LOWERING THE ERROR FLOOR OF LDPC CODES BASED ON BIT-RECTIFICATION STRATEGY

Hui Xue[1*], Lijun Zhang[1]

[1]School of Electronics and Information Engineering,Beijing Jiaotong University,Beijing, China
**17120194@bjtu.edu.cn*

Keywords: ERROR FLOOR, BIT-SEARCHING, PARALLEL CONCATENATED

Abstract

This paper proposes a bit-rectification strategy to lower error floor of Low-density parity-check codes. Most erroneous bits are located by bit-searching algorithm and obtain more reliable message over parallel concatenated decoder. The negative effect of erroneous bits along with corresponding trapping sets will be avoided.

1 Introduction

With the rapid development of wireless communication systems, Low-density parity-check (LDPC) codes [1] with the ability of approaching the Shannon limit have become the focus of the channel coding field. Due to the excellent bit error rate (BER) performance of the LDPC code under the iterative decoding algorithm based on belief propagation theory, it has been incorporated into the coding scheme of the fifth generation communication system. However, most LDPC codes have a serious error floor [2] phenomenon in the iterative decoder. That is, as the signal-to-noise ratio (SNR) increases, the BER reaches a saturated state and does not continue to decrease. This problem seriously limits its application of the field with higher BER requirement. In summary, reducing the error floor of LDPC codes is an important and meaningful research.

There has been many research aiming to optimize the error floor phenomenon, most of them aim at breaking the trapping set [3], which is the main cause of error floor. The existing research can be roughly divided into three categories: code construction [4][5], optimization of decoding scheme [6][7] and concatenated strategy [8][9]. The main motivation of code construction is to avoid generating harmful trapping sets in the Tanner graph. However, the construction strategy usually has limitations because it only apply to specific code. The optimization can also be considered on decoding schemes like the backtracking scheme and the two-stage algorithm, both of them need to get a priori information of the harmful trapping sets, which is extremely difficult to get. Concatenated strategy has also be implemented according the cooperation of component decoders, but the weakness is the increased redundancy may lead to rate loss. In general, it is necessary to obtain a efficient strategy with low complexity and widely available.

In this paper, we proposed a bit-rectification strategy over Monte Carlo simulation and optimized parallel concatenated Gallager codes (PCGC) decoder [10]. The reliable level and selected number of the most erroneous bits are determined by Monte Carlo simulation. And then the erroneous bits will be re-encoded to obtain a targeted parallel concatenated Gallager codes (PCGC), so it can obtain more reliable channel information from another component decoder under the optimized PCGC decoder. Finally, the negative effects of these erroneous bits can be exhausted with high possibility. Simulation results show that it can effectively reduce the error floor while keeping the coding rate basically unchanged.

2 PRELIMINARIES

1.1 Monte Carlo simulation

In our research, the bit-searching method is based on Monte Carlo simulation. For the LDPC codes, this simulation can locate the most erroneous bits rather than locate the trapping sets directly. The Monte Carlo simulation formula can be mathematically expressed as:

$$\hat{P}_{MC} = \frac{1}{N}\sum_{i=1}^{N} E_j(i) \tag{1}$$

Where the $\hat{P}_{MC}$ denoted the Monte Carlo simulation value of bit j, $E_j(i)=1$ if the bit j do not correctly converge during the i th iteration. $E_j(i)=0$ if the bit j correctly converge. N is the number of Monte Carlo simulation runs.

1.2 Decoding scheme of PCGC

The code construction after bit-searching is a special kind of parallel concatenation structure, so some improvements can be considered on the conventional decoder.

The conventional decoder has already dedicated in [10]. As shown in Fig.1, y_c^0 represents the received bits which corresponding to the information bits x, y_c^1 and y_c^2 indicate the bits corresponding to the parity check bits of two component codes. A whole iterate decoding process is named a super iteration. At the first super iteration, the decoder 1 computes the extrinsic information $p_{1e}(\hat{c})$ by utilizing the received sequence y_c^0 and y_c^1 . The decoder 2 will calculate the extrinsic information $p_{2e}(\hat{c})$ by $p_{1e}(\hat{c})$ obtaining from decoder1. In the next iterations, the decoder 1 will calculate the extrinsic information $p_{1e}(\hat{c})$ by $p_{2e}(\hat{c})$, which is product from the decoder 2. This process of super iteration will continue until all the component decoders get a valid word or reach the maximum number of super iterations. If all decoders do not successfully converge until the max number of iterations is reach, the final output is the decoding result of the decoder 2.

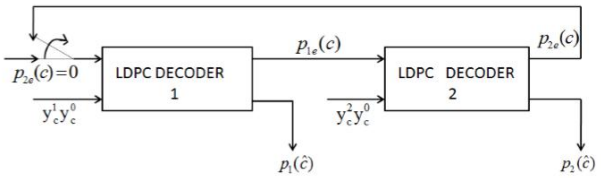

Fig.1 Conventional decoder of PCGC

3 Bit-rectification strategy

For the error floor phenomenon of LDPC codes, we proposed an effective Bit-rectification strategy based on bit-searching method and modified PCGC decoder. Firstly, we obtain the reliable level of bits by bit-searching method, which is based on the Monte Carlo simulation. The detail is shown in algorithm 1.

Algorithm 1 The Monte Carlo simulation algorithms

Intialization: Give the codeword Y ,code length n and run number N.
$E=(E_1,E_2,...,E_n)$ *and* $E_1=E_2=...=E_n=0$
for i=1 to N **do**
  Decode the codeword Y under AWGN Channel
  for j=1 to n **do**
    if bit j is wrongly decoded
      $E_j=E_j+1$
    end if
  end for
end for
Obtain the reliable level vector $E=(E_1,E_2,...,E_n)$

Now consider the reliable level vector $\boldsymbol{E}$. as the value of E_j get bigger, the reliable level will become smaller. So the erroneous bits can be selected from the top of the vector E.

As shown in Fig.2 (a), the variable nodes (*VN*s) with different gray-scales represent different reliable levels, the darker the bits, the less reliable the bits. We assume that the number of erroneous bits we selected by Monte Carlo simulation is a. As shown in Fig.2 (b), the a bits will be consider as the message sequence to be re-encoded by the (a+b, a) LDPC code. In addition, for the different original code length and rate, the quantity of the erroneous bits should be carefully considered. The purpose of this idea is to avoid serious rate loss while ensuring better performance. If we choose the (a+b, a) code for (n, m) original code in the process of re-encoding, and then the modified coding rate $r_m=m/(n+b)$. Since only the a little percentage of bits are involved in trapping sets, so $b \ll n$ and it will not lead much rate loss.

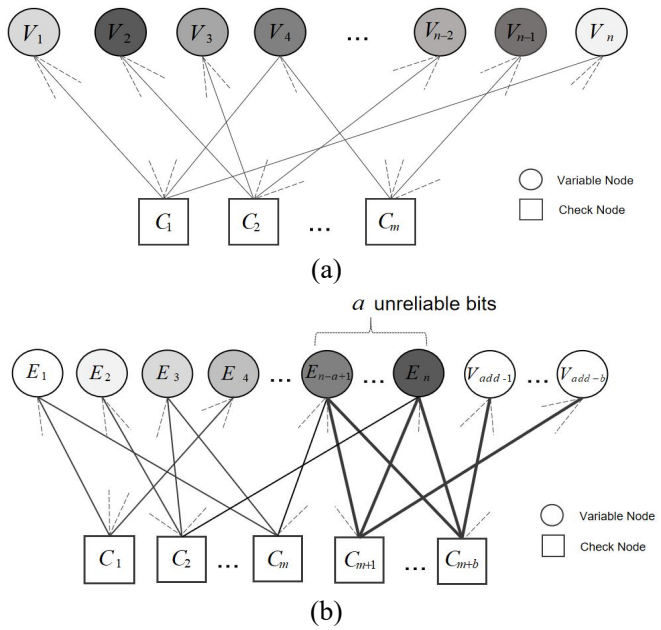

Fig.2 (a) Original Tanner graph (b) Modified Tanner graph before bit-searching

As we know, the trapping sets usually causes the bit cannot correctly converge all the time. Therefore, the bits we select through Monte Carlo simulation will have a high probability of being in the trapping sets. After re-encoding, these bits will exist in two different Tanner graph, the probability that the different structures have a trapping set on the same bit is very low. For the erroneous bits, more reliable information can be obtained from another structure, so the erroneous information of the original code can be rectified. The final result is that the original code may jump out of the trapping set with a high probability. For the complexity and latency, the bit-searching algorithm choose to rectify the bits involving in the trapping sets with high probability rather than determine the exact location of the harmful trapping sets by complex analysis of Tanner graph, so that it can eliminate the negative effects of trapping sets while keeping the complexity and latency basically unchanged.

In order to make better use of this structural feature and ensure the full interaction of extrinsic information, we do an improvement on the conventional PCGC decoder. Two independent maximum number k of iteration is set for the component decoders respectively to ensure sufficient local iterations. The global iteration number is set as W.

As shown in Fig.3, a whole iterative process is called a super iteration. In the first super iteration, decoder 1 computes the $p_1(\hat{c})$ with the received sequence under SPA, this process continue until correctly converging or the maximum number of iterations is reached. The decoder 2 is activated in the case where the decoder 1 fail to converge, its only output is the extrinsic information $p_{1e}(\hat{c})$ transmitting to decoder 1 as a priori information for next super iteration. This process of decoding will continue until correctly converge or the maximum number of the maximum number of super iteration is reach.

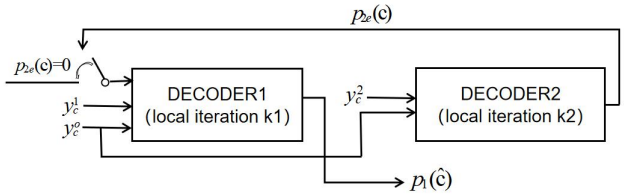

Fig.3 Modified PCGC decoder

It is noted that the (Log-likelihood Ratio) LLR held by the erroneous bits is wrong with high probability, so we set the decoder 1 do not transmit the extrinsic information to the decoder 2. Since these erroneous bits are in different Tanner graph structures denoted by different parity check matrix, the decoder 2 is set for the purpose of ensuring the erroneous bits can get correct LLR information from another structure. In the case where the decoder 1 cannot converge correctly, the correct LLR information can be obtained from the decoder 2 with high probability. These erroneous bits transmit the correct LLR information to the decoder 1, effectively avoiding the trapping sets. Finally, the output of the decoder 1 is taken as the final output.

We find that in the improved decoder, the transmission of extrinsic information is equivalent to the LLR information rectification or flipping of the bits in the trapping sets, and then effectively reducing the error floor.

4 Simulation result

4.1 Simulation of (1057, 813) MacKay code

In this section, we will give the example to show the effectiveness of the bit-rectification strategy. We assume that all the simulation experiments are under BPSK modulation and AWGN channel.

The LDPC codes we use for the simulation is the (1057, 813) MacKay codes. Because the bit-searching algorithm based on Monte Carlo simulation is performed at a certain SNR, we choose a medium SNR=4.15dB to obtain a reliable level sequence. ***E***. Set the running number of Monte Carlo as 5000 and 10000 respectively. Considering the rate loss, we need to carefully select the number of erroneous bits. We sort the sequences ***E*** to observe its characteristic. As shown in Figure 4, the two curves represent the sorted ***E*** with running number of 5000 and 10000, we found that there is always a waterfall area in ***E***. This waterfall area divides all the bits into two levels of reliability.

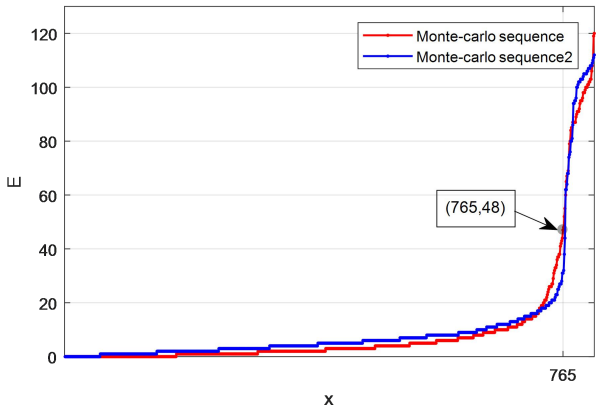

Fig.4 The curve of sorted ***E***

We randomly select the critical point (765, 48) at the waterfall area, the number of bits with lower reliable level is 48. So we selected the number of erroneous bits 36, 48, and 60 respectively for the test of performance. The code words used for re-encoding all have a coding rate of 1/2. The effect caused by the number of selected-bits can be observed from the Fig.5.

Taking the 48-selected as an example, after re-encoding, the coding rate is reduced from 0.769 to 0.735. The number of iterations is large enough to ensure reliability. It can be found from Fig. 5 that the bit-rectification strategy based on the bit-searching and re-encoding scheme can effectively improve the error floor phenomenon. Without considering the adding redundancy, the more erroneous bits are selected, the better the performance will be. The random-selected scheme do not have efficient improvement of error floor.

Consider the trade-off may be made between the BER performance and rate loss to get the optimal solution. We found that when the selected number of erroneous bits is 60, there is just fewer improvement of performance compared to that of the 48-selected, the advantage of more redundancy is almost negligible. Therefore, to obtain the number of the erroneous bits from the waterfall area can be the optimal solution.

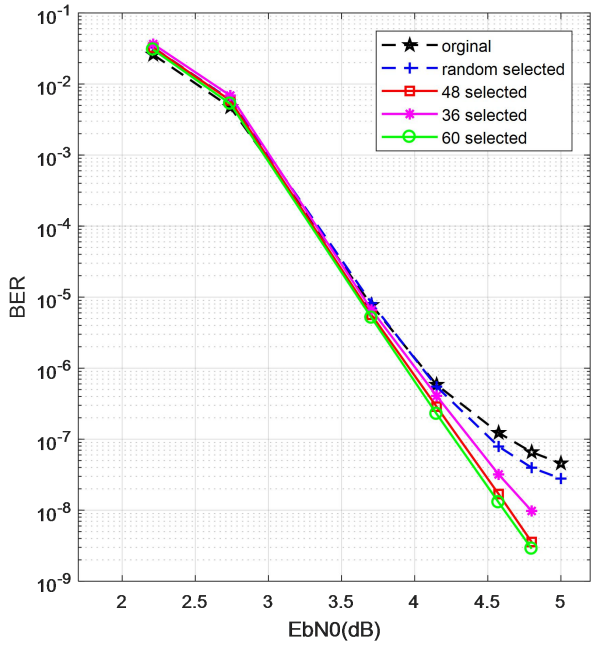

Fig.5 The performance of the bit-rectification strategy with different bit-selected

It can be seen from Fig. 5 that near the error floor area of the original (1057, 813) MacKay code, the bit-rectification strategy with 48-selected bits reduce the error floor from 10^{-7} to 10^{-9}. There is no error floor phenomenon appear until the BER of 10^{-9}. It strongly confirms that the BER performance is improved better in the high E_b / N_0 area with the bit-searching algorithm and modified PCGC decoder, despite a negligible loss of coding rate.

In order to further optimize the performance of the bit-rectification strategy, we replaced the (96, 48) MacKay code with the (96, 48) PEG code, which has a better structure. Through many experiments, the performance has been further improved. As shown in Figure 6, the error floor is also further reduced.

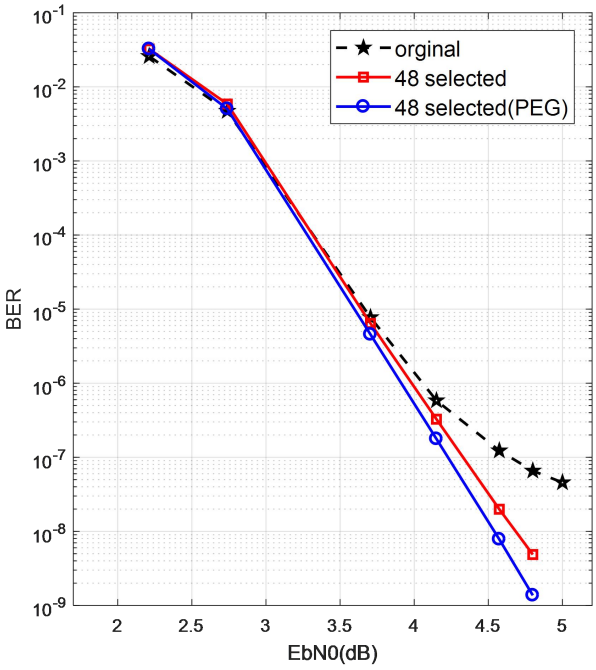

Fig.6 The performance of bit-rectification strategy with (96, 48) PEG code

4.2 Complexity Analysis

As we know, Conventional SPA decoder has about $6n_v d_{ave}$ floating-point multiplications per iteration, where d_{ave} is the average column-weight of the parity-check matrix and n_v is the number of variable nodes. Hence, the total multiplication operations $c = 6n_v d_{ave} N_s$, where N_s is the average number of iteration. The complexity comparison between original code and the code under bit-rectification strategy can be seen from this aspect. For the original (1057,

813) MacKay code, $n_v = 1057$ and $d_{ave} = 3$, and for the modified code with 48-selected erroneous bits, $n_v = 1105$ and $d_{ave} = d_v + (d_v * n_{add}) / n_v = 3.13$, where the is the number of unstable bits being selected. In the high SNR=4.15 dB, the average numbers of original code is N_s =2.869 and the average numbers of modified code is N_s =2.876. Hence, the percentage of multiplication operations is approximate $1 - c_{original} / c_{\mathrm{mod}ified} = 8.54\%$ reduction. Therefore, with the significantly reduction of error floor, the bit-rectification strategy only has a increased complexity of 8.54%.

5 Conclusion

In this paper, we proposed a bit-rectification strategy based on Monte Carlo simulation and modified PCGC decoder. The bit-searching simulation can locate the erroneous bits without analyzing specific structure of the trapping sets. Furthermore, the simulation result shows that it can efficiently lower the error floor. In the further research, we are committed to find more precise bit-selection criteria and achieve better performance.

6 Acknowledgments

This work is supported by NSFC under grant number U1534201, 61872031.

7 Reference

[1] R. G. Gallager, “Low-density parity-check codes,” IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, 1962.

[2] T. Richardson, “Error floors of LDPC codes,” in Proceedings of the annual Allerton conference on communication control and computing, vol. 41. The University; 1998, 2003, pp. 1426–1435.

[3] S. Kang, J. Moon, J. Ha, and J. Shin, “Breaking the trapping sets in LDPC codes: check node removal and collaborative decoding,” IEEE Trans. Commun., vol. 64, no. 1, pp. 15–26, Jan. 2016

[4] Le Dong, Ziyu Zhao, Jing Lei, Er-bao Li, " Lower error floor of LDPC codes based on trapping sets elimination", in Proc. International Conference on Wireless Communications and Signal Processing (WCSP) 2014

[5] J. Li, S. Lin, K. Abdel-Ghaffar, W. E. Ryan, and D. J. Costello, Jr, "Globally Coupled LDPC Codes." in Proc. Inf. Theory Applic. Workshop, San Diego, CA., Jan. 31 - Feb. 5, 2016.

[6] J. Kang, L. Zhang, Z. Ding, and S. Lin, “An Iterative Decoding Algorithm with Backtracking to Lower the Error-Floors of LDPC Codes," in Proc. IEEE Globecom. Conf., New Orleans, LA, Nov. 2008

[7] J. Kang, L. Zhang, Z. Ding, and S. Lin, “A two-stage iterative decoding of LDPC codes for lowering error floors," in Proc.IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 59, NO. 1, JANUARY 2011

[8] X. Xiao, M. Nasseri, B. Vasić, and S. Lin, “Serial concatenation of ReedMüller and LDPC codes with low error floor,” in Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton), Oct. 2017, pp. 688–693

[9] M. Nasseri, T. Wang, S. Lin, "Concatenated finite geometry and finite field ldpc codes", International Conference on Signal Processing and Communication Systems (ICSPCS), 2017

[10] H. Behairy and S. Chang, "Parallel concatenated Gallager codes," Electronics Letters, vol. 36, no. 24, pp. 2025-2026, 2000