{1-\gamma}}{\sum_{i=2}^{d_v} i^{-\gamma}}$.
6. Probability that a check node has $k \in \{d_c - 2, d_c - 1, d_c\}$ connections follows a Poisson distribution with parameter ν, i.e., $\Pr_\rho(k) = \frac{\nu^k \exp(-\nu)}{k!}$.
7. Fraction of edges connecting to check nodes with degree $k \in \{d_c - 2, d_c - 1, d_c\}$ equals $\rho_k = \frac{\frac{\nu^k \exp(-\nu)}{(k-1)!}}{\sum_{j=d_c-2}^{d_c} \frac{\nu^j \exp(-\nu)}{(j-1)!}}$.
8. Check-node degree distribution in (1) becomes
$$\rho(x) = \sum_{k=d_c-2}^{d_c} \frac{\frac{\nu^k \exp(-\nu)}{(k-1)!}}{\sum_{j=d_c-2}^{d_c} \frac{\nu^j \exp(-\nu)}{(j-1)!}} x^{k-1}. \qquad (2)$$
9. Combining the above results gives, for a code rate R,
$$\frac{< k_v >}{1-R} = \frac{(d_c-2)(d_c-1)d_c + (d_c-1)d_c\nu + d_c\nu^2}{(d_c-1)d_c + d_c\nu + \nu^2}. \qquad (3)$$
10. Since d_c is an integer greater than 2, we have $d_c - 2 < \frac{<k_v>}{1-R} < d_c$ and $d_c = \left\lceil \frac{<k_v>}{1-R} \right\rceil, \left\lceil \frac{<k_v>}{1-R} \right\rceil + 1$ where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x.
11. Once d_c is selected, the corresponding ν is found using (3).

3. Constrained SF-LDPC codes

When LDPC codes are randomly constructed, there is a high probability that small cycles, say cycles of length less than 10, consisting of only degree-2 variable nodes are formed. These small-size cycles are creating decoding errors even at the high SNR region. There are effective code construction algorithms, such as progressive edge growth (PEG) [8], that can maximize the length of possible cycles involving only degree-2 variable nodes. However, if the number of degree-2 variable nodes, denoted by N_{v2} is far larger than the check length M, the excess number of degree-2 nodes over the check length, i.e., $N_{v2} - M$ degree-2 variable nodes, will produce small-size cycles, giving rise to a high error rate. Therefore, in practice, the fraction of degree-2 variable nodes in any optimized degree distributions should not greatly exceed $\frac{M}{N} = 1-R$. To overcome the aforementioned problem, an additional constraint has been proposed when the fraction of DE-optimized degree-2 variable nodes is far larger than $1 - R$ [9, 10]. The DE mechanism that has incorporated the degree-2 variable-node constraint is called "constrained DE", and the corresponding degree distribution obtained is called "constrained degree distribution".

Applying the above concept to the SF-LDPC codes, we can form "constrained SF-LDPC" codes. Denote the fraction of degree-2 variable nodes by $F_r(2) = \frac{N_{v2}}{N}$. For constrained SF-LDPC codes with a maximum variable-node degree of d_v, the variable-node degree distribution will be given as

$$\lambda_k = \begin{cases} \dfrac{2F_r(2)}{2F_r(2) + \frac{\sum_{i=3}^{d_v} i^{1-\gamma}(1-F_r(2))}{\sum_{i=3}^{d_v} i^{1-\gamma}}} & \text{if } k = 2 \\[2ex] \dfrac{k^{1-\gamma}(1-F_r(2))}{2F_r(2) + \frac{\sum_{i=3}^{d_v} i^{1-\gamma}(1-F_r(2))}{\sum_{i=3}^{d_v} i^{1-\gamma}}} & \text{otherwise.} \end{cases} \qquad (4)$$

Using the same method as described in Sect. 2, the optimized check-node degree distribution, ν, d_c and $< k_v >$ can be readily found.

4. Results and Discussions

In this section, we present the analytical performance and the simulated results for the constrained SF-LDPC (CSF-LDPC) codes. We assume an AWGN channel and a code rate of 0.75.

4.1. Achievable Threshold

Assuming a rate-0.75 code, we use the algorithm in Sect. 2 to attempt optimizing the degree distributions for the SF-LDPC codes. However, the proportional of degree-2 variable nodes obtained, i.e., $F_r(2)$, is far larger than $1 - R = 0.25$ and is therefore not acceptable. Hence, we resort to constructing rate-0.75 constrained SF-LDPC codes.

Table 1: Comparison of the threshold value and the average number of connections between constrained SF-LDPC codes and other LDPC codes. Code rate R equals 0.75. The letters in the code name denote the type of code, including DE, constrained DE (abbreviated by "CDE") , and constrained SF-LDPC (abbreviated by "CSF"). The digits in the code name denote the maximum variable-node degree of the code.

Code Name	d_v	$F_r(2)$	σ^*	$< k_v >$	d_c	γ
DE14	14	0.446	0.664	4.526	20	/
CDE12	12	0.250	0.663	4.000	16	/
CSF12	12	0.278	0.647	3.974	17	2.38
CSF20	20	0.280	0.651	4.005	17	2.80
CSF28	28	0.294	0.653	4.054	17	2.90

In addition, to ensure an easy implementation of the encoder, $F_r(2)$ is set equal to or slightly larger than $1 - R$ [9].

We begin with $F_r(2) = 1 - R$, and increase it with a step size of 0.001 until $1 - R + 0.05$ is reached. For each value of $F_r(2)$, the largest achievable threshold σ^*[1] and the corresponding constrained degree distributions of the SF-LDPC code are recorded. Among all the results, the largest threshold and the corresponding optimized, constrained degree distributions of the CSF-LDPC code is then selected. Table 1 presents the highest thresholds achieved by rate-0.75 CSF-LDPC codes as well as other LDPC codes. The corresponding parameters used are also tabulated. The results indicate that the "pure" DE produces a slightly larger σ^* compared with other LDPC codes. However, "pure" DE also produces the fraction of degree-2 variable nodes almost two times of $(1-R)$, i.e., $F_r(2) \approx 2(1-R)$. We also observe that constrained DE and constrained SF-LDPC produce very similar σ^* and $< k_v >$.

4.2. Simulated Performance

Further, three codes of rate-0.75 are constructed using the progressive-edge-growth (PEG) method, which has been shown to produce codes with both large girth and large Hamming distance [8]. The first one has a DE-optimized variable-node degree distribution given by

$$\lambda_1(x) = 0.1970x + 0.0801x^2 + 0.2410x^3 + 0.0082x^4 + 0.4736x^{13}, \tag{5}$$

and the second one possesses a constrained DE-optimized variable-node degree distribution given by [12, 9]

$$\lambda_2(x) = 0.1250x + 0.4460x^2 + 0.4078x^{10} + 0.0213x^{11}. \tag{6}$$

They are abbreviated as "DE14" and "CDE12", respectively. The third one is a CSF-LDPC code with a maximum variable-node degree of 20 and is denoted by "CSF20". Details of the aforementioned three codes are listed in Table 1. Moreover, the lengths of the three types of codes are set to 2016.

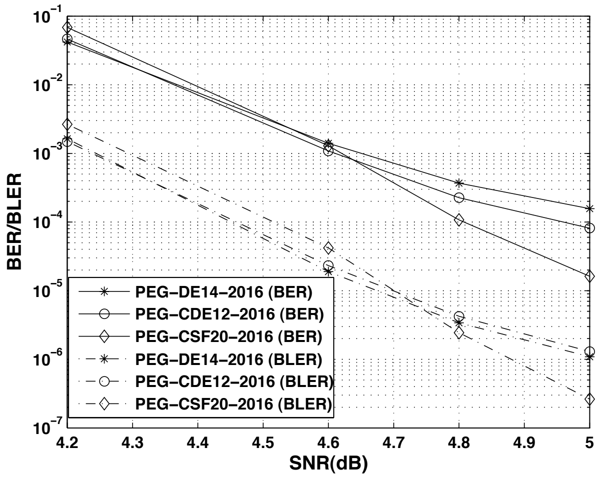

Figure 2: BER and BLER performance of three different LDPC codes — "DE14", "CDE12"and "CSF20". Code lengths are 2016 and the code rate is 0.75.

Figure 2 plots the simulated bit/block error rates (BERs/BLERs) for the three rate-0.75 codes. The performance curves show that the constrained SF-LDPC code "CSF20" suffers slight degradation of BER and BLER performance compared with "DE14" and "CDE12" at low SNR, but outperform them at higher SNR values.

For a given code, define the average path length (APL) of the corresponding bipartite graph as the path length between any two variable nodes averaged over the whole bipartite graph. Consider codes with the same code length and the same average variable node degree. The code graphs with smaller APLs are more efficient in spreading information than those with large APLs. Similarly, under same code length and the same APL, code graphs with smaller average variable-node degree can be regarded as less complex. To measure the merit of a given code, we define the average-path-length-variable-node-degree-product (APVP) of a code as the product of the average path length of the bipartite graph and the average variable-node degree of the code. In general, codes with smaller APVPs are preferred. The APLs and the APVPs of the three types of codes are listed in Table 2. It is observed that even under the constraint in the fraction of degree-2 variable nodes, "CSF20" and "CDE12" have similar APVPs and have lower APVPs compared with "DE14".

To further compare the performance of the codes, we define the metric "average convergence time", denoted by t_c, as the product of the average number of iterations to converge ($\bar{I}$) and the average variable-node degree ($< k_v >$). In general, the smaller the "average convergence time", the

[1] σ^* can be regarded as the maximum noise standard deviation below which error-free communication can always be achieved. To evaluate the threshold value, "density evolution (DE)" is used [1, 2, 11].

Table 3: Comparison of the average convergence times for rate-0.75 codes at high SNR values.

SNR/Code length	5.0 dB/2016	4.8 dB/2016	4.6 dB/2016
Code type	DE14/CDE12/CSF20	DE14/CDE12/CSF20	DE14/CDE12/CSF20
$< k_v >$	4.53/4.00/4.01	4.53/4.00/4.01	4.53/4.00/4.01
$\bar{I}$	5.94/5.69/5.57	6.88/6.63/6.48	8.20/7.97/7.85
$t_c = \bar{I} \times < k_v >$	26.91/22.76/22.34	31.17/26.52/25.98	37.15/31.88/31.48
Normalized t_c	1.20/1.02/1.00	1.20/1.02/1.00	1.18/1.01/1.00

Table 2: Comparison of APVPs of several rate-0.75 codes.

Code	$< k_v >$	APL	APVP
DE14-2016	4.526	2.059	9.319
CDE12-2016	4.000	2.161	8.644
CSF20-2016	4.005	2.154	8.627

less time the decoder takes to decode a codeword. The results in Table 3 indicates that "CDE12" and "CSF20" have almost identical "average convergence time" (t_c) while "DE14" requires, on average, 20% more time (resources) to decode a codeword. The results are consistent with the fact that "CSF20" and "CDE12" have similar APVPs and have smaller APVPs than "DE14". In addition, the table indicates that "CSF20" requires, on average, a slightly smaller number of iterations for decoding compared with the DE-optimized and constrained DE-optimized codes.

5. Conclusions

In this paper, we have proposed a new constraint on the design of scale-free LDPC (SF-LDPC) codes. The constraint limits the proportional of variable nodes with degree 2, and is applicable to the design of high rate codes. Using rate-0.75 codes as an example, we have illustrated that compared with DE-optimized and constrained DE-optimized codes, the constrained SF-LDPC (CSF-LDPC) code can accomplish very similar achievable error performance (threshold), lower bit/block error rate at the high SNR region and require a smaller number of iterations for convergence.

Acknowledgments

The authors would like to thank The Hong Kong Polytechnic University (G-YG34) and the National Science Foundation of China for supporting this research.

References

[1] T.J. Richardson and R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, 2001.

[2] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, 47(2):619–637, 2001.

[3] X. Zheng, F.C.M. Lau, and C.K. Tse. Short-length LDPC codes with power-law distributed variable-node degrees. In *Proc. Int. Symp. on Inform. Theory*, pages 168–171. Budapest, Hungary, Sept. 2008.

[4] X. Zheng, F.C.M. Lau, C.K. Tse, and Yejun He. Application of complex-network theories to the design of short-length LDPC codes. *IET Communications*, to appear.

[5] T. Richardson. Error floors of LDPC codes. In *Proc. 41st Annu. Allerton Conf. on Communication, Control, and Computing*, pages 1426–1435. Urbana-Champaign, USA, 2003.

[6] D.J.C. MacKay and M.S. Postol. Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes. *Electronic Notes in Theoretical Computer Science*, 74:97–104, 2003.

[7] S. Lin and D.J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice Hall, 1983.

[8] X.Y. Hu, E. Eleftheriou, and D.M. Arnold. Regular and irregular progressive edge-growth tanner graphs. *IEEE Trans. Inform. Theory*, 51(1):386–398, 2005.

[9] M. Yang, W.E.Y. Li, and Y. Li. Design of efficiently encodable moderate-length high-rate irregular LDPC codes. *IEEE Trans. Commun.*, 52(4):564–571, 2004.

[10] L. Dinoi, F. Sottile, S. Benedetto, I. Superiore, M. Boella, and I. Torino. Design of variable-rate irregular LDPC codes with low error floor. In *Proc. IEEE Int. Conf. Commun.*, pages 647–651. Seoul, Korea, 2005.

[11] M. Luby, M. Mitzenmacher, A. Shokrollah, and D. Spielman. Analysis of low density codes and improved designs using irregular graphs. In *Proc. the Thirtieth Annual ACM Symposium on Theory of computing*, pages 249–258. New York, USA, 1998.

[12] M. Chiani and A. Ventura. Design and performance evaluation of some high-rate irregular low-density parity-check codes. In *IEEE GLOBECOM*, pages 990–994. Texas,USA, 2001.