AE-BP: Adaptive Erasure Belief Propagation Decoding Algorithm of LDPC Codes

Chaonian Guo*, Xiangxue Li†‡, Dong Zheng†, Shengli Liu*, and Jianhua Li†
*Department of Computer Science and Engineering, Shanghai Jiao Tong University
†School of Information Security Engineering, Shanghai Jiao Tong University
Shanghai Key Laboratory of Information Security Management and Technology Research
‡ Corresponding author : xxli@sjtu.edu.cn

***Abstract*—In the decoding process of LDPC codes, variable node messages often fluctuate continuously and thereby become the hindrance to the successful decoding. This paper proposes an Adaptive Erasure Belief Propagation (AE-BP) decoding algorithm to reduce the influence of these unreliable messages. The idea behind AE-BP is recording the continuous fluctuant times of variable nodes by introducing a sequence of counters, and then adaptively erasing the messages according to their fluctuant times so that other messages would recover to more precise states. Semi-Gaussian approximation demonstrates adaptive erasure is a good candidate towards offsetting the defect of unreliable messages. Experimental simulations show that, for LDPC codes AE-BP outperforms most decoding algorithms in the literature.**

***Keywords*-LDPC codes; belief propagation; iterative decoding;**

I. Motivation

Low-Density Parity-Check (LDPC) codes, first proposed by Gallager [1] and brought into prominence by MacKay and Neal [2], [3], have been attracting a great deal of academic interests. LDPC codes achieve near Shannon limit performance [4], and have the advantages of simple descriptions, low complexity and easy implementations.

The LLR-BP or belief propagation decoding algorithm (also known as sum-product algorithm [5]) is one of the iterative decoding algorithms that approach maximum a posterior probability decoding for LDPC codes. The UMP-BP algorithm [6] was proposed to reduce the decoding complexity of LLR-BP. However, UMP-BP amplifies the check node messages, which leads to performance loss with respect to the LLR-BP decoding. Several correction methods were proposed in the literature in order to recover the performance loss of the UMP-BP decoding. The most popular correction method is Norm-BP algorithm [7]. This algorithm shares the idea that the performance loss can be recovered by reducing the amplitudes of check node messages, which is compelling as these messages are overestimated within the UMP-BP algorithm. The resulting performance of Norm-BP is better than that of UMP-BP. Another modified LLR-BP decoding algorithm was proposed in [8] (Osci-BP), which focuses on the log-likelihood ratios (LLRs) of the variable nodes and considers the LLRs in two consecutive iterations.

We observe that in the decoding process of LDPC codes, the behaviors of variable nodes are distinct, especially some of them may fluctuate continuously and frequently. These nodes are believed to be unreliable. To distinguish these variable nodes in decoding process, we introduce a sequence of counters CFT, which is defined as the continuous fluctuant time of variable node according to the sign of its Log-likelihood Ratio (LLR).

Table I
The Average Percentages of the Fluctuant LLRs.

counter CFT	1 DB	1.5 DB	2 DB	2.5 DB	3 DB
0	99.22%	94.91%	97.97%	98.18%	98.41%
1	.56%	2.73%	1.59%	1.44%	1.23%
≥ 2	.22%	2.36%	.44%	.38%	.36%

More precisely, we use LLR-BP algorithm to test the (504, 252) regular LDPC code, and Table I illustrates the statistics of continuous fluctuant times for variable nodes w.r.t. E_b/N_0. As showed in Table I, all variable nodes are classified into three types: (1) $CFT = 0$, (2) $CFT = 1$, and (3) $CFT \geq 2$. Since the variable nodes in type (1) do not make any sense to the reliability of the node, and the nodes in type (2) may reveal some unreliable information, we focus on those nodes in type (3). To damp the influence of these nodes, we adopt an adaptive erasure rule: if a node is recognized to fall into type (3), we set its LLR to zero so that other messages would recover to more precise states in the coming iteration. Semi-Gaussian approximation demonstrates adaptive erasure is a good candidate towards offsetting the influence of unreliable messages. Experimental simulations show that, for LDPC codes AE-BP outperforms most decoding algorithms in the literature, including the LLR-BP algorithm.

The rest of this paper is organized as follows. In Section II, we describe the notations and present the AE-BP decoding algorithm. Section III analyzes the average error probability by using semi-Gaussian approximation, and gives the results of Monte-Carlo simulations. Finally, a conclusion is given in Section V.

II. The Proposed AE-BP Algorithm

In the remaining sections, we assume BPSK modulation, which maps a codeword $\boldsymbol{c} = (c_1, c_2, ..., c_N) \in F_2^N$ into a modulated sequence $\boldsymbol{x} = (x_1, x_2, ..., x_N)$ according to $x_n = (-1)^{c_n}$, for $n = 1, 2, ..., N$. Then $\boldsymbol{x}$ is transmitted over AWGN channel. The received value corresponding to

x_n is $y_n = x_n + v_n$, where v_n is a random Gaussian variable with zero mean and δ^2 variance. With the parity check $H_{M\times N}$, we denote the set of bits that participate in check m as $\mathcal{N}(m) = \{n|H_{mn} = 1, 1 \le n \le N\}$. Similarly, we denote the set of checks in which bit n participates by $\mathcal{M}(n) = \{m|H_{mn} = 1, 1 \le m \le M\}$. Other notations used in AE-BP algorithm are defined as follows.

• $P_n^{(0)}$ denotes the LLR of bit n derived from the received value y_n, and the value is given by

$$P_n^{(0)} \equiv ln\frac{Pr(c_n = 0|y_n)}{Pr(c_n = 1|y_n)} = \frac{2y_n}{\delta^2}. \quad (1)$$

• $q_n^{(t)}$ denotes the posteriori LLR of bit n at iteration t.

• $q_{mn}^{(t)}$ denotes the LLR of bit n which is sent from variable node n to check node m at iteration t. It is defined as the log-likelihood ratio that the n-th bit of the input data c_n has the value 0 versus 1, given the information obtained via the check nodes other than check node m at iteration t.

• $r_{mn}^{(t)}$ denotes the LLR of bit n which is sent from check node m to variable node n at iteration t. It is defined as the log-likelihood ratio that the check node m is satisfied when the n-th bit of the input data c_n is fixed to value 0 versus value 1 and the other bits are independent with log-likelihood ratios $q_{mn'}^{(t-1)}$, $n' \in \mathcal{N}(m)\backslash n$.

• $CFT_{mn}^{(t)}$ denotes the continuous fluctuant time of variable node n in check m at iteration t.

LLR-BP contains two main steps, horizontal scan and vertical scan. In horizontal scan the check node messages are computed, and the main operation is multiplication. In vertical scan the variable node messages are computed, and the main operation is addition. The iterative message passing decoding process is described by using the model of factor graph in [5].

In our algorithm, we introduce a counter CFT_{mn} to record the continuous fluctuant time for each q_{mn}. At the beginning, the CFT_{mn} is initialized as zero. In the iterative process, CFT_{mn} will be determined by the signs of variable node messages in two consecutive iterations. If the variable node LLR of current iteration is different from the previous iteration, its CFT will increase by one. The details of AE-BP algorithm are described as follows.

Input: *Received vector* $\mathbf{y} = (y_1, y_2, ...y_N)$,
maximum iteration T.
Output: *Codeword* $\hat{\mathbf{c}} = (\hat{c_1}, \hat{c_2}, ..., \hat{c_N})$.

Initialization: For each n and each $m \in \mathcal{M}(n)$,

$$q_{mn}^{(0)} = P_n^{(0)}, CFT_{mn}^{(0)} = 0, t = 1.$$

Iteration

1) *Horizontal scan (updating the check node messages):*
For each m and each $n \in \mathcal{N}(m)$,

$$r_{mn}^{(t)} = 2tanh^{-1}((\prod_{n' \in \mathcal{N}(m)\backslash n} tanh(\frac{1}{2}q_{mn'}^{(t-1)}))). \quad (2)$$

2) *Vertical scan (updating the variable node messages):*
For each n and each $m \in \mathcal{M}(n)$,

$$tmp = P_n^{(0)} + \sum_{m' \in \mathcal{M}(n)\backslash m} r_{m'n}^{(t)}.$$

$$CFT_{mn}^{(t)} = \begin{cases} CFT_{mn}^{(t-1)} + 1, & sgn(tmp) \neq sgn(q_{mn}^{(t-1)}); \\ 0, & otherwise. \end{cases} \quad (3)$$

$$q_{mn}^{(t)} = \begin{cases} 0, & CFT_{mn}^{(t)} \ge 2; \\ q_{mn}^{(t-1)} + tmp, & CFT_{mn}^{(t)} = 1; \\ tmp, & otherwise. \end{cases} \quad (4)$$

For each n,

$$q_n^{(t)} = P_n^{(0)} + \sum_{m \in \mathcal{M}(n)} r_{mn}^{(t)}.$$

3) *Decoding and stopping criterion test:*
a) Estimate every bit $\hat{c_n}$ $(n = 1, 2, ..., N)$, such that

$$\hat{c_n} = \begin{cases} 0, & q_n^{(t)} > 0; \\ 1, & otherwise. \end{cases}$$

b) If $H_{M\times N}\hat{\mathbf{c}} = 0$, then $\hat{\mathbf{c}}$ is regarded to be a valid codeword and the decoding process terminates; if the number of iteration exceeds T, the final estimated codeword $\hat{\mathbf{c}}$ will be considered as the solution and the decoding process terminates; otherwise increment t and continue from step 1).

In the condition of equation (3), the sign of tmp is different from that of $q_{mn}^{(t-1)}$ means the product of them is negative, which excludes the case where tmp or $q_{mn}^{(t-1)}$ is zero. By the condition of equation (4) we erase $q_{mn}^{(t)}$ if and only if $CFT_{mn}^{(t)}$ is lager than one, allowing other nodes to re-initialize this variable node LLR. In the case $CFT_{mn}^{(t)} = 1$, however, because we cannot determine which message (of tmp and $q_{mn}^{(t-1)}$) is more reliable, and we assign the sum of them to damp the influence. When the $CFT_{mn}^{(t)}$ is zero, which means the sign of $q_{mn}^{(t-1)}$ is the same as tmp, we just assign tmp to $q_{mn}^{(t)}$.

III. Performance Analysis and Experimental Results

A. Performance Analysis

Density evolution was first proposed in [9] to calculate the capacity of LDPC codes. Gaussian approximation (all-Gaussian approximation method) was developed to approach the capacity quickly in [10]. In [11] semi-Gaussian approximation method was used to approach the capacity more accurately compared with all-Gaussian approximation method. In summary, Gaussian method is a good approach for tracking the error probability of message passing decoding based on Gaussian density.

[10] showed that the *symmetry condition* can be treated as $f(x) = e^x f(-x)$, where $f(x)$ is the density of the LLR message. It means that, under sum-product decoding, the probability distribution function (pdf) of LLR intrinsic messages remains symmetric if the pdf satisfies this condition. It is obvious that a Gaussian pdf with mean m and variance δ^2 is symmetric if and only if $2m = \delta^2$. Therefore we need only one parameter m in the following analysis. Without loss of generality, we assume that all-zero codeword is sent according to *Lemma 1* in [9]. Hence, the error probability $P_e^{(t)}$ at iteration t is the average probability that the variable node messages are negative. Let m_0 and $m_q^{(t)}$ be the means of extrinsic channel messages and variable node messages respectively. According to equation (1), we have

$$m_0 = E[P_n^{(0)}] = \frac{2}{\delta^2},\ VAR[P_n^{(0)}] = \frac{4}{\delta^2},$$

which imply that the extrinsic channel LLR has a symmetric Gaussian distribution and its pdf has the form

$$f_m(x) = \frac{1}{\sqrt{4\pi m}} e^{-\frac{(x-m)^2}{4m}}, \tag{5}$$

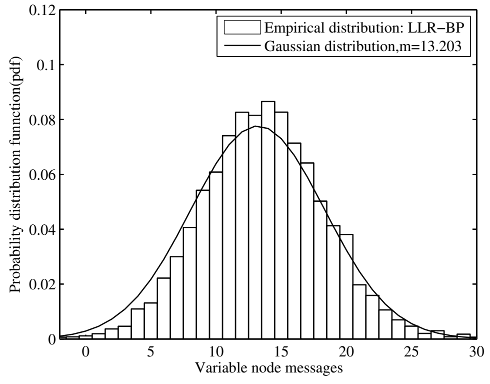
where $m = m_0$.

In the following, we use semi-Gaussian approximation method to compute the average error probability $P_e^{(t)}$ of LLR-BP (and of AE-BP). Semi-Gaussian approximation method abandons the Gaussian assumption at the output of check nodes, and it only assumes that the output LLRs of variable nodes have a symmetric Gaussian distribution. Fig. 1(a) and Fig. 1(b) show our empirical results of the variable node messages by using a (3,6)-regular LDPC code with semi-Gaussian approximation method. From the two figures, in both LLR-BP and AE-BP algorithms the empirical distribution of variable node messages approximates well to the Gaussian density.

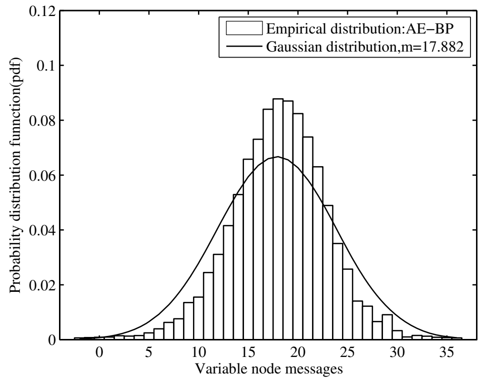
According to the assumption and analysis above, the output LLRs distribution of variable nodes at iteration t can be seen as a symmetric Gaussian with the pdf in equation (5), where $m = m_q^{(t)}$. The average error at iteration t is given by

$$P_e^{(t)} = \int_{-\infty}^{0} f_{m_q^{(t)}}(x)dx. \tag{6}$$

According to equation (6), we compute the average error probability of regular LDPC codes of various code rates and give the results in Table II. It can be seen that AE-BP has lower average error probability than LLR-BP according to semi-Gaussian approximation.



(a) LLR-BP algorithm.



(b) AE-BP algorithm.

Figure 1. Empirical probability density vs. Gaussian density after 5 iterations ($E_b/N_0 = 3$ DB).

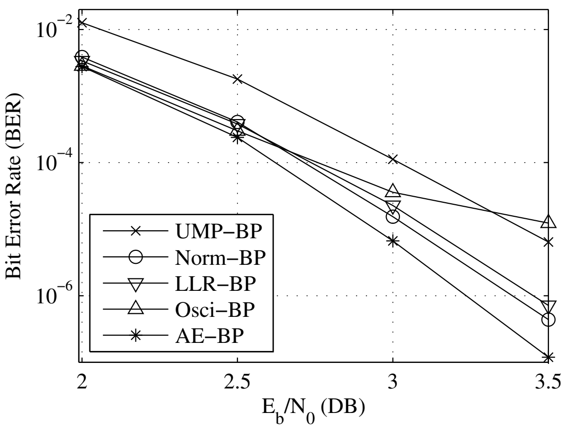
Table II
AVERAGE ERROR PROBABILITY($E_b/N_0 = 2$ DB, ITERATION = 100).

d_v	d_c	Code rate	LLR-BP		AE-BP	
			m_q	P_e	m_q	P_e
3	4	1/4	10.3201	.0114	11.7493	.0076
4	6	1/3	7.0855	.0294	12.2993	.0063
3	6	1/2	14.6546	.0034	16.5747	.0020
4	8	1/2	14.0154	.0041	22.3881	.0004
3	9	2/3	10.2129	.0119	11.4919	.0083
3	12	3/4	6.2647	.0383	6.2753	.0382

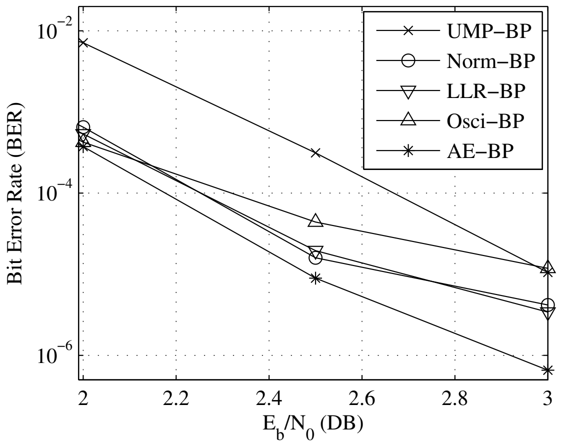
B. Experimental Results

There are two main parts in our simulations. The former focuses on the regular LDPC code, and the latter focuses on the irregular LDPC code. Both of them compute the general BER by using various decoding algorithms including UMP-BP, Norm-BP, LLR-BP, Osci-BP and AE-BP. In all simulations the maximum iteration is set as 100.

Fig. 2(a) and Fig. 2(b) show the BER performances for the (504, 252) and (1008, 504) regular LDPC codes with variable node degree 3 and check node degree 6. For the Norm-BP algorithm, we set the normalization factor $\alpha = 1.25$ [7]. One can see at BER $= 10^{-5}$, about 0.1- and 0.2-



(a) The (504, 252) regular LDPC code.



(b) The (1008, 504) regular LDPC code.

Figure 2. Error performances for iterative decoding of regular LDPC code.

DB coding gain can be achieved by our AE-BP algorithm compared to LLR-BP.

Fig. 3 illustrates the error performance of (504, 252) irregular LDPC code with $\lambda(x) = 0.308x+0.238x^2+0.454x^3$ and $\rho(x) = 0.068x^4+0.624x^5+0.264x^6+0.044x^7$, whose column and row weight are 2, 3, 4 and 5, 6, 7, 8 respectively. From the figure, one notes that the performance of AE-BP outperforms the other decoding methods. Moreover, the performance gap between the AE-BP and LLR-BP decoders is about 0.05- and 0.1-DB at BER = 10^{-5}. As the E_b/N_0 goes high, AE-BP decoder converges to the LLR-BP slowly and still has better performance than other schemes.

According to the simulation results, we can argue that for both regular and irregular LDPC codes the proposed decoding algorithm outperforms existing methods.

IV. Conclusion

In this paper, we presented the AE-BP algorithm to offset the influence of unreliable variable node messages during the

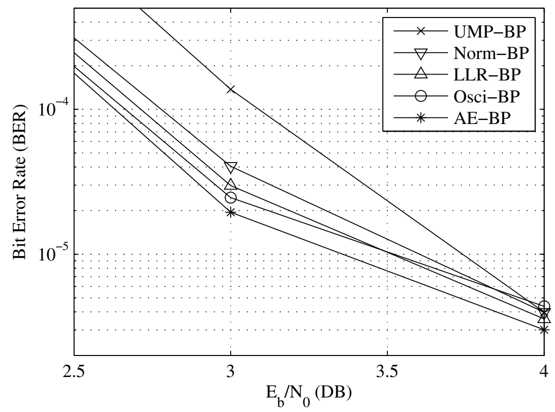


Figure 3. Error performances for the (504, 252) irregular LDPC code.

iterative decoding. In the scheme we specified a sequence of counters to record the behaviors of variable node LLRs and erased the messages adaptively. Analysis and simulations showed that the proposed algorithm performs better than other BP-based schemes for both regular and irregular LDPC codes.

Acknowledgment

The authors would like to thank the reviewers for their comments that help improve this manuscript. This work is supported by NSFC 60703031, 60673076, 60672068, 60673077 and 863 program 2008AA01Z403.

References

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[3] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399–431, Mar. 1999.

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

[5] F. R. Kschischang, B. J. Frey, and H. andrea Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[6] M. P. C. Fossorier, M. Mihaljević, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, 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 Trans. Commun.*, vol. 50, pp. 406–414, Mar. 2002.

[8] S. Gounai, T. Ohtsuki, and T. Kaneko, “Modified belief propagation decoding algorithm for low-density parity check code based on oscillation,” *Vehicular Technology Conference*, vol. 3, pp. 1467–1471, 2006.

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

[10] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[11] M. Ardakani and F. R. Kschischang, “A more accurate one-dimensional analysis and design of irregular ldpc codes,” *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.