2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Short-length LDPC Codes with Power-law Distributed Variable-node Degrees

X. Zheng, Francis C.M. Lau† and Chi K. Tse

Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong,
Website: http://chaos.eie.polyu.edu.hk † Email: encmlau@polyu.edu.hk

Abstract— In the design of low-density-parity-check (LDPC) codes, it has been proven that codes with variable-node and check-node degree distributions optimized by the "Density Evolution" (DE) algorithm can accomplish theoretical error performance very close to the Shannon limit. But the use of the DE algorithm requires two basic assumptions — infinite code length and infinite number of iterations performed by the decoder. Unfortunately, neither requirement can be fulfilled in practice. When the LDPC code has a finite length, say a few thousand symbols, the "DE-optimized" degree distributions may not be the best solutions. In this paper, we propose constructing short-length LDPC codes with variable-node degrees following power-law distributions. We show that the proposed scale-free LDPC (SF-LDPC) codes, when compared with codes constructed with "DE-optimized" degree distributions, can achieve lower complexity (in terms of average number of node degrees), faster convergence time (in terms of number of iterations executed at the decoder) and similar/lower error rates.

1. Introduction

In the design of LDPC codes, the error performance of the codes has been one of the major considerations. It has also been shown that, in general, irregular LDPC codes outperform regular ones in terms of error rate under similar scenarios [1]. Suppose the most common algorithm, namely the sum-product iterative decoding algorithm or the belief propagation (BP) algorithm [1], is used in the decoder. Given the variable-node and check-node degree distributions. The best possible error performance of an LDPC code with such degree distributions can be evaluated from the well-known "density evolution" (DE) mechanism. Therefore, by varying the variable-node and check-node degree distributions, researchers have been able to optimize the achievable error performance of the LDPC codes under different channel conditions [1]–[3].

Three assumptions have been made, however, during the application of DE. First, it is assumed that in the BP decoding algorithm, the updated messages passing forward and backward between the set of variable nodes and the set of check nodes are of analog nature, i.e., the messages can assume the values of any real numbers. Practically, all numbers have to be quantized or discretized for computation by hardware. To evaluate the exact behavior of the discretized BP decoder, "discretized density evolution" (DDE) has been developed [4]. It has been concluded that when the messages are quantized into discrete levels using practical quantizers with 10 or more bits, there is little discrepancy between the results obtained by DE and DDE. The other two assumptions made in the DE algorithm are that the code has an infinite length and iterations are performed at the decoder endlessly. Unfortunately, such assumptions are not realizable in practice. Nonetheless, under the condition that the same degree distributions are used, the achievable error performance of an LDPC code with finite length will approach that of an infinite-length LDPC code asymptotically as the code length increases.

With DDE and code rate $1/2$, the achievable error performance of an LDPC code has been found to lie within 0.0045 dB of the Shannon limit under a binary-input additive white Gaussian noise (AWGN) channel [4]. Simulations have also shown that within 0.04 dB of the Shannon limit, a bit error rate of 10^{-6} can be achieved with a code length of 10^7 and about 800 to 1100 iterations [4]. Yet, code lengths larger than 10^6 are not very practical for many applications because of the huge hardware complexity involved and the long time taken to conduct one iteration — not to mention the time delay incurred if 1000 iterations are required to decode one codeword. In reality, short-length (less than several thousands) LDPC codes will find a lot more applications, but as mentioned above, their error performance may deviate significantly from the best values. So, is "DE-optimized" degree distributions the best choice when the code length is comparatively short?

In recent years, complex networks have been studied across many fields of science, including computer networks, biological networks and social networks [5, 6]. Complex networks, very much like graphs, are structures consisting of nodes interconnected by edges. Among the various types of complex networks, scale-free networks are characterized by their node degrees following power-law distributions. Compared with regular coupled networks, small-world networks, and random networks, scale-free networks of the same size (number of nodes) and with the same number of connections are found to accomplish the shortest average path length [7]. That is, the average distance between any two nodes is the shortest for networks with a scale-free property. Recall that in the BP decoding algorithm, updated messages are passed forward and backward between the set of variable nodes and the set of check nodes iteratively. If the messages can be conveyed from one node to all other nodes more efficiently, the number of

iterations it takes for the decoding algorithm to converge should be lower, particularly in the high signal-to-noise-ratio (SNR) region where the algorithm is most likely to converge.

In this paper, we exploit the "shortest average path length" property of scale-free networks and apply it to the design of short-length LDPC codes, in which "DE-optimized" degree distributions may not be the best solutions. Specifically, we will propose constructing short-length LDPC codes with variable-node degrees following power-law distributions. Here, we refer such LDPC codes to as *scale-free LDPC* (SF-LDPC) codes. We will compare the achievable error performance (threshold) and the complexity (in terms of average number of node degrees) between the proposed short-length SF-LDPC codes and "DE-optimized" LDPC codes. Moreover, we will construct SF-LDPC codes with length 1008 and 504 using the popular progressive-edge-growth (PEG) technique [8]. Finally, we will compare the error rates and the convergence time (in terms of number of iterations executed at the decoder) between the constructed SF-LDPC codes and some other best-known LDPC codes under an AWGN environment.

2. Density Evolution

In the bipartite graph representation of LDPC codes, for each node, the number of edges connected is called the "degree" of the node. For a given distribution pair (λ, ρ) of an LDPC ensemble, $\lambda(x) := \sum_{k=2}^{d_v} \lambda_k x^{k-1}$ and $\rho(x) := \sum_{k=2}^{d_c} \rho_k x^{k-1}$ specifies, respectively, the variable-node and check-node degree distributions. Also, d_v is the maximum variable-node degree and d_c denotes the maximum check-node degree. Moreover, the coefficients λ_k and ρ_k, respectively, represent the fraction of edges connected to the variable and check nodes with degree k. Based on the degree distributions, the code rate of the system, denoted by R, can be obtained using

$$R = 1 - \int_0^1 \rho(x)dx / \int_0^1 \lambda(x)dx. \tag{1}$$

In an LDPC code, each coded bit, denoted by $b \in \{0, 1\}$, is represented by a signal with amplitude $a = (-1)^b$ for transmission through the channel. Without loss of generality, suppose an all-zero codeword has been sent. At the receiving side, assume that the transmitted signal is corrupted by independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance (noise power) denoted by σ^2. Given an LDPC code that is corrupted by noise. The message-passing algorithm employed by the LDPC decoders allows the exchange of information between the variable nodes and check nodes during each iteration. Moreover, the density evolution algorithm enables us to track the density of the messages. In the following, we summarize the main steps in the density evolution algorithm. For details, please refer to [9].

Suppose a belief propagation algorithm based on the log-likelihood ratio (LLR) is used in the iterative decoder [1]. In the algorithm, each of the LLR variables passing along the edges connecting the variable nodes and check nodes is a message. During the lth iteration, it is denoted by $m^{(l)}$. More specifically, the message passing from variable node v to check node c is represented by $m_{vc}^{(l)}$ and that from check node c to variable node v by $m_{cv}^{(l)}$. The value of the message is given by

$$m = \ln \frac{\Pr(a = 1|\, y)}{\Pr(a = -1|\, y)} = \ln \frac{p(y\,|a = 1)}{p(y\,|a = -1)} \tag{2}$$

where a is a random variable (RV) describing the bit value of variable node v, and y is a RV representing all the information integrated into the message. Moreover, $\Pr(\cdot)$ and $p(\cdot)$ denote the probability and the probability density function, respectively.

Based upon the received signal, the initial message at variable node v, denoted by m_0, can be computed. Then the messages along the edges can be updated iteratively as follows [9]:

$$m_{vc}^{(l)} = \begin{cases} m_0 & \text{for } l = 0 \\ m_0 + \sum_{c' \in C_v \setminus \{c\}} m_{c'v}^{(l)} & \text{for } l > 0 \end{cases} \tag{3}$$

$$m_{cv}^{(l)} = \prod_{v' \in V_c \setminus \{v\}} \operatorname{sgn}(m_{v'c}^{(l-1)}) \sum_{v' \in V_c \setminus \{v\}} \ln\left(\tanh\left|\frac{m_{v'c}^{(l-1)}}{2}\right|\right). \tag{4}$$

In the above two equations, C_v denotes the set of check nodes connected to variable node v whereas V_c represents the set of variable nodes linked to check node c. Moreover, the unconventional probabilistic definition of the sign function $\operatorname{sgn}(x)$ is given by

$$\operatorname{sgn}(x) = \begin{cases} 0 & \text{for } x > 0 \\ 0 & \text{with probability } 1/2 \text{ for } x = 0 \\ 1 & \text{with probability } 1/2 \text{ for } x = 0 \\ 1 & \text{for } x < 0 \end{cases}. \tag{5}$$

Denote the density associated with the messages from the variables nodes to the check nodes during the lth iteration by P_l and that from the check nodes to the variable nodes by Q_l. Moreover, the initial message density has been shown equal to [9]

$$P_0(y) = \frac{\sigma}{\sqrt{8\pi}} \exp\left[\frac{-\left(y - \frac{2}{\sigma^2}\right)^2 \sigma^2}{8}\right]. \tag{6}$$

Assuming that all the messages at each node are independent of one another, it has been shown that [9]

$$P_l = P_0 \otimes \lambda(Q_l) \tag{7}$$

where $\otimes$ represents convolution.

To compute Q_l, we represent the messages $m_{vc}^{(l)}$ in an alternative way. We define the map $\beta : [-\infty, +\infty] \to \mathrm{GF}(2) \times [0, +\infty]$. Given a RV $z \in [-\infty, +\infty]$ with distribution F_z and $z \neq 0$. Let

$$\beta(z) := (\beta_1(z), \beta_2(z)) := (\operatorname{sgn}(z), -\ln(\tanh|z/2|)). \tag{8}$$

We also define the "distribution" of $\beta(z)$ as

$$\Gamma(F_z)(s, x) = \chi_{\{s=0\}}\Gamma_0(F_z)(x) + \chi_{\{s=1\}}\Gamma_1(F_z)(x) \tag{9}$$

Table 1: Comparison of threshold value and average number of connections between SF-LDPC codes and other best-known LDPC codes.

Common Parameters		Optimized Codes in [10]		Optimized SF-LDPC Codes		
Range of Variable-node Degrees	Range of Check-node Degrees	σ^*	$< k >$	γ	σ^*	$< k >$
2 to 15	7 to 9	0.9622	4.0087	2.35	0.9430	3.5117
2 to 20	7 to 9	0.9649	4.1638	2.35	0.9449	3.7192
2 to 30	7 to 9	0.9690	4.4963	2.36	0.9513	3.9723
2 to 50	7 to 9	0.9718	5.0765	2.35	0.9535	4.3039

Table 2: Details of LDPC code types used in simulations.

Abbreviation	Type of Code	Range of Variable-node Degrees	Range of Check-node Degrees	σ^*	$< k >$
PEG-DE10	DE-optimized Codes in [10]	2 to 10	6 to 7	0.9558	3.6631
MacKay-DE15	DE-optimized Codes in [12]	2 to 15	7 to 9	0.9622	4.0087
PEG-SF20	Optimized SF-LDPC Codes	2 to 20	7 to 9	0.9449	3.7192

where

$$\begin{aligned}\Gamma_0(F_z)(x) &= \Pr\{\beta_1(z)=0, \beta_2(z)\le x\} \\ &= \Pr\{z \ge -\ln\tanh(x/2)\} \qquad (10)\\ \Gamma_1(F_z)(x) &= \Pr\{\beta_1(z)=1, \beta_2(z)\le x\} \\ &= \Pr\{z \le \ln\tanh(x/2)\} \qquad (11)\end{aligned}$$

and $\chi_{\{s=a\}}$ equals 1 if $s = a$ and 0 otherwise. Thus, (4) can be written as

$$m_{cv}^{(l)} = \beta^{-1}\left(\sum_{v'\in V_c\setminus\{v\}} \beta\left(\frac{m_{v'c}^{(l-1)}}{2}\right)\right). \qquad (12)$$

and the density of $m_{cv}^{(l)}$ is given by

$$Q_l = \Gamma^{-1}(\rho(\Gamma(P_{l-1}))). \qquad (13)$$

Finally, P_l can be expressed as [9]

$$P_l = P_0 \otimes \lambda(\Gamma^{-1}(\rho(\Gamma(P_{l-1})))). \qquad (14)$$

Having found the density P_l, the associated distribution can be computed using integration. Thus, the above iterative process can determine whether the expected fraction of incorrect message will go to zero when the number of iterations increases. For fixed d_v and d_c, one then can find the largest value of σ, i.e., best achievable performance or threshold of the LDPC code, by varying the distribution pair (λ, ρ) such that the expected fraction of incorrect message will go to zero. Optimization of the threshold has been carried out and the degree distributions of some good codes have already been found [10]. However, such codes provide optimal error performance only under an infinite code length and an infinite number of iterations for decoding.

3. Scale-Free LDPC (SF-LDPC) Codes

Scale-free networks [6] are characterized by power-law degree distributions. In other words, $\Pr(k) = Ak^{\gamma}$, where $\Pr(k)$ denotes the probability of a randomly selected node having a degree k, γ is the characteristic exponent and A is the normalizing coefficient. It has also been shown that when the value of the characteristic exponent, i.e., γ, lies between 2 and 3, the average path length of scale-free networks is $O(\log(\log(N)))$ [7]. Furthermore, it has been proven that if the degree distribution of one set of nodes in a bipartite graph follows a power-law distribution, the degree distribution of the unipartite graph (network) formed when the other set of nodes is removed, also follows a power-law with the same exponent [11]. Because of the aforementioned reasons, it is natural to expect that LDPC codes with variable-node degree distributions following power laws (termed as *scale-free LDPC codes* subsequently) should have faster convergence rates because messages can be exchanged more efficiently among the variable nodes and the check nodes.

To construct a scale-free LDPC (SF-LDPC) code, we assign the fraction of variable nodes with degree k, denoted by $\Pr_\lambda(k)$, according to a power-law function, i.e., $\Pr_\lambda(k) \propto k^{-\gamma}$. To minimize the number of parameters in the optimization process, we restrict the check-node degrees to three consecutive integers, i.e., $d_c - 2, d_c - 1$ and d_c. Moreover, the fraction of check nodes of degree k, denoted by $\Pr_\rho(k)$, is assumed to be

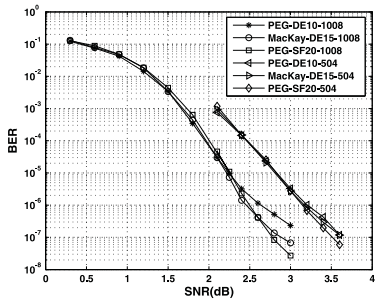
$$\Pr_\rho(k) = \frac{\mu^k e^{-\mu}}{k!}. \qquad (15)$$

Finally, given the maximum variable-node degree d_v and code rate R. We can alter the values of γ and μ in order to optimize the achievable error performance of SF-LDPC codes.

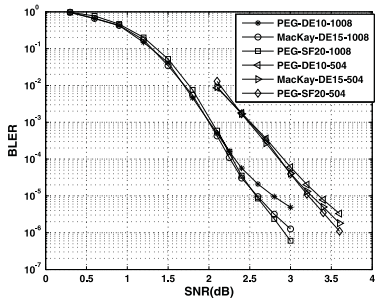
4. Results

First, we compare the achievable error-correcting capability (threshold) between SF-LDPC codes and other best-known LDPC codes [10]. We assume a code rate of 0.5 and an AWGN channel, at which the Shannon limit equals 0.9787 [10]. Table 1 presents the parameters used and the results. It can be observed that in all cases, the optimized threshold values σ^* for the SF-LDPC codes are comparable with those for other best-known LDPC codes (less than 2% difference). But the average number of connections for the SF-LDPC codes is significantly smaller (12 to 15% reduction) compared to those for other LDPC codes.

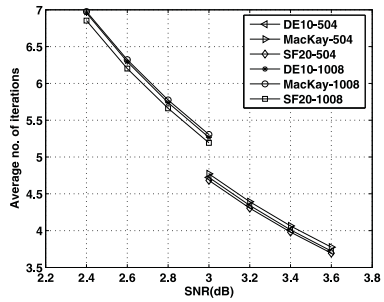
Next, we compare the simulation results of three different types of LDPC codes. Details of the codes are given in Table 2. The first two code types, abbreviated by "PEG-DE10" and "MacKay-DE15", are LDPC codes



(a)



(b)



(c)

Figure 1: Performance of three different types of LDPC codes — 'PEG-DE10", "MacKay-DE15"and "PEG-SF20". Code lengths are 1008 and 504 while the code rate is 0.5. (a) Bit error rate; (b) block error rate; (c) average number of iterations per codeword.

with variable-node degree distributions optimized by the DE algorithm [10, 12]. The third code, abbreviated by "PEG-SF20", is an optimized SF-LDPC code. The variable nodes and the check nodes for "PEG-DE10" and "PEG-SF20" codes are connected using the progressive-edge-graph (PEG) method [8], which has been shown to produce codes with both large girth and large Hamming distance. For the codes denoted by "MacKay-DE15", we directly apply the codes constructed in [12], which are the best known LDPC codes in terms of error performance that possess the properties listed in Table 2. Two different code lengths are used — 1008 and 504, while the code rate is remained at 0.5. The maximum number of iterations performed to decode one codeword is limited to 50 and the decoding process will be terminated once the maximum number is reached. In Fig. 1(a) and (b), we plot the simulated bit error rates (BERs) and block error rates (BLERs), respectively, for the three types of codes under study. It can be observed that the optimized SF-LDPC codes provide similar BER and BLER performance as "PEG-DE10" and "MacKay-DE15" codes at low SNR, and outperform them at higher SNR values. Figure 1(c) also depicts that optimized SF-LDPC codes can be decoded with a smaller number of iterations, on average, compared with the other DE optimized codes.

5. Conclusion

In this paper, we have proposed LDPC codes with variable-node degrees following power-law distributions. Theoretical results indicate that under the same conditions, the threshold value of the proposed LDPC codes (called scale-free LDPC or SF-LDPC codes) is slightly less than that of the best-known LDPC codes. But the average number of connections for SF-LDPC codes is significantly smaller. We have also constructed SF-LDPC codes of lengths 1008 and 504, with rate 0.5. We conclude that the bit error rate and block error rate performances of the SF-LDPC are comparable to and sometimes better than some of the best-known codes. Moreover, the SF-LDPC codes take fewer iterations to converge.

Acknowledgments

This work was supported by a research grant (G-YG34) provided by Hong Kong Polytechnic University.

References

[1] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[2] M. Luby, M. Mitzenmacher, A. Shokrollahi and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," *Proc., ACM Symposium on Theory of Computing*, Dallas, USA, May 1998, pp. 249–258.

[3] S.-Y. Chung, T. J. Richardson and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.

[4] S.-Y. Chung, G. D. Forney, T. J. Richardson and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commu. Lett.,* vol. 5, no. 2, pp. 58–60, 2001.

[5] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, June 1998.

[6] Albert-L*á*szl*ó* Barab*á*si and R*é*ka Albert, "Emergence of scaling in random networks," *Science,* vol. 286, no. 5439, pp. 509–512, 1999.

[7] R. Cohen and S. Havlin, "Scale-free networks are ultrasmall," *Phy. Rev. L*, vol. 90, 058701, Feb. 2003.

[8] X. Y. Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inform. Theory,* vol. 51, no. 1, pp. 386–398, 2005.

[9] T. J. Richardson, M. A. Shrokrollahi and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[10] C. B. Schlegel, L. c. Perez, *Trellis and turbo coding*, Wiley-IEEE Press, March 2004.

[11] J.-L. Guillaume, M. Latapy, *A realistic model for complex networks*, available from: http://arxiv.org/abs/cond-mat/0307095.

[12] D. J. C, MacKay, *Encyclopedia of sparse graph codes*, available from: http://wol.ra.phy.cam.ac.uk/mackay/codes/data.html.