AN IMPROVED LOG-DOMAIN BELIEF PROPAGATION ALGORITHM OVER GRAPHS WITH SHORT CYCLES

Nithin Raveendran[1], Shayan G. Srinivasa[2], Bane Vasić[1]

[1]Department of Electrical and Computer Engineering,

University of Arizona, Tucson, AZ 85721, USA.

[2]Department of Electronic Systems Engineering

Indian Institute of Science, Bangalore, India, 560012.

{nithin, vasic}@email.arizona.edu, shayan.gs@dese.iisc.ernet.in

ABSTRACT

We present a modified belief propagation (BP) algorithm for decoding low density parity check codes having graphs with short cycles. The modified algorithm in log domain is superior in terms of numerical stability, precision, computational complexity and ease of implementation when compared to the algorithm in the probability domain. Simulation results show improvement in decoding performance for the modified BP compared to the original algorithm. The modified approach is also generalized for graphs with isolated cycles of arbitrary length by considering the statistical dependency among messages passed in such cycles.

INTRODUCTION

Low density parity check (LDPC) [1] codes have been actively studied in the past several years due to their superlative performance under iterative decoding techniques. To further reduce the gap to Shannon capacity, we need to deal with practically challenging constraints like the use of ultra-large block lengths, random codes, and an optimal decoding algorithm. Though widely used in practice, LDPC codes decoded with belief propagation (BP) algorithm suffer from performance degradation at high SNRs. This error floor phenomenon is normally tackled by using controlled code construction techniques resulting in Tanner graphs with high girth, and by avoiding graphical structures like trapping sets and absorbing sets [2], [3]. Zhang and Siegel have looked into non-uniform quantization techniques [4] to overcome error floors. However, these methods breakdown when Tanner graph have numerous length-4 cycles. The introduction of short cycles into a Tanner graph degrades the performance of the decoding algorithm, thus leading to significant error floors. The general approach of LDPC code design is to avoid short cycles and work with higher girth ensembles. This compensates for the sub-optimality of the algorithm quite effectively. However, to obtain the full potential of a sparse parity check matrix, such restrictions must be avoided. Our aim is to look at these graphical structures from an algorithmic perspective to gain deeper theoretical understanding.

Message passing algorithms over graphs with cycles have been studied in the past. The analysis of the loopy belief propagation (BP) algorithm over graphs with a single cycle [5] showed that the correct maximum *a-posteriori* (MAP) estimates can be obtained for a category of graphical structures. Guyader and Fabre have investigated decoding over short cycles [6] using the idea of conditioning with respect to a variable node. A modification to the BP algorithm based on joint check node decoding was suggested in [7]. The joint check node message is computed using the forward-backward algorithm on a joint trellis section. However, the variable node update remains the same as in the usual BP. Also, the message passing schedule of BP and its variants affects the decoding performance of LDPC codes [8]. In the flooding approach to BP, we perform a simultaneous update of all the messages in the Tanner graph. In the sequential approach like the layered belief propagation, the message passing order is determined based on some predefined order. An informed dynamic scheduling (IDS) is proposed in [9], where the present state of messages in the graph is used to dynamically update the schedule. A performance improvement in IDS over the flooding and the layered BP comes with increase in computational complexity. However, these informed scheduling strategies suffer from performance degradation when the graphs have cycles of length 4. This observation also calls for the analysis of the dependency among short cycles. The algorithm gains more significance when such short cycles cannot be avoided in quantum LDPC code constructions, as observed in [10], wherein the cycles are tackled by using randomness. In [11], the dependency among messages passed within short cycles in a Tanner graph is analyzed to understand the fundamental limitations of the algorithm. The dependency within a cycle of length 4 is understood by looking back at iterative time steps, leading to a modified variable node update in the probability domain [11] and the modified check node update. The comparison of decoders show performance improvement of the modified sum-product algorithm [12], [13] over BP. The BP as well as the modified BP in the probability domain encounters certain drawbacks due to numerical instability. These can be overcome by using the modified algorithm in the log domain as proposed in this paper.

This paper is organized as follows. In Section II, we give a quick outline of the modified BP in the probability domain along with the motivation to look at the log domain implementation. Then, we present the modified BP in the log domain. The simulation curve showing a comparison of decoding performance with the original BP is also presented. In Section III, a generalized version of the algorithm for isolated cycles of girth g is presented. In Section IV, we present the conclusions and future works.

MODIFIED BELIEF PROPAGATION ALGORITHM

In this section, we first give a quick outline of the modified BP in the probability domain along with the notations used. Then, we propose an exact log domain version of the modified algorithm in order to overcome the numerical instability issues associated with the algorithm in the probability domain.

A. Notations:

The Tanner graph consists of variable nodes $v_i,\ i \in \{1,\dots,n\}$ and check nodes $c_j,\ j \in \{1,\dots,m\}$. Let a binary codeword $x = \{x_1, x_2, \dots, x_n\}$ be transmitted over a noisy channel. Denote y_i as the received bit at the variable node v_i and the channel reliability be $p_i = \Pr(x_i^{(0)} =$

$0 \mid y_i)$ and $\bar{p}_i = \Pr(x_i^{(0)} = 1 \mid y_i)$. Using the same notations as in the sum-product algorithm [11], we denote $v_{ij}^{(t)}(b)$ as the message passed from the variable node v_i to the check node c_j at time t. This is the probability that ${x_i}^{(t)} = b$ given the channel and the extrinsic check node messages, b taking binary value 0 or 1. Similarly, $c_{ji}^{(t)}(b)$ denotes the message from the check node c_j to the variable node v_i at time t.

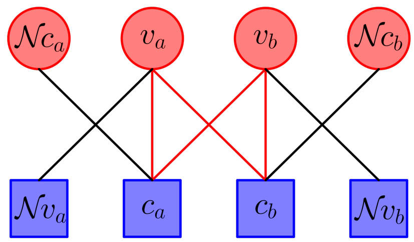

Figure 1: The sub graph shows an isolated cycle of length 4: $v_a - c_a - v_b - c_b - v_a$ (marked in red). The neighboring nodes are labeled as per their connections to the cycle nodes.

Suppose the Tanner graph has an isolated short cycle-$v_a - c_a - v_b - c_b - v_a$ as shown in Fig. 1. The modified BP considers the dependency among messages passed within the short cycle. The neighboring nodes are categorized into smaller sets. The set of check nodes connected to v_a excluding c_a and c_b is denoted as $\mathcal{N}v_a$. The set of variable nodes connected to c_a excluding v_a and v_b is denoted as $\mathcal{N}c_a$. Likewise we have $\mathcal{N}v_b$ and $\mathcal{N}c_b$. The messages passed from these sets to the nodes part of short cycle are considered to be weakly dependent with each other and with respect to messages passed among the nodes in cycle set.

B. Modified BP in Probability Domain:

The modified check node update from cycle node c_a to its neighboring node is as follows:

$$\begin{aligned} c_{ak}^{(t+1)}(0) = {} & h_0^{(t)}(\mathcal{N}c_a \backslash v_k)\Big[\Pr(v_{aa}^{(t)} = 0, v_{ba}^{(t)} = 0) + \Pr(v_{aa}^{(t)} = 1, v_{ba}^{(t)} = 1)\Big] \\ & + h_1^{(t)}(\mathcal{N}c_a \backslash v_k)\Big[\Pr(v_{aa}^{(t)} = 0, v_{ba}^{(t)} = 1) + \Pr(v_{aa}^{(t)} = 1, v_{ba}^{(t)} = 0)\Big], \end{aligned} \tag{1}$$

where

$$\begin{aligned} h_0^{(t)}(\mathcal{N}c_a \backslash v_k) &= \frac{1 + \prod_{v_i \in \mathcal{N}c_a \backslash v_k} (2v_{ia}^{(t)}(0) - 1)}{2}. \\ h_1^{(t)}(\mathcal{N}c_a \backslash v_k) &= 1 - h_0^{(t)}(\mathcal{N}c_a \backslash v_k). \end{aligned} \tag{2}$$

The joint probability terms of the form $\Pr(v_{aa}^{(t)}, v_{ba}^{(t)})$ in eq. (1) are evaluated by tracking the messages back in time as follows:

$$\Pr(v_{aa}^{(t)} = 0, v_{ba}^{(t)} = 0) = p_a p_b \, h_0^{(t-1)}(\mathcal{N}c_b) \prod_{\substack{c_l \in \mathcal{N}v_a \\ c_k \in \mathcal{N}v_b}} c_{la}^{(t)}(0) c_{kb}^{(t)}(0). \tag{3}$$

$$\Pr(v_{aa}^{(t)}\!=\!1, v_{ba}^{(t)}\!=\!1) = \bar{p}_a\ \bar{p}_b\ h_0^{(t-1)}(\mathcal{N}c_b) \prod_{\substack{c_l \in \mathcal{N}v_a \\ c_k \in \mathcal{N}v_b}} c_{la}^{(t)}(1) c_{kb}^{(t)}(1). \tag{4}$$

$$\Pr(v_{aa}^{(t)}\!=\!0, v_{ba}^{(t)}\!=\!1) = p_a\ \bar{p}_b\ h_1^{(t-1)}(\mathcal{N}c_b) \prod_{\substack{c_l \in \mathcal{N}v_a \\ c_k \in \mathcal{N}v_b}} c_{la}^{(t)}(0) c_{kb}^{(t)}(1). \tag{5}$$

$$\Pr(v_{aa}^{(t)}\!=\!1, v_{ba}^{(t)}\!=\!0) = \bar{p}_a\ p_b\ h_1^{(t-1)}(\mathcal{N}c_b) \prod_{\substack{c_l \in \mathcal{N}v_a \\ c_k \in \mathcal{N}v_b}} c_{la}^{(t)}(1) c_{kb}^{(t)}(0). \tag{6}$$

Substituting the equations (3) - (6) in Eq. (1), we have the modified check node update for the cycle check node c_a.

The modified variable node update from cycle node v_a to its neighboring node is as follows:

$$v_{aj}^{(t)}(0) = p_a\ \Pr(c_{aa}^{(t)} = 0, c_{ba}^{(t)} = 0) \prod_{c_k \in \mathcal{N}v_a \backslash c_j} c_{ka}^{(t)}(0), \tag{7}$$

where the joint probability term $\Pr(c_{aa}^{(t)} = 0, c_{ba}^{(t)} = 0)$ is computed by evaluating the messages in previous time steps and thereby unfurling the dependencies within the cycle. Let us denote it by $\gamma_{v_a=0}^{(t)}$.

$$\gamma_{v_a=0}^{(t)} = h_0^{(t-1)}(\mathcal{N}c_a)\ h_0^{(t-1)}(\mathcal{N}c_b) D_b^{(t-1)}(0)\ + h_1^{(t-1)}(\mathcal{N}c_a)\ h_1^{(t-1)}(\mathcal{N}c_b) D_b^{(t-1)}(1). \tag{8}$$

The decision metrics $D_b^{(t)}(0)$ and $D_b^{(t)}(1)$ are used at the variable nodes to make a decision on the value of the bit. It is computed by using the channel reliability and all the incoming check node messages. Also, we normalize the probability term representing the probability of even number of ones among both the sets $\mathcal{N}c_a$ and $\mathcal{N}c_b$ using $H_{(\mathcal{N}c_a,\mathcal{N}c_b)}^{(t-1)}(0)$, and odd number of ones using $H_{(\mathcal{N}c_a,\mathcal{N}c_b)}^{(t-1)}(1)$. Substituting these normalized probability values in the expressions for γ, we have

$$\gamma_{v_a=0}^{(t)} \!=\! H_{(\mathcal{N}c_a,\mathcal{N}c_b)}^{(t-1)}(0) D_b^{(t-1)}(0) + H_{(\mathcal{N}c_a,\mathcal{N}c_b)}^{(t-1)}(1) D_b^{(t-1)}(1). \tag{9}$$

We observe that the direct implementation of the sum-product algorithm in the probability domain has several drawbacks namely, numerical instability and precision issues, increased complexity and quantization effects. To overcome these, we prefer using log-likelihood ratios (LLRs). This scheme is mathematically equivalent to the direct implementation as well has the following advantages:

- The message multiplications are simplified to message additions in the log-BP implementation. This increases the efficiency of the algorithm in case of fixed point implementations [14]. Also, the normalization step is not needed in the log-BP implementation.

- The BP in the probability domain is more sensitive to the quantization effects and requires more quantization levels than the BP in the log domain [15].

In this paper, we show that the probability terms in the modified algorithm can be converted into LLRs. The algebraic modifications in order to achieve this goal are described in the following section.

C. *Modified BP in Log Domain*

First, we focus on the computation of the modified check node update in the log domain. Using the $\tanh(.)$ function, we can express Eq. (1) into LLRs as follows:

$$Lc_{ak}^{(t+1)} = 2\tanh^{-1}\left[\tanh\left(\frac{1}{2}\log\Big(\frac{h_0^{(t)}(\mathcal{N}c_a\backslash v_k)}{h_1^{(t)}(\mathcal{N}c_a\backslash v_k)}\Big)\right)\right.$$
$$\left.\tanh\left(\frac{1}{2}\log\left(\frac{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=0)+\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)}{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=1)+\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)}\right)\right)\right]. \quad (10)$$

Using a simplified notation, Eq. (10) is rewritten as

$$Lc_{ak}^{(t+1)} = 2\tanh^{-1}\left[\tanh\left(\frac{1}{2}L_{\mathcal{N}c_a\backslash v_k}^{(t)}\right)\ \tanh\left(\frac{1}{2}L\gamma_{c_a}^{(t)}\right)\right]. \quad (11)$$

In Eq. (11), there are two $\tanh(.)$ terms. Evaluation of $L_{\mathcal{N}c_a\backslash v_k}^{(t)}$ is done using the same approach as in the well known BP log domain implementation. The second $\tanh(.)$ term has the following form.

$$L\gamma_{c_a}^{(t)} = \log\left(\frac{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=0)+\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)}{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=1)+\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)}\right).$$

Using the algebraic manipulations as described in detail in Appendix D., we convert the terms in the expression for $L\gamma_{c_a}^{(t)}$ into log-likelihood ratios and substitute back in Eq. (11) to get the modified check node update in the log domain. Now, we focus on the modified variable node update conversion starting from Eq. (7).

$$v_{ak}^{(t)}(0) = p_a \Pr(c_{aa}^{(t)}=0, c_{ba}^{(t)}=0) \prod_{c_j\in\mathcal{N}v_a\backslash c_k} c_{ja}^{(t)}(0), \quad (12)$$

Similarly, we can express probability of message from $v_a = 1$ as follows:

$$v_{ak}^{(t)}(1) = \bar{p}_a \Pr(c_{aa}^{(t)}=1, c_{ba}^{(t)}=1) \prod_{c_j\in\mathcal{N}v_a\backslash c_k} c_{ja}^{(t)}(1), \quad (13)$$

Observe that the joint terms are indeed the γ functions $\gamma_{v_a=0}^{(t)}$ and $\gamma_{v_a=1}^{(t)}$ respectively. Now, dividing Eq. (12) by Eq. (13) and taking logarithm on both sides, we have as per our notations:

$$Lv_{ak}^{(t)} = l_a + \log\left(\frac{\gamma_{v_a=0}^{(t)}}{\gamma_{v_a=1}^{(t)}}\right) + \sum_{c_j\in\mathcal{N}v_a\backslash c_k} Lc_{ja}^{(t)}. \quad (14)$$

Denoting $\log\left(\frac{\gamma_{v_a=0}^{(t)}}{\gamma_{v_a=1}^{(t)}}\right)$ by $L\gamma_{v_a}^{(t)}$, we have

$$Lv_{ak}^{(t)} = l_a + L\gamma_{v_a}^{(t)} + \sum_{c_j\in\mathcal{N}v_a\backslash c_k} Lc_{ja}^{(t)}. \quad (15)$$

We evaluate $L\gamma_{v_a}^{(t)}$ by using the equivalent $\tanh(.)$ expression in the log domain for Eq. (9).

$$L\gamma_{v_a}^{(t)} = 2\tanh^{-1}\left[\tanh\left(\frac{1}{2}LH_{\mathcal{N}c_a\mathcal{N}c_b}^{(t-1)}\right)\,\tanh\left(\frac{1}{2}LD_b^{(t-1)}\right)\right], \tag{16}$$

where $LH_{\mathcal{N}c_a\mathcal{N}c_b}^{(t-1)} = \log\left(\frac{H_{(\mathcal{N}c_a,\mathcal{N}c_b)}^{(t-1)}(0)}{H_{(\mathcal{N}c_a,\mathcal{N}c_b)}^{(t-1)}(1)}\right)$. We can express $LH_{\mathcal{N}c_a\mathcal{N}c_b}^{(t-1)}$ in terms of LLRs as follows:

$$LH_{\mathcal{N}c_a\mathcal{N}c_b}^{(t-1)} = LH_{\mathcal{N}c_a}^{(t-1)} + LH_{\mathcal{N}c_b}^{(t-1)}. \tag{17}$$

Substituting for value of $L\gamma_{v_a}^{(t)}$ into Eq. (15), we get the modified variable node update in the log domain. Note that the log domain algorithm derived here is *exactly* equivalent to the modified algorithm in the probability domain.

D. *Simulation Results*

In this section, we compare the performance of the log domain modified BP versus the conventional BP. We constrain the H matrix to be regular with a column weight of 3 and a row weight of 6. A parity check matrix of size $(n-k)\times n$ = (100×200) is used. Block length n of the code is 200 with a code rate equal to 0.5. For simulation, we introduce 40 isolated cycles of length 4 in the Tanner graph. We consider a simple memoryless channel with additive white Gaussian noise (AWGN) of variance $N_0/2$. The noise conditions are maintained 'exactly' the same for both the decoding algorithms.

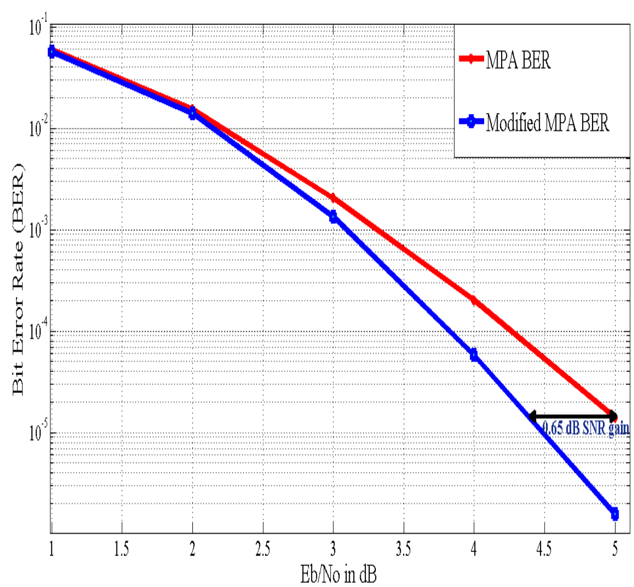

Figure 2: BER vs. E_b/N_0 (in dB) curves comparison between modified algorithm and original MPA in the log domain after 10 iterations on H matrix with size $(n-k)\times n$ = (100×200) with 40 cycles of length 4.

Fig. 2 shows gain at higher SNR for the modified update over the original decoding algorithm. An SNR gain of around 0.65 dB is obtained after 10 iterations at 5 dB SNR with the modified algorithm over the original BP. This motivates us to develop generalized algorithms for Tanner graphs with higher girth.

GENERALIZATION TO GIRTH g

In the modified BP, we considered isolated cycles of length 4. The dependency over time in the computation of joint terms in Eq. (9) and Eq. (16) is analyzed. This leads us to a generalized approach for isolated cycles of higher girth.

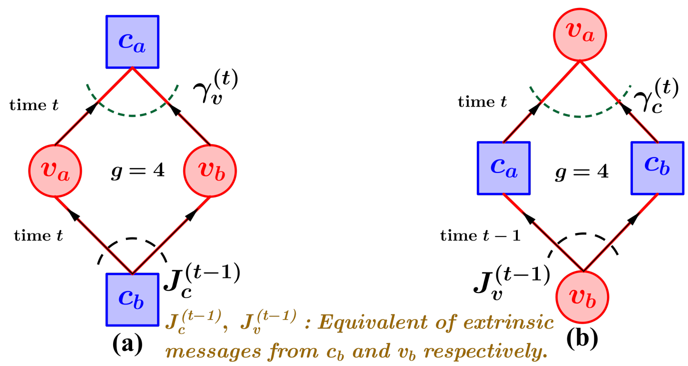

Figure 3: The modified updates at time t in a cycle of length 4. (a) The modified check node update uses the joint message $\gamma_v^{(t)}$ from cycle variable nodes. $\gamma_v^{(t)}$ is computed as a function of $J_c^{(t-1)}$ which represents the extrinsic messages sent from the other check node c_b at time $t-1$. (b) Similarly, the modified variable node update uses the joint message $\gamma_c^{(t)}$, which depends on $J_v^{(t-1)}$.

In the modified algorithm, we observed that tracing back to the $(t-1)^{\text{th}}$ iteration helps in computing the joint messages. In Fig. 3, $J_c^{(t-1)}$ and $J_v^{(t-1)}$ represents the equivalent extrinsic messages sent from the nodes c_b and v_b respectively, computed using the modified algorithm. Fig. 3a and Eq. (9) show that $\gamma_v^{(t)}$ depends on the message $J_c^{(t-1)}$. Similarly, Fig. 3b and Eq. (16) show the dependence of $\gamma_c^{(t)}$ on $J_v^{(t-1)}$. It is noted that the dependency for girth g case is observed from the node which is $g/2$ edges far.

In a similar approach, we can evaluate the joint messages in cycles of length 6. We observe that $\gamma_c^{(t)}$ depends on the message $J_c^{(t-2)}$ from the cycle check node which is $g/2 = 3$ edges far. Similarly, $\gamma_v^{(t)}$ depends on the message $J_v^{(t-1)}$ from the cycle variable node $g/2 = 3$ edges away from the check node. Hence, in girth 6 graphs, overwhelming of information is avoided by using the joint term evaluation[1] tracking back to $J_c^{(t-2)}$ and $J_v^{(t-1)}$.

Summarizing the above results, we can express the joint terms as a function of J_c and J_v. For girth 4 graphs, $\gamma_c^{(t)} = f_c(J_v^{(t-1)})$ and $\gamma_v^{(t)} = f_v(J_c^{(t-1)})$. For girth 6 graphs, $\gamma_c^{(t)} = f_c(J_c^{(t-2)})$ and $\gamma_v^{(t)} = f_v(J_v^{(t-1)})$. Generalizing the above for Tanner graphs with isolated cycles of length g, the joint terms

[1] We omit the details of the derivation due to lack of space.

in the modified algorithm has the following relation with messages J_v and J_c over time.

$$\gamma_v^{(t)} = \begin{cases} f_v(J_c^{(t-g/4)}) & \text{if } \mathrm{mod}(g,4)=0 \\ f_v(J_v^{(t-\lfloor \frac{g}{4} \rfloor)}) & \text{else.} \end{cases} \tag{18}$$

$$\gamma_c^{(t)} = \begin{cases} f_c(J_v^{(t-g/4)}) & \text{if } \mathrm{mod}(g,4)=0 \\ f_c(J_c^{(t-\lceil \frac{g}{4} \rceil)}) & \text{else.} \end{cases} \tag{19}$$

CONCLUSIONS AND FUTURE WORKS

We derived a modified BP in the log-domain which shows performance improvement over the conventional BP. We showed that the algorithm can be generalized for isolated cycles of arbitrary girth. Future work needs to be done by considering the effect of nested cycles in the Tanner graph. Also, a unified theory for tractable statistical dependencies during belief propagation over arbitrary graphs has to be developed.

REFERENCES

[1] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[2] D. V. Nguyen, S. K. Chilappagari, B. Vasić, and M. W. Marcellin, "On the construction of structured LDPC codes free of small trapping sets," *IEEE Trans. Inf. Theory*, vol. 58, pp. 2280–2302, Apr. 2012.

[3] L. Dolecek, Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, pp. 181–201, Jan. 2010.

[4] X. Zhang and P. H. Siegel, "Quantized min-sum decoders with low error floor for LDPC codes," in *Proc. Int. Symp. Inf. Theory*, (Boston, MA, USA), pp. 2871–2875, Jul. 2012.

[5] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, 2001.

[6] A. Guyader and E. Fabre, "Dealing with short cycles in graphical codes," in *2000 IEEE Intl. Symp Info. Theory*, pp. 10–, June 2000.

[7] Kyuhyuk Chung and Jun Heo, "Improved belief propagation decoding for LDPC codes with a large number of short cycles," in *IEEE 63rd VTC*, vol. 3, pp. 1464–1466, May 2006.

[8] N. Raveendran and B. Vasic, "Trapping set analysis of horizontal layered decoder," in *IEEE Intl. Conf. Comm., Kansas City, MO, USA, May 20-24*, pp. 1–6, 2018.

[9] A. I. V. Casado, M. Griot, and R. D. Wesel, "LDPC decoders with informed dynamic scheduling," *IEEE Trans. Commun.*, vol. 58, pp. 3470–3479, Dec. 2010.

[10] N. Raveendran, P. Nadkarni, S. Srinivasa, and B. Vasić, "Stochastic resonance decoding for quantum LDPC codes," in *Proc. IEEE Int. Conf. Comm.*, (Paris, France), May 21 – 25 2017.

[11] N. Raveendran and S. G. Srinivasa, "An analysis into the loopy belief propagation algorithm over short cycles," *Proc. IEEE. Intl. Conf. Comm.*, pp. 2009–2014, June 2014.

[12] N. Raveendran and S. G. Srinivasa, "A modified sum-product algorithm over graphs with isolated short cycles," *Proc. IEEE. Intl. Symp. Info. Theory*, vol. 54, pp. 2619–2623, June 2014.

[13] N. Raveendran, "A modified sum-product algorithm over graphs with short cycles," *M.S. thesis, http://etd.iisc.ac.in/handle/2005/3847*, 2018.

[14] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Computational complexity and quantization effects of decoding algorithms for non-binary LDPC codes," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, vol. 4, p. 4, May 2004.

[15] Li Ping and W. K. Leung, "Decoding low density parity check codes with finite quantization bits," *IEEE Commun. Letters*, vol. 4, pp. 62–64, Feb 2000.

APPENDIX

A. Computing $\mathbf{L\gamma_{c_a}^{(t)}}$ *in check node update*

$$Lc_{ak}^{(t+1)} = 2\tanh^{-1}\left[\tanh\left(\frac{1}{2}L_{\mathcal{N}c_a\backslash v_k}^{(t)}\right)\ \tanh\left(\frac{1}{2}L\gamma_{c_a}^{(t)}\right)\right]. \tag{1}$$

In Eq. (1), there are two $\tanh(.)$ terms. Evaluation of $L_{\mathcal{N}c_a\backslash v_k}^{(t)}$ is done using the same approach as in the well known BP log domain implementation. The second $\tanh(.)$ term has the following form.

$$L\gamma_{c_a}^{(t)} = \log\left(\frac{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=0) + \Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)}{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=1) + \Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)}\right).$$

We divide and multiply the numerator term by $\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)$ and the denominator term by $\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)$. Now, we have

$$L\gamma_{c_a}^{(t)} = \log\left(\frac{\dfrac{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=0)}{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)}+1}{\dfrac{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=1)}{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)}+1}\right) + \log\left(\frac{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)}{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)}\right). \tag{2}$$

Right hand side of Eq. (2) can be rewritten as

$$\log\left[\frac{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=0)}{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)}+1\right] - \log\left[\frac{\Pr(v_{aa}^{(t)}=0, v_{ba}^{(t)}=1)}{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)}+1\right] + \log\left[\frac{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=1)}{\Pr(v_{aa}^{(t)}=1, v_{ba}^{(t)}=0)}\right]. \tag{3}$$

The ratio terms in Eq. (3) are converted into LLRs using the following approach.

- **LLR expression for** $\dfrac{\mathrm{Pr}(\mathbf{v_{aa}^{(t)}}=\mathbf{0},\mathbf{v_{ba}^{(t)}}=\mathbf{0})}{\mathrm{Pr}(\mathbf{v_{aa}^{(t)}}=\mathbf{1},\mathbf{v_{ba}^{(t)}}=\mathbf{1})}$:

Logarithm of the term is evaluated using Eq. (3) and Eq. (4).

$$\log\left(\frac{\mathrm{Pr}(v_{aa}^{(t)}=0,v_{ba}^{(t)}=0)}{\mathrm{Pr}(v_{aa}^{(t)}=1,v_{ba}^{(t)}=1)}\right)=\log\left(\frac{p_a}{\bar{p}_a}\frac{p_b}{\bar{p}_b}\frac{\prod_{\substack{c_l\in\mathcal{N}v_a\\c_k\in\mathcal{N}v_b}}c_{la}^{(t)}(0)c_{kb}^{(t)}(0)}{\prod_{\substack{c_l\in\mathcal{N}v_a\\c_k\in\mathcal{N}v_b}}c_{la}^{(t)}(1)c_{kb}^{(t)}(1)}\right) \tag{4}$$

Simplifying Eq. (4) by using the notations used in the log domain BP, we have

$$\log\left(\frac{\mathrm{Pr}(v_{aa}^{(t)}=0,v_{ba}^{(t)}=0)}{\mathrm{Pr}(v_{aa}^{(t)}=1,v_{ba}^{(t)}=1)}\right)=l_a+l_b+\sum_{c_l\in\mathcal{N}v_a}Lc_{la}^{(t)}+\sum_{c_k\in\mathcal{N}v_b}Lc_{kb}^{(t)}. \tag{5}$$

By taking exponent on both sides of Eq. (5), we have

$$\frac{\mathrm{Pr}(v_{aa}^{(t)}=0,v_{ba}^{(t)}=0)}{\mathrm{Pr}(v_{aa}^{(t)}=1,v_{ba}^{(t)}=1)}=e^{l_a+l_b+\sum\limits_{c_l\in\mathcal{N}v_a}Lc_{la}^{(t)}+\sum\limits_{c_k\in\mathcal{N}v_b}Lc_{kb}^{(t)}}. \tag{6}$$

- **LLR expression for** $\dfrac{\mathrm{Pr}(\mathbf{v_{aa}^{(t)}}=\mathbf{0},\mathbf{v_{ba}^{(t)}}=\mathbf{1})}{\mathrm{Pr}(\mathbf{v_{aa}^{(t)}}=\mathbf{1},\mathbf{v_{ba}^{(t)}}=\mathbf{0})}$:

Using the same approach as in Eq. (5) and then taking exponents on both sides, we have

$$\frac{\mathrm{Pr}(v_{aa}^{(t)}=0,v_{ba}^{(t)}=1)}{\mathrm{Pr}(v_{aa}^{(t)}=1,v_{ba}^{(t)}=0)}=e^{l_a-l_b+\sum\limits_{c_l\in\mathcal{N}v_a}Lc_{la}^{(t)}-\sum\limits_{c_k\in\mathcal{N}v_b}Lc_{kb}^{(t)}}. \tag{7}$$

- **LLR expression for** $\log\left[\dfrac{\mathrm{Pr}(\mathbf{v_{aa}^{(t)}}=\mathbf{1},\mathbf{v_{ba}^{(t)}}=\mathbf{1})}{\mathrm{Pr}(\mathbf{v_{aa}^{(t)}}=\mathbf{1},\mathbf{v_{ba}^{(t)}}=\mathbf{0})}\right]$:

Using Eq. (4) and Eq. (6), and taking logarithm of the ratio, we have

$$\log\left[\frac{\mathrm{Pr}(v_{aa}^{(t)}=1,v_{ba}^{(t)}=1)}{\mathrm{Pr}(v_{aa}^{(t)}=1,v_{ba}^{(t)}=0)}\right]=L_{\mathcal{N}c_b}^{(t-1)}-\sum_{c_k\in\mathcal{N}v_b}Lc_{kb}^{(t)}-l_b. \tag{8}$$

Substituting the LLR expressions of the three ratio terms into Eq. (3), we obtain the expression for $L\gamma_{c_a}^{(t)}$ as follows:

$$\begin{aligned}L\gamma_{c_a}^{(t)}&=\log\left[e^{l_a+l_b+\sum\limits_{c_l\in\mathcal{N}v_a}Lc_{la}^{(t)}+\sum\limits_{c_k\in\mathcal{N}v_b}Lc_{kb}^{(t)}}+1\right]-\log\left[e^{l_a-l_b+\sum\limits_{c_l\in\mathcal{N}v_a}Lc_{la}^{(t)}-\sum\limits_{c_k\in\mathcal{N}v_b}Lc_{kb}^{(t)}}+1\right]\\&\quad+L_{\mathcal{N}c_b}^{(t-1)}-\sum_{c_k\in\mathcal{N}v_b}Lc_{kb}^{(t)}-l_b.\end{aligned} \tag{9}$$