LETTER

Lowering Error Floors of Irregular LDPC Codes by Combining Construction and Decoding

Xiaopeng JIAO[†], Jianjun MU[†a)], Fan FANG[†], *Nonmembers, and* Rong SUN[††], *Member*

SUMMARY Irregular low-density parity-check (LDPC) codes generally have good decoding performance in the waterfall region, but they exhibit higher error floors than regular ones. In this letter, we present a hybrid method, which combines code construction and the iterative decoding algorithm, to tackle this problem. Simulation results show that the proposed scheme decreases the error floor significantly for irregular LDPC codes over binary-input additive white Gaussian noise (BIAWGN) channel.

key words: *error floor, low-density parity-check (LDPC) codes, minimum distance, trapping sets*

1. Introduction

Carefully designed irregular low-density parity-check (LDPC) codes can achieve better capacity-approaching performance than regular LDPC codes under iterative belief propagation (BP) decoding. With finite block lengths, irregular LDPC codes generally have good decoding performance in the waterfall region, but they exhibit higher error floors than regular ones. This is mainly caused by the following two reasons: small trapping set induced errors [1] and undetected errors induced by weak minimum distance property of the irregular codes.

The techniques that can lower the error floors of LDPC codes mainly fall into three categories: (1) Construct LDPC codes that are free of small error events, such as cycles, stopping sets and trapping sets [2]–[4]. For example, in [4] the authors proposed an efficient method to eliminate trapping sets by using Tanner graph covers. (2) Modify the message passing decoders. According to the requirement that whether the small trapping sets are identified or not, we can divide this method into two kinds: (I) Trapping set dependent modification, such as look-up table method [5], bi-mode and bit-pinning decoder [6], on-off attenuated method [7], and trapping set neutralization method [8]. (II) Trapping set independent modification, such as average decoding [9], two-stage decoding [10], and selective biasing method [11]. (3) Add an outer algebraic code such as RS code or BCH code [6], [7]. Since trapping sets generally have complicated combinatorial property, it is difficult to enumerate them by mathematical methods or computer programmes. Therefore, coping with trapping sets directly during the code construction or decoding to lower the error floors is difficult. Thus, the trapping set independent scheme is more appealing and more applicable.

In this letter, we focus on lowering the error floors of irregular LDPC codes. A hybrid method, which combines code construction and iterative decoding to combat the error-floor problem, is presented. Firstly, in order to reduce the undetected errors, we propose a method to improve the minimum distance of the LDPC code. Secondly, a postprocessing scheme is proposed to modify the iterative decoder to alleviate decoding errors induced by trapping sets with size smaller than the minimum distance. Thus, we can obtain an irregular LDPC code with performance approaching its asymptotic error performance bound calculated by the weight distribution of the code. The remainder of this letter is organized as follows. In Sect. 2, distance set, trapping set and asymptotic error performance bound are reviewed. Section 3 explains the detailed scheme to lower the error floors of irregular LDPC codes. Simulation results for the proposed method are given in Sect. 4. Finally, Sect. 5 concludes the letter.

2. Distance Set, Trapping Set and Asymptotic Error Performance Bound

Definition 1 [12]: A *distance set* $\mathcal{D}$ is a subset of variable node set $\mathcal{V}$, such that all neighbors of the variable nodes in $\mathcal{D}$ are connected to $\mathcal{D}$ an even number of times*.

Note that there exists a one-to-one correspondence between the distance sets and the codewords. A distance set with cardinality s represents a codeword with Hamming weight s. Moreover, a distance set is also a stopping set.

Definition 2 [1]: An (a, b) *trapping set* $\mathcal{T}$ is a set of a variable nodes, for which the induced subgraph of the a variable nodes and their neighborhood check nodes contains exactly b odd-degree check nodes.

Definition 3 [10]: For an (a, b) trapping set, if the code bits associated with the a variable nodes are all the wrong bits, then the check-sums corresponding to the b odd-degree check nodes will not be satisfied. These b check nodes are called *unsatisfied check nodes.*

*Note that the indices of a distance set are indeed the support set of a codeword. However, in terms of Tanner graph, we use distance set instead of support set.

Manuscript received February 15, 2011.
Manuscript revised July 31, 2011.

[†]The authors are with the School of Computer Science and Technology, Xidian University, Xi'an, 710071, P.R. China. They are also with Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu, 610031, P.R. China.
[††]The author is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, 710071, P.R. China.
a) E-mail: jjmu@xidian.edu.cn

DOI: 10.1587/transcom.E95.B.1

The dominant trapping sets, which largely contribute to the BP decoding performance in the error floor region, both have small values of a and b. Given the weight distribution of a code, the asymptotic frame error rate (FER) over additive white Gaussian noise (AWGN) channel with maximum-likelihood decoding can be estimated by

$$P_F(E_b/N_0) \approx \frac{1}{2}\sum_{i=d_{min}}^{d_{min}+\upsilon} w_i \cdot erfc\left(\sqrt{i \cdot R \cdot \frac{E_b}{N_0}}\right), \qquad (1)$$

where w_i denotes the number of codewords with Hamming weight i, d_{min} is the minimum distance, R is the code rate, υ is a non-negative small integer value and $erfc(x)$ is the complementary error function: $erfc(x) = 2/\sqrt{\pi}\int_x^\infty e^{-t^2}dt$.

3. Error Floor Performance Improvement

The dominant error events in the error floor region is the existence of the small trapping sets in the LDPC codes. However, if the trapping set induced errors are attacked by some methods, the low weight codewords in the irregular LDPC codes will dominate the error events. Thus, to mitigate the error floor to a relatively lower level, these two factors should be considered simultaneously.

3.1 Minimum Distance Improvement

In this subsection, we consider the minimum distance improvement of a given code. Since there is a one-to-one correspondence between the distance sets and codewords, the minimum distance of a code can be improved by removing small distance sets in the code's Tanner graphs. In [13] the authors presented a simple algorithm to eliminate small stopping sets in irregular Tanner graphs. The similar method can also be used to eliminate small distance sets.

By adding new check nodes and having their edges connected to the distance sets that need to be removed, no new distance set is introduced in the code's Tanner graph and the original distance sets are eliminated. To illustrate this, one can consider the distance set $\mathcal{D}$ in Fig. 1(a). c_a is a new check node added to the code graph, and it has an edge connected to a variable node in $\mathcal{D}$. It can be easily seen that, the original codeword corresponding to $\mathcal{D}$ is not a codeword after adding c_a since the check equation of c_a is not satisfied. Indeed, the original distance set $\mathcal{D}$ is transformed to a $(|\mathcal{D}|, 1)$ trapping set. Moreover, newly added check nodes and edges do not affect the code's Tanner graph except the distance sets, so it introduces no new distance set. With simulations, we found that many distance sets have common variable nodes. Thus, several distance sets can be removed by adding only one edge. Figure 1(b) shows that four distance sets with a common variable node υ can be removed simultaneously by adding only one edge between c_a and υ. This observation can be used to design near-optimal algorithm to add new check nodes and edges.

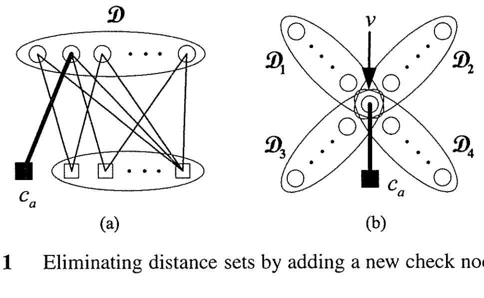

Fig. 1 Eliminating distance sets by adding a new check node c_a.

Keep this idea in mind, we can design a greedy algorithm like [13] to eliminate as many distance sets as possible. The algorithms presented in [12] and [14] are used to enumerate the small distance sets in a code's Tanner graph. For short length LDPC codes, we use the algorithm in [14] which can list all small distance sets. For medium length LDPC codes, the algorithm in [14] is quite complex, so we use the algorithm in [12] to list most of the small distance sets with manageable complexity.

Remarks: (1) The minimum distance improvement by adding new check nodes is also presented in [3] (adding independent rows in the code's parity check matrix), but the edge selection method presented in this letter is different. (2) To improve the minimum distance, we need to find the small distance sets in the Tanner graphs (equivalent to find a subset of small stopping sets). Fortunately for us, finding small stopping sets is much easier than finding small trapping sets.

3.2 Trapping Set Induced Errors Improvement

Trapping sets with size smaller than the minimum distance of a code will dominate the error floor performance. Thus, a method must be introduced to attack these trapping set induced errors. To address this problem, a modification of the two-stage BP (TS-BP) decoding method [10] is proposed.

In the first stage, the conventional iterative decoding is performed. If the decoding fails and the smallest number of the unsatisfied check nodes in the iterative process is less than the minimum distance d_{min}, then the second stage is done. The main idea is to find a few error bits based on the unsatisfied check nodes. Then flip these error bits by setting their initial log-likelihood-ratios to the maximum possible value with opposite signs, and perform iterative decoding again. The TS-BP decoder improves the error performance significantly, especially in high SNR regions. In [10], the authors showed that flip only two error bits can get good performance gains. Definitely, we do not know which bits are in error in advance, so we should try all bits connected with the unsatisfied check nodes.

With simulations, we find that when the number of the unsatisfied check nodes is exactly one, the TS-BP decoder may fail to converge to right codewords. This is mainly because that only flipping one error bit may not be sufficient to correct all the error bits in the trapping set. To address this problem, we find another error bit to be flipped by the following method. For the error bit υ connected with the unsatisfied check node c (see Fig. 2), all the neighborhood check

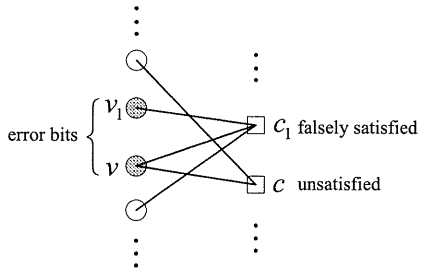

Fig. 2 Subgraph of finding another error bit when the number of the unsatisfied check nodes is exactly one.

nodes connected with v are falsely satisfied except c. Pick a check node c_1 randomly from these falsely satisfied check nodes. There is at least one neighborhood variable node v_1 (not v) connected with c_1 is in error. Finally, flip error bits v, v_1 and perform iterative decoding again. For the degree distribution optimized by density evolution, there is no variable node with degree one. Thus, we can always find the second error bit when the the number of the unsatisfied check nodes is exactly one. Simulations show that this method can correct many error patterns which can not be corrected by the TS-BP decoder, especially in high SNR regions.

The complexity of the modified and the original TS-BP algorithm is the same when the number of the unsatisfied check nodes is greater than one. Therefore, we only compare the complexity of the two algorithms when the number of the unsatisfied check nodes is exactly one. Let I_{max} be the maximum number of iterations, and d_c and d_{c_1} be the degree of c and c_1 respectively. For the original TS-BP algorithm, the number of iteration is upper bounded by $d_c I_{max}$. While for the modified TS-BP algorithm, the number is upper bounded by $d_c d_{c_1} I_{max}$. For the whole decoding process, the increase of the complexity by the modified TS-BP algorithm is negligible.

4. Simulation Results

In this section, we do some simulations to illustrate the efficiency of the minimum distance improvement and the performance gains of the proposed hybrid method. As we know, PEG algorithm [2] is one of the most effective construction method to generate irregular LDPC codes. Thus, we use the PEG codes as the base codes to start our improvement. For all of our simulations, binary phase shift keying (BPSK) transmission over an AWGN channel is assumed. The maximum number of iterations for the BP decoder is set to 50. All the asymptotic curves in the following figures are plotted according to equation (1). The variable node degree distribution [15] used in this section is: $\lambda(x) = 0.25105x + 0.30938x^2 + 0.00104x^3 + 0.43853x^9$.

Firstly, consider an LDPC code C_1 constructed by PEG algorithm with a block length of 200 and rate of 0.5. The exact weight distribution of C_1 calculated by [14] is: $w_1(x) = 1 + x^8 + 3x^{11} + 10x^{12} + 20x^{13} + 41x^{14} + \cdots$. C_2 is obtained by adding one check node (with a degree of 11) to C_1 using the method proposed in Sect. 3.1. The minimum

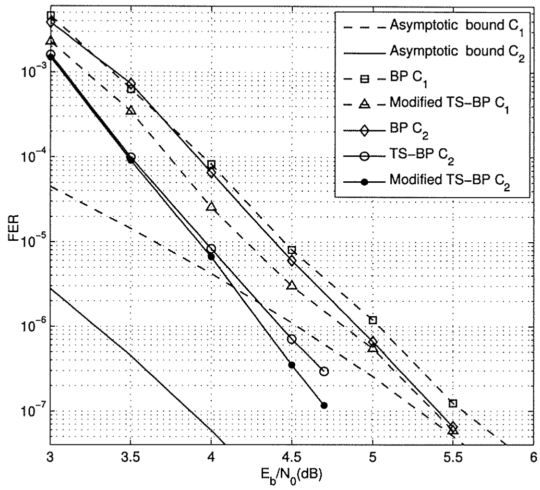

Fig. 3 FER performance of C_1 and C_2 under BP decoder and TS-BP decoder over AWGN channel.

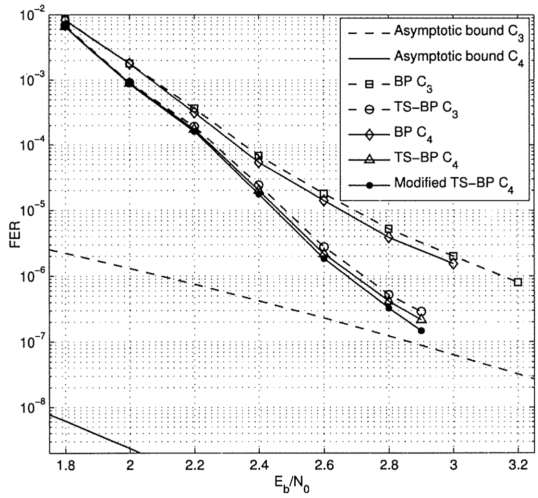

Fig. 4 FER performance of C_3 and C_4 under BP decoder and TS-BP decoder over AWGN channel.

distance of C_2 is exactly 14 and its weight distribution is $w_2(x) = 1 + 21x^{14} + 66x^{15} + \cdots$. The minimum distance is improved from 8 to 14 by the proposed method, while the rate loss is only 0.005. Figure 3 shows the FER performance of C_1 and C_2 under the BP decoder and the TS-BP decoder. From Fig. 3 we can see that: (1) There are only 0.1 dB performance gains at FER 10^{-7} for the minimum distance improvement of C_1. Thus, the dominant error events in high SNR regions are small trapping sets. (2) By modifying the decoder while not improving the code itself, the decoding performance can not achieve low regions below the asymptotic performance bound of C_1. (3) The modified TS-BP decoder is slightly better than the TS-BP decoder in high SNR regions. (4) By utilizing the hybrid method, which combines code construction and the decoder modification, the performance of C_2 outperforms the asymptotic performance bound of C_1. At FER 10^{-7}, there are 0.8 dB performance gains for the modified TS-BP decoder of C_2 when compared to the BP decoder of C_1.

Table 1 The number of E occurs and the number they are corrected by TS-BP and modified TS-BP decoder for C_2 and C_4.

Code C_2	3.5 dB	4.0 dB	4.5 dB	4.7 dB
$\#E$	47	105	253	333
#corrected by TS-BP	33	78	194	255
#corrected by modified TS-BP	42	96	246	331
Code C_4	2.4 dB	2.6 dB	2.8 dB	2.9 dB
$\#E$	16	58	89	52
#corrected by TS-BP	12	38	62	40
#corrected by modified TS-BP	13	50	84	51

Secondly, an LDPC code C_3 is constructed by PEG algorithm with a block length of 1000 and rate of 0.5. The approximate weight distribution of C_3 calculated by [12] is: $w_3(x) = 1 + x^{14} + x^{19} + 3x^{20} + 4x^{21} + 5x^{22} + 11x^{23} + \cdots$. By adding two check nodes (with degrees of 7 and 8) to C_3, C_4 has an estimated minimum distance of 25 and its weight distribution is $w_4(x) = 1 + 2x^{25} + 17x^{26} + 20x^{27} + \cdots$. Note that the exact minimum distance of C_4 may be less than 25[†]. Figure 4 shows the FER performance of C_3 and C_4 under BP, TS-BP and the modified TS-BP decoder. It can be seen that the performance improvement of our proposed hybrid method is significant when compared to the BP decoder. We can also note that the performance improvement of C_4 under modified TS-BP decoder is not evident when compared to TS-BP decoder. It is because that we can not proceed to the high SNR regions due to the long time needed for the Monte Carlo simulation. However, it is expected that the performance gap will become large between our proposed decoder and the TS-BP decoder when SNR increases.

Let E denote the event that the number of the unsatisfied check nodes is exactly one when the BP decoder fails. Table 1 lists the number of E occurs and the number they are corrected by the TS-BP and modified TS-BP decoder for codes C_2 and C_4, respectively[††]. It can be seen that a large percentage of errors induced by E for the TS-BP decoder can be corrected by the modified TS-BP decoder. It should be noted that the performance improvement on the modified TS-BP decoder over the TS-BP decoder is invisible in low SNR regions and becomes more obvious in high SNR regions. It is because that the error event E is more dominate in high SNR regions for the TS-BP decoder.

5. Conclusion

In this letter, a hybrid method to lower the error floors of irregular LDPC codes has been proposed. By eliminating small distance sets in the original code, the influence of the low-weight codewords to the decoding performance is mitigated. Then the modified TS-BP decoder is used to improve the decoding performance in the presence of small trapping sets. Simulations show that the proposed method reduce the error floor significantly for the irregular LDPC codes.

[†]Note that a more recently proposed probabilistic method [16] can be used to compute the weight distribution of the relative long LDPC codes, though we do not investigate it.

[††]For C_2, we collect 100, 100, 50, 50 error frames at 3.5 dB, 4.0 dB, 4.5 dB and 4.7 dB, respectively. For C_4, we collect 100, 100, 50, 20 error frames at 2.4 dB, 2.6 dB, 2.8 dB and 2.9 dB, respectively.

Acknowledgments

This research is supported by the National Nature Science Foundation of China (Grant Nos. 61001132 and 61001131), the open research fund of Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University (No.2010-03), and the National Basic Research Program (973) of China (Grant No. 2010CB328300).

References

[1] T.J. Richardson, "Error floors of LDPC codes," Proc. 41st Annu. Allerton Conf. Commun., Contr. Comput., 2003.
[2] X.Y. Hu, E. Eleftheriou, and D.M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," IEEE Trans. Inf. Theory, vol.51, no.1, pp.386–398, Jan. 2005.
[3] O. Fainzilber, E. Sharon, and S. Litsyn, "Decreasing error floor in LDPC codes by parity-check matrix extensions," Proc. IEEE Intern. Symp. Inf. Theory, pp.374–378, Seoul, Korea, June 2009.
[4] M. Ivkovic, S.K. Chilappagari, and B. Vasic, "Eliminating trapping sets in low-density parity-check codes by using Tanner graph covers," IEEE Trans. Inf. Theory, vol.54, no.8, pp.3763–3768, Aug. 2008.
[5] E. Cavus and B. Daneshrad, "A performance improvement and error floor avoidance technique for belief propagation decoding of LDPC codes," Proc. 16th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun., pp.3763–3768, Berlin, Germany, Sept. 2005.
[6] Y. Han and W.E. Ryan, "Low-floor decoders for LDPC codes," IEEE Trans. Commun., vol.57, no.6, pp.1663–1673, June 2009.
[7] Y.F. Zhang and W.E. Ryan, "Toward low LDPC-code floors: A case study," IEEE Trans. Commun., vol.57, no.6, pp.1566–1573, June 2009.
[8] E. Alghonaim, A. El-Maleh, and M.A. Landolsi, "New technique for improving performance of LDPC codes in the presence of trapping sets," EURASIP Journal on Wireless Communications and Networking, vol.2008, Article ID 362891.
[9] S. Laendner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," Proc. 2005 IEEE Int. Conf. on Wireless Networks, Commun. and Mobile Computing, pp.630–635, Hawaii, USA, June 2005.
[10] J. Kang, L. Zhang, Z. Ding, and S. Lin, "A two-stage iterative decoding of LDPC codes for lowering error floors," Proc. IEEE Global Commun. Conf., pp.1–4, New Orleans, USA, Dec. 2008.
[11] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M.J. Wainwright, "Lowering LDPC error floors by postprocessing," Proc. IEEE Global Commun. Conf., pp.1–6, New Orleans, USA, Dec. 2008.
[12] G. Richter, "Finding small stopping sets in the Tanner graphs of LDPC codes," Proc. 4th Intern. Symp. Turbo Codes and Related Topics, Munich, Germany, April 2006.
[13] X. Jiao, J. Mu, J. Song, and L. Zhou, "Eliminating small stopping sets in irregular low-density parity-check codes," IEEE Commun. Lett., vol.13, no.6, pp.435–437, June 2009.
[14] E. Rosnes and Ø. Ytrehus, "An efficient algorithm to find all small-size stopping sets of low-density parity-check matrices," IEEE Trans. Inf. Theory, vol.55, no.9, pp.4167–4178, Sept. 2009.
[15] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," IEEE Trans. Inf. Theory, vol.47, no.2, pp.619–637, Feb. 2001.
[16] M. Hirotomo, M. Mohri, and M. Morii, "A probabilistic algorithm for computing the weight distribution of LDPC codes," IEICE Trans. Fundamentals, vol.E92-A, no.7, pp.1677–1689, July 2009.