

An Iterative Scheme for Lowering the Error-Floor of Low-Density Parity-Check Codes

Zuohong Xu ^a, Yanmin Ma ^b, Qian Cheng ^c, Jiang Zhu ^d and Kai Gao ^e

University of Defense Technology, Changsha 410000, China.

^azuohong.xu@outlook.com, ^bmayanmin16@nudt.edu.cn, ^cchengqian14a@nudt.edu.cn,
^djiangzhu@nudt.edu.cn, ^egaokai000@nudt.edu.cn

Abstract. This paper proposes a two-stage iterative decoding scheme named matrix transform belief propagation scheme (MT-BP) based on conventional belief propagation (BP) algorithm for low-density parity-check (LDPC) codes. By introducing information measurement, location vectors and parity check matrix transform, the proposed method can establish additional parity check equations and take more information bits of variable nodes into consideration compared with conventional BP algorithm. The simulations are performed on random regular LDPC codes and algebraic quasi-cyclic codes (QC-LDPC), the experimental results show MT-BP method can efficiently reduce the bit error rate (BER).

Keywords: Error floor, two-stage iterative decoding scheme, belief propagation, LDPC, parity check equations.

1. Introduction

It is well-known that low-density parity-check (LDPC) codes are currently the most promising and widely-used channel codes [1-4]. They can be represented by a bipartite graph named Tanner graph, where the messages pass along the edges connecting variable nodes (VNs, denoted by $c_{t,t \in \{1, \dots, M\}}$) to check nodes (CNs, denoted by $v_{j,j \in \{1, \dots, N\}}$). The most classic iterative decoding method for LDPC codes is BP algorithm [5]. However, some factors are influential to the decoding performance including error floor, for example, trapping sets, girth and rate. A great deal of research is dedicated to improving the decoding performance and practical applications. For example, in [6], authors show that it is efficient to lower the error floor by adding redundant rows to parity check matrix, but these methods need to know the location of TS at first. [7] proposes a technique for breaking TS through removing certain check nodes. Concisely, the method first locates the bit involved in the TS, then removes the check nodes (CNs) connected with the chosen variable nodes (VNs). In [8], authors show that once TS can be identified, a sum-product algorithm (SPA) decoder can be custom-designed to improve the decoding performance. However, all above-mentioned methods must know the location of TS first, which is time-consuming to implement with long LDPC codes. Besides, [9] proposes a scheme for bit recovery of LDPC codes by adding new check nodes to connect the most erroneous and the most reliable bits. However, the error level of the codeword bit is estimated through Monte Carlos simulations, so it requires new simulations for a new code, which has a high complexity.

To balance the decoding complexity and decoding performance, a scheme named BP-MT is proposed in this paper. Firstly, we define IM of VNs and IM of CNs, which are used to simply represent the probability that a bit of VN is correct and a parity check equation holds, respectively. Then, based on IM, we construct location vectors and perform matrix transform on parity check matrix H to obtain a new parity check matrix H_{new} . The meaning of matrix transform is that based on existing parity check equations, we establish additional parity check equations. Therefore, more information of VNs can be taken into consideration to help decode.

2. Conventional BP Algorithm

BP algorithm is a soft-decision algorithm which takes the probability as input. Assume the signal transmitted on an AWGN channel with variance σ^2 and binary phase shift keying (BPSK) modulation, $y = (y_0, y_1, \dots, y_{N-1})$ is the soft-decision received vector in the output of the receiver

detector, $\mathbf{z} = (z_0, z_1, \dots, z_{N-1})$ is the hard-decision vector and $\mathbf{s} = (s_0, s_1, \dots, s_{M-1})$ is the syndrome. Assume \mathbf{H} is a $M \times N$ matrix and \mathbf{H}^T is the transpose. We use $r_{tj}^{(l)}$ to represent the probability information from c_t to v_j in the l -th iteration, and $q_{jt}^{(l)}$ to represent the information from v_j to c_t in the l -th iteration. Let $N(j) = \{t: 0 \leq t \leq M-1, h_{t,j} = 1\}$ be a set consisting of CNs which are connected to v_j , and $M(t) = \{j: 0 \leq j < N-1, h_{t,j} = 1\}$ be a set consisting of VNs which are connected to c_t . The conventional BP algorithm is expressed as follows.

Algorithm for Conventional BP Algorithm

Initialization. set $l = 1$ and preset the maximum iteration number, compute the log-likelihood ratio (LLR) probability information for VNs,

$$q_j^{(0)} = \log \frac{1 + \exp(\frac{2y_j}{\sigma^2})}{1 + \exp(-\frac{2y_j}{\sigma^2})} = 2 \frac{y_j}{\sigma^2}. \quad (1)$$

Step 1. CNs update. Calculate the information of CNs received from VNs:

$$r_{tj}^{(l)} = 2 \tanh^{-1} \left\{ \prod_{j' \in M(t) \setminus j} \left(\frac{1}{2} q_{j't}^{(l-1)} \right) \right\}. \quad (2)$$

Step 2. VNs update. Calculate the information of VNs received from CNs:

$$q_{jt}^{(l)} = r_t^{(l-1)} + \prod_{t' \in N(j) \setminus t} (r_{t'j}^{(l)}) \quad (3)$$

Step 3. Hard decision and compute the syndrome

$$z_j = \begin{cases} 0, & q_j > 0 \\ 1, & q_j < 0 \end{cases} \quad (4)$$

If $\mathbf{s} = \mathbf{z} \cdot \mathbf{H}^T = \mathbf{0}$, the decoding is successful, output the decoded codeword; if $\mathbf{s} = \mathbf{z} \cdot \mathbf{H}^T \neq \mathbf{0}$, $l = l + 1$, return to Step 1. If l exceeds the preset maximum iteration number, the decoding fails.

3. Our Proposed Scheme

3.1 Information Measurement.

In BP algorithm, we usually check whether the decoding is correct by judging whether the syndrome \mathbf{s} is 0. If a CN c_t satisfies the parity check equation, its corresponding element s_t in \mathbf{s} equals to zero, we believe it has a very high probability that the bits of its adjacent VNs are correct. Similarly, if s_t is not zero, there must be some wrong bits of its adjacent VNs.

Based on \mathbf{s} , we define the IM of VNs and IM of CNs. The steps for computing the IM is summarized as follows:

Algorithm for Computing the Information Measurement

Step 1. Perform the hard decision \mathbf{z} on \mathbf{y} .

Step 2. Compute the syndrome for every CN.

Step 3. Compute the IM for each VN by adding up values of syndrome of adjacent CNs,

$$\text{IM}_{VN_j} = \sum_{t \in N(j)} s_t, \quad (5)$$

Step 4. Compute the IM for every CN by adding up IM of adjacent VNs,

$$IM_{CN_t} = \sum_{j \in M(t)} IM_{VN_j}. \quad (6)$$

From the definition of IM, we can easily see the higher IM_{VN_j} that v_j has, the more incorrect parity check equations are related to v_j , the more likely that v_j is a wrong bit. Similarly, the higher IM_{CN_t} that c_t has, the more wrong VNs are connected to it, the higher probability that this parity check equation does not hold.

3.2 Location Vectors and Matrix Transform

Assume I is an identity matrix with size $M \times M$, then we have $H = I \cdot H$. For the CNs connected to v_j , assume c_t has the maximum IM and the c_k has the minimum IM. We can construct a vector $v_{\text{location}} = (0,0,0,1, \dots, 0,1,0,0)$ named location vector with length M , where both t -th and k -th elements in v_{location} equal to 1 while other elements are 0.

Then, we transform the identity matrix I as follows:

$$I_{\text{Transform}} = \begin{bmatrix} I \\ v_{\text{location}} \end{bmatrix} = \begin{bmatrix} I \\ 0,0,0,1, \dots, 0,0,1,0,0 \\ \uparrow \quad \uparrow \\ t\text{-th} \quad k\text{-th} \end{bmatrix}. \quad (7)$$

Thus, the size of matrix $I_{\text{Transform}}$ becomes $(M + 1) \times M$ and we can get the matrix H_{new} by multiplying the $I_{\text{Transform}}$ with H as

$$H_{\text{new}} = I_{\text{Transform}} \cdot H \quad (8)$$

If we want to construct c location vectors, we can first sort IM of CNs in descending order. Then we choose first c CNs (with maximum IM) and last c CNs (with minimum IM). Furthermore, the h -th location vector v_{location_h} can be constructed with the h -th maximum CN and h -th minimum CN. $I_{\text{Transform}}$ Can be represented as follows:

$$I_{\text{Transform}} = \begin{bmatrix} I \\ v_{\text{location}_1} \\ v_{\text{location}_2} \\ \vdots \\ v_{\text{location}_c} \end{bmatrix} = \begin{bmatrix} I \\ 0,0,0,1, \dots, 0,0,1,0,0 \\ 1,0,0,0, \dots, 0,0,0,0,1 \\ \vdots \\ 0,0,1,0, \dots, 0,1,0,0,0 \end{bmatrix}. \quad (9)$$

Theorem: The process of checking whether the decoding is correct based on H is equivalent to checking based on H_{new} .

Proof: From (8), we have $z \cdot H_{\text{new}}^T = z \cdot (I_{\text{Transform}} \cdot H)^T = (z \cdot H^T) \cdot I_{\text{Transform}}^T$, from (9) we can easily see Hamming distance between any two location vectors is 4 (there are no elements 1 in the same position of any two location vectors), so it is easy to see that $I_{\text{Transform}}$ is a row full rank matrix ($I_{\text{Transform}}^T$ is column full rank). Therefore, checking whether the decoding is correct based on $z \cdot H_{\text{new}}^T$ is equivalent to computing $z \cdot H^T$.

3.3 The Whole Decoding Algorithm

The decoding scheme includes two stages. In the first stage, we adapt the conventional BP algorithm, if the decoding fails, we perform the MT-BP decoding scheme. The whole algorithm is as follows.

The Whole Decoding Algorithm

Stage 1. The conventional BP algorithm

Step 1. Carry out the BP algorithm on received information y .

Step 2. Check whether the decoding is correct by computing the syndrome $s = z \cdot H^T$. If $s = 0$, output the decoded codeword; if $s \neq 0$, go to Stage 2.

Stage 2. Carry out MT-BP scheme

Step 1. Set c , carry out the algorithm to compute the information measurement for check node.

Step 2. Set $i = 1$.

(1) Construct location vectors v_{location_i} and perform matrix transform to get matrix H_{new} .

(2) Perform the BP algorithm based on H_{new} . Check whether the decoding is correct by computing $s = z \cdot H_{\text{new}}^T$. If $s = 0$, output the decoded codeword; if $s \neq 0$ and $i < c$, $i = i + 1$ and go to Step 2. (1); if $s \neq 0$ and $i = c$, the decoding fails.

Table 1. Information of Codes

C_i	Length	Rate	Construction method
C_1	(1024,512)	1/2	Mackay 1A
C_2	(1016,508)	1/2	CPM-QC-LDPC
C_3	(504,252)	1/2	CPM-QC-LDPC
C_4	(195,150)	0.77	CPM-QC-LDPC

4. Simulation Results

We compare the conventional BP algorithm [10] and MT-BP algorithm on random regular LDPC codes and algebraic QC-LDPC codes constructed based on circulant permutation matrices (CPM-QC-LDPC) [2], the codes are shown in Table 1. Simulations are performed over the additive white Gaussian noise (AWGN) channel with binary phase shift keying (BPSK) modulation. The maximum iteration number for BP algorithm is 35. E_b Represents the average energy per information bit and N_0 denotes the one-sided power spectral density. The simulations are shown in Fig. 1.

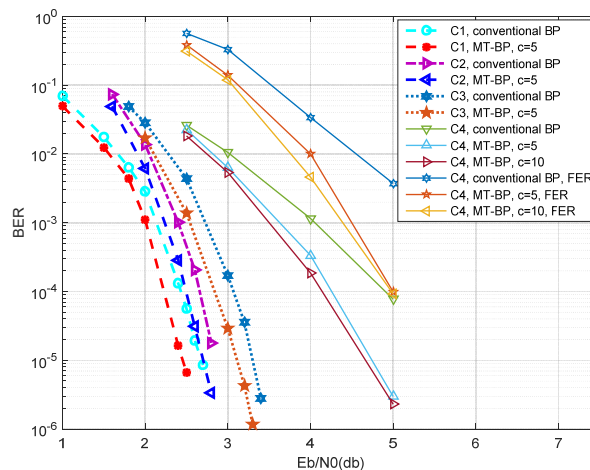


Fig. 1 BER Performances of Various LDPC Codes.

From Fig. 1, we can see our proposed MT-BP algorithm has a better decoding performance compared with conventional BP algorithm. For (1024, 512) regular LDPC code, when E_b/N_0 exceeds 2 dB, there is an around 0.2 dB coding gain. For (1016, 508) CPM-QC-LDPC code, when E_b/N_0 exceeds 2.2 dB, the coding gain is around 0.2 dB. For (504,252) CPM-QC-LDPC code, the coding gain is more than 0.2 dB when E_b/N_0 is more than 2.3 dB. For (195,150) CPM-QC-LDPC code, it is clear to see the coding gain for BER and FER is over 0.5 dB when E_b/N_0 is over 3.5 dB. Moreover, the decoding performance gets better as c increases.

5. Conclusion

A two-stage iterative decoding scheme named MT-BP, which can make more use of the information of VNs by parity check matrix transform, is proposed in this paper. Experimental results illustrate that MT-BP algorithm can efficiently improve the BER performance.

References

- [1]. S. J. Johnson. Introducing low-density parity-check codes. University of Newcastle, Australia, 2010.
- [2]. J. Li, K. K. Liu, S. Lin, et al. Algebraic quasi-cyclic ldpc codes: construction, low error-floor, large girth and a reduced-complexity decoding scheme. *IEEE Trans. Commun.* vol. 62 (2013) No. 8, p. 2626-2637.
- [3]. J. Li, K.K. Liu, S. Lin. Quasi-cyclic ldpc codes on two arbitrary sets of a finite field. *IEEE International Symposium on Information Theory*, Honolulu, HI, USA, 29 June 2014, p. 2454-2458.
- [4]. J. Li, K.K. Liu, S. Lin. Decoding of quasi-cyclic ldpc codes with section-wise cyclic structure. *Information Theory and Applications Workshop*, San Diego, CA, USA February 2014, p. 1-10.
- [5]. J. Barbier, F. Krzakala. Approximate message-passing decoder and capacity-achieving sparse superposition codes. *IEEE Trans. Information Theory*, vol. 63 (2017), p. 4894-4927.
- [6]. S. Laendner, T. Hehn, O. Milenkovic, et al. When does one redundant parity-check equation matter. *IEEE Global Telecommunications Conference*. San Francisco, CA, USA. 27 Nov.- 1 Dec. 2006, p.1-6.
- [7]. S. Kang, J. Moon, J. Ha, et al. breaking the trapping sets in LDPC codes: check node removal and collaborative decoding. *IEEE Trans. Commun.*, Vol.64 (2016) No. 1, p.15-26.
- [8]. H. Yang, W. E. Ryan. Low-floor decoders for LDPC codes. *IEEE Trans. Commun.*, vol.57 (2009) No.6, p.1663-1673.
- [9]. D. Yuan, L. Li, Y. Liu. Lowering the Error Floors of Low-Density Parity-Check Codes with Additional Check Nodes. *Information Theory Workshop*. Kaohsiung, Taiwan, 6-10 Nov. 2017.
- [10]. D. J. C. Mackay. *Encyclopedia of Sparse Graph Codes*. Information on: <http://wol.ra.phy.ca>., 1999.