An Improved Normalized BP Based Decoding Algorithm for LDPC Codes

Haigang Zhang†, Dongfeng Yuan†, and Cheng-Xiang Wang††

†School of Information Science and Engineering, Shandong University, Jinan, Shandong, 250100, P. R. China.
E-mail: hgzhang@sdu.edu.cn dfyuan@sdu.edu.cn
†† Joint Research Institute for Signal and Image Processing, School of Engineering & Physical Sciences, Heriot-Watt University. Edinburgh EH14 4AS, UK. E-mail: cheng-xiang.wang@hw.ac.uk

Keywords: LDPC codes, BP algorithm, Normalized BP.

Abstract

A uniformly most powerful (UMP) belief propagation (BP) based algorithm is referred to as a simplified version of a BP algorithm with reduced complexity but performance loss for low density parity check (LDPC) codes. To compensate the performance loss, the normalized BP based algorithm was proposed, where the normalization factor was derived by mean ratio or by minimizing the mean square error. In this paper, an improved novel normalized BP based algorithm is proposed. The normalization uses multiplicative factor instead of divisional factor. The novel scheme shows better performance than the existing normalized BP based algorithms while keeping the same implementation complexity. The simulation is done for two kinds of LDPC codes: random constructed codes and finite geometry codes. At high signal-to-noise ratio (SNR) region, the proposed scheme can achieve even better performance than BP algorithm for short length random constructed LDPC codes.

1 Introduction

LDPC codes [1, 2] can achieve near Shannon limit performance and have attracted more and more attention in recent years. LDPC codes are defined by a very sparse parity check matrix which contains only a few ones while mostly zeros. We can represent the parity check matrix of LDPC codes by using a bipartite graph with one subset of check nodes and the other subset of bit nodes [3]. LDPC codes can be decoded based on either soft information or hard decision. Soft decoding can provide a much better performance than hard decoding and therefore is widely applied. Soft decoding can be implemented by iterative decoding based on the belief propagation (BP) or log-likelihood ratio BP (LLR-BP) algorithm [4], which can achieve the maximum a posterior (MAP) decoding if there are no cycles in the parity check matrix. Although the BP or LLR-BP algorithm is a powerful tool for iterative decoding, its hardware implementation is restricted due to the high complexity, which mainly lies in the processing in check nodes and bit nodes of the bipartite graph.

To reduce the implementation complexity, different simplified versions of the BP algorithm were proposed. In [5], the iterative a posteriori probability (APP) based algorithm was used to simplify the processing in bit nodes. In [6], the uniformly most powerful (UMP) BP based algorithm was proposed to simplify the processing in check nodes using the symbol with the smallest absolute value instead of multiplication. The complexity reduction in [5] and [6] was induced by approximations, which in turn cause performance degradation, especially for LDPC codes with check sums of large weights. In [7], a normalization scheme was proposed to improve the performance of the UMP BP based algorithm, resulting in the so-called normalized BP based algorithm. The normalization was done by adding a normalization factor to adjust the soft values obtained from the first iteration [7]. The normalization factor in [7] was calculated through the mean ratio and is not necessarily the optimum. In [8], a modified version of the normalized BP based algorithm was proposed by using a new normalization factor, which yields the minimum of mean square error in the first iteration. It was shown that the modified normalized BP based algorithm [8] is better than the original normalized BP based algorithm [7] in terms of the error performance.

There are two statements in [7]: the value calculated by the horizontal step of UMP BP based algorithm has the same sign as and greater magnitude than the value in BP algorithm. So the normalization factor in [7] and [8] use divisional factor greater than 1 to adjust the value in UMP BP based algorithm to approach the value in BP algorithm. In this paper, multiplicative factor smaller than 1 is used for normalization. The new scheme can achieve better performance than the scheme in [8] and keeps the same complexity. At high SNR area, the proposed method can get even a little better performance than BP algorithm for short random constructed code since the short cycle in code matrix.

The paper is organized as follow. In Section 2, the LLR-BP, UMP BP based and normalized BP based algorithms are reviewed. The novel universal normalized scheme is introduced in detail in Section 3. Simulation results and discussions are given in Section 4. Finally, Section 5 concludes the letter.

2 Existing soft decoding algorithms

The parity check matrix of an LDPC code is denoted by H=$[H_{mn}]$, which is an $M \times N$ matrix indicating that the LDPC code has the transmitted block length N and information block length N-M if the matrix has full rank. The matrix H has ρ 1s in each row and γ 1s in each column. We denote the set of bits participating in check m by $N(m) = \{n : H_{mn} = 1\}$. Similarly, the set of checks that bit n participates in is denoted by $M(n) = \{m : H_{mn} = 1\}$. Also, we denote M(n)\m and N(m)\n as the set M(n) with check m excluded and the set N(m) with bit n excluded, respectively.

We assume BPSK modulation, which maps a codeword c= $(c_1, c_2, \cdots, c_N)$ to a transmitted sequence x= $(x_1, x_2, \cdots, x_N)$ according to $x_n = 2c_n - 1$ for n=1, 2, …, N. Then the modulated sequence x is transmitted over an additive white Gaussian noise (AWGN) channel. The received sequence is denoted by y=$(y_1, y_2, \cdots, y_N)$ with $y_n = x_n + w_n$, where w_n represents a Gaussian random variable with zero-mean and variance N0/2. For reasons of completeness, let us briefly define the following notations F_n, L_{mn}, z_{mn} and z_n related to a given iteration. The details can also be found in [7]. F_n is the LLR of bit n derived from the channel output yn and is initialized to $(4/N_0)y_n$. L_{mn} denotes the LLR of bit n sending from check m to bit n. z_{mn} represents the LLR of bit n sending from bit n to check m. It initially equals F_n in the first iteration. z_n is a posteriori LLR of bit n for hard decision to test whether the decoding iteration should stop.

For each iteration, the horizontal step and vertical step of the LLR-BP algorithm [7] are given as follows.

1) Horizontal step (processing in check nodes):

For each m and n,

$$T_{mn} = \prod_{n' \in N(m)\backslash n} \frac{1 - \exp(z_{mn'})}{1 + \exp(z_{mn'})} \tag{1}$$

$$L_{mn} = \ln \frac{1 - T_{mn}}{1 + T_{mn}}. \tag{2}$$

2) Vertical step (processing in bit nodes):

For each m and n, update z_{mn} and z_n by

$$z_{mn} = F_n + \sum_{m' \in M(n)\backslash m} L_{m'n} \tag{3}$$

$$z_n = F_n + \sum_{m \in M(n)} L_{mn} \tag{4}$$

respectively. Clearly, the complexity of the LLR-BP algorithm mainly locates in the horizontal step.

The UMP BP based algorithm [6] simplifies the horizontal step as follows:

$$L_{mn} = (-1)^{\overline{\sigma_m + \sigma_{mn}}} \min_{n \in N(m)\backslash n} |z_{mn'}| \tag{5}$$

$$\sigma_{mn} = \begin{cases} 1, & if\ z_{mn} > 0 \\ 0, & if\ z_{mn} \le 0 \end{cases} \tag{6}$$

$$\sigma_m = \sum_{n \in N(m)} \sigma_{mn} \bmod 2. \tag{7}$$

Due to the introduced approximation in (5), decision errors of the second decoding iteration usually exceed those of the first iteration in the UMP BP based algorithm. So it is necessary to improve the information accuracy obtained in the first iteration.

Let L_1 and L_2 denote the value L_{mn} computed by the LLR-BP and UMP-BP-based algorithms, respectively. From [7], we know that L_1 and L_2 have the same sign and $|L_2| > |L_1|$. A factor α is added naturally in order to adjust L_2 to approach L_1. The algorithm is known as normalized BP based algorithm. In [7], the normalization factor α was calculated by the ratio of the mean values of L_2 and L_1, i.e.,

$$\alpha = \frac{E(|L_2|)}{E(|L_1|)}. \tag{8}$$

Then, in a modified normalized BP based decoding algorithm [8], a new normalization factor was calculated through minimizing the mean square error E[$(\alpha |L_1| - |L_2|)^2$] and is given by

$$\alpha_{new} = \frac{E\left[\left|L_1 \cdot L_2\right|\right]}{E\left[\left|L_1\right|^2\right]}. \qquad (9)$$

3 Improved normalized BP based algorithm

The novel algorithm also begins from (5). A multiplicative factor is used on L_2 to approach L_1 instead of divisional factor in [7] and [8]. The idea is similar with the method in [10] which is adapted to weighted bit-flipping decoding. Obviously, The novel factor is smaller than 1. It is denoted as α_p. The resulted value is denoted as L_3. So

$$L_3 = \alpha_p L_2. \qquad (10)$$

The coefficient α_p is also calculated by minimizing the mean square error $E\left[\left(\left|L_3\right| - \left|L_1\right|\right)^2\right]$.

$$E\left[\left(\left|L_3\right| - \left|L_1\right|\right)^2\right] = E\left[\left|L_2\right|^2\right]\alpha_p^2 - 2E\left[\left|L_1 L_2\right|\right]\alpha_p + E\left[\left|L_1\right|^2\right]$$

Then

$$\frac{dE\left[\left(\left|L_3\right| - \left|L_1\right|\right)^2\right]}{d\alpha} = 2E\left[\left|L_2\right|^2\right]\alpha_p - 2E\left[\left|L_1 L_2\right|\right]. \qquad (11)$$

So

$$\alpha_p = \frac{E\left[\left|L_1 \cdot L_2\right|\right]}{E\left[\left|L_2\right|^2\right]} \qquad (12)$$

The calculation process is identical to [8] but with different result. We can compare (9) and (12). Since L_1 and L_2 is not statistical independent, so α_p is not equal to the reciprocal of α_{new}. Therefore, the proposed algorithm is different from the modified normalized BP based algorithm in [8]. In the next section, we will give the different coefficients under different SNRs and show that the proposed scheme has better performance than the algorithms in [6]—[8].

As in [7] and [8], the coefficients are dependent on the SNR. We can use different coefficients at different SNRs. In this paper, for simplicity reasons, we only adopt the coefficients obtained at 3 dB SNR and apply them to all SNR values in the simulation.

4 Numerical Results

The parameters of the 1/2 rate random constructed LDPC code used in the simulations are N=816, M=408, ρ=5, and γ=10. The finite geometry code is (273,191) code with ρ=17 and γ=17 [9]. The maximum decoding iteration number in simulation is fixed at 50. In Table 1 we give α_p and α_{new} on different SNR for (816,408) code. The reciprocal of α_{new} is also given. The coefficients for (273,191) finite geometry code are given in Table 2. We can see that the inverse of coefficients in [8] seems to converge toward the proposed coefficients with the increasing of SNR.

SNR(dB)	α_{new}	$1/\alpha_{new}$	α_p
0	3.307	0.3024	0.2081
1	2.4466	0.4087	0.3193
2	1.9104	0.5235	0.4507
3	1.571	0.6365	0.5866
4	1.3554	0.7378	0.7084
5	1.2198	0.8198	0.8046

Table 1: Coefficients on different SNR for (816,408) LDPC code

SNR(dB)	α_{new}	$1/\alpha_{new}$	α_p
0	4.9147	0.2035	0.1107
1	3.0977	0.3228	0.2236
2	2.1531	0.4645	0.3816
3	1.6425	0.6088	0.5556
4	1.3592	0.7357	0.7088

Table 2: Coefficients on different SNR for (273,191) LDPC code

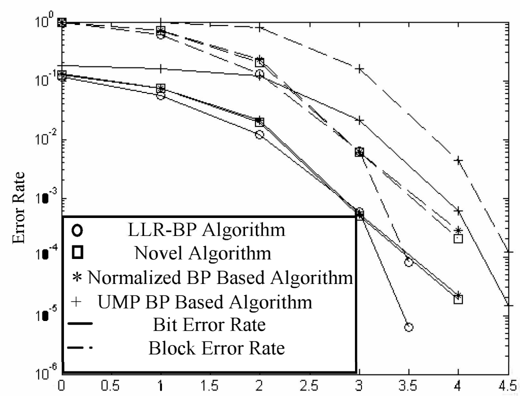

Figure 1: Bit error rate (solid line) and block error rate (dotted line) performance of the proposed decoding algorithm for different decoding algorithms (N=273, M=191, ρ=17, γ=17).

In Figure 1, we demonstrate the error performance simulation results for finite geometry code. The curves of normalized UMP BP based algorithm and proposed scheme are very close to the curve of LLR-BP algorithm at low SNR area. But the gap enlarges after 3dB. The performance curves of the bit error rate and block error rate have the same trend. The performance improvement of the novel scheme is very limited compared with normalized UMP BP based algorithm.

In Figure 2, we compare the bit error rate and block error rate performance of using different decoding algorithms for (816,408) code. The proposed scheme is better than the modified normalized BP based scheme in [8] and get even a little better performance than LLR-BP algorithm at the SNR of 3.5 dB. The curves of the bit error rate and block error rate have the same trend.

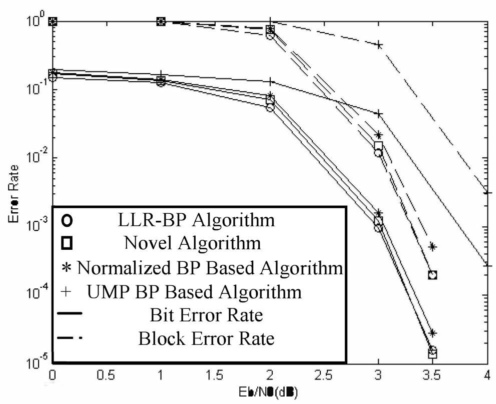

Figure 2: Bit error rate (solid line) and block error rate (dashed line) performance for different decoding algorithms (N=816, M=408, ρ=5, γ=10).

5 Conclusions

From the simulation results we can see that the proposed scheme can achieve near LLR-BP performance for random constructed LDPC codes. At a high SNR, it performs even a little better than LLR-BP algorithm. At the same time it remains the same implementation complexity as the normalized BP based algorithms in [7] and [8]. Therefore, the novel scheme provides a better tradeoff between the implementation complexity and error rate performance. Normalized UMP BP based algorithm performs comparatively worse for finite geometry LDPC codes. But the proposed scheme also performs best in this kind of codes.

Acknowledgments

The authors give thanks to David Mackay for sharing the LDPC matrix on the website. Thanks are also expressed to Dr. Ming Jiang in Southeast University, Nanjing, China.

This work is partly supported by: National Scientific Foundation of China (No. 60672036), Key Project of Provincial Scientific Foundation of Shandong (No. Z2006G04).

References

[1] R. G. Gallager. "Low-Density Parity-Check Codes", Cambridge, MA: MIT Press, 1963.

[2] D. J. C. MackKay, and R. M. Neal. "Near Shannon limit performance of low density parity check codes", Electronic Letters, vol. 32, no. 18, pp. 1645—1646, Aug. 1996.

[3] R. M. Tanner. "A recursive approach to low complexity codes", IEEE Transactions on Information Theory, vol. 27, pp. 533—547, Sept. 1981.

[4] J. Pearl. "Probabilistic Reasoning in Integlligent Systems: Networks of Plausible Inference", San Mateo, CA: Morgan Kaufmann, 1988.

[5] J. L. Massey. "Threshold Decoding", Cambridge, MA: MIT Press, 1963.

[6] M. Fossorier, M. Mihaljevic, and H. Imai. "Reduced complexity iterative decoding of low density parity check codes based on belief propagation", IEEE Transactions on Communications, vol. 47, pp. 673—680, May 1999.

[7] J. Chen, and M. P. C. Fossorier. "Near optimum universal belief propagation based decoding of low-density parity check codes", IEEE Transactions on Communications, vol. 50, no. 3, pp. 406—414, Mar. 2002.

[8] N. Kim, and H. Park. "Modified UMP-BP decoding algorithm based on mean square error", Electronic Letters, vol. 40, pp. 816—817, Jun. 2004.

[9] Y Kou, S Lin, and M P C Fossorier. "Low-density Parity-check Codes Based on Finite geometries: A rediscovery and New Results", IEEE Transactions on Information Theory, vol. IT-47, pp. 2711 - 2736, November 2001.

[10] Haigang Zhang, Dongfeng Yuan, Cheng-Xiang Wang, "A Modified Weighted Bit-Flipping Algorithm for LDPC Codes", to be published in Proceedings of the Second IET International Conference on Wireless, Mobile and Multimedia Networks-2008 (ICWMMN 2008), 12-15 October, 2008, Beijing China.