MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Improved Min-Sum Decoding of LDPC Codes Using 2-Dimensional Normalization

Juntan Zhang, Marc Fossorier, Daqing Gu, Jinyun Zhang

TR2005-158   June 2006

Abstract

A two-dimensional post normalization scheme is proposed to improve the performance of conventional min-sum (MS) and normalized MS decoding of irregular low density parity check codes. An iterative procedure based on parallel differential algorithm is presented to obtain the optimal two-dimensional normalization factors. Both density evolution analysis and specific code simulation show that the proposed method provides a comparable performance as belief propagation decoding while requiring less complexity. Interestingly, the new method exhibits a lower error floor than that of belief propagation decoding in the high SNR region. With respect to standard MS and one-dimentional normalizated MS decodings, the two-dimensional normailzed MS offers a considerably better performance.

Globecom, November 2005

Improved Min-Sum Decoding of LDPC Codes Using 2-Dimensional Normalization

Juntan Zhang and Marc Fossorier
Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, HI 96822
Email: juntan, marc@spectra.eng.hawaii.edu

Daqing Gu and Jinyun Zhang
Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge, MA 02139
Email: dgu, jzhang@merl.com

Abstract— **A two-dimensional post normalization scheme is proposed to improve the performance of conventional min-sum (MS) and normalized MS decoding of irregular low density parity check codes. An iterative procedure based on parallel differential optimization algorithm is presented to obtain the optimal two-dimensional normalization factors. Both density evolution analysis and specific code simulation show that the proposed method provides a comparable performance as belief propagation decoding while requiring less complexity. Interestingly, the new method exhibits a lower error floor than that of belief propagation decoding in the high SNR region. With respect to standard MS and one-dimensional normalized MS decodings, the two-dimensional normalized MS offers a considerably better performance.**

I. Introduction

Low-density parity-check (LDPC) codes [1] were first introduced in the 1960's and rediscovered in the 1990's. Recently, it was found that the belief propagation (BP) algorithm [2] provides a powerful tool for iterative decoding of LDPC codes. As shown in [3], LDPC codes with iterative decoding based on BP achieve a remarkable error performance that is very close to the Shannon limit [4]. Consequently, LDPC codes have received significant attention recently.

An LDPC code is specified by a parity-check matrix containing mostly zeros and only a small number of ones. In general, LDPC codes can be categorized into regular LDPC codes and irregular LDPC codes. An LDPC code is called regular if the weights of rows and columns in its parity check matrix are equal and is called irregular if not. It has been shown, both theoretically and by simulation, that with properly chosen structure, irregular LDPC codes have better performance than regular ones [5].

To decode LDPC codes, either soft decision, hard decision or hybrid decision decoding can be used. It has been shown that soft decision decoding based on BP is one of the most powerful decoding methods for LDPC codes. Although BP decoding offers good performance, it can become too complex for hardware implementation because of floating point computations. By approximating the calculation at the check nodes with a simple minimum operation, the min-sum (MS) algorithm reduces the complexity of BP [6], [7]. While MS is hardware efficient, its ultimate performance is often much worse than that of BP. It has been observed that the degradation of MS can be compensated by linear post processing (normalization) of the messages delivered by check nodes [8], [9]. The optimal normalization factors can be determined using density evolution to search for the factors which yield the lowest threshold. Simulation results and density evolution analysis show that for decoding regular LDPC codes, normalized MS with a single normalization factor is sufficient to achieve good performance which is near that of BP. For decoding many irregular LDPC codes, however, conventional normalized MS exhibits a large performance degradation compared to that of BP. In this paper, we present a two-dimensional (2-D) normalized MS decoding of irregular LDPC codes. In 2-D normalized MS scheme, belief messages outgoing from both check and bit node processors are normalized. The normalization factor of a check (bit) node processor depends on the degree of that node. To circumvent brute force search, parallel differential optimization and density evolution are used to obtain the 2-D optimal normalization factor pair.

This paper is organized as follows. Section 2 briefly reviews the standard BP decoding. Section 3 and Section 4 analyze density evolution of conventional MS and normalized MS decoding, respectively. Section 5 presents 2-D normalized MS decoding and the corresponding density evolution. Section 6 describes the parallel differential optimization procedure to obtain the optimal 2-D normalization factor pairs. Section 7 reports simulation results and Section 8 concludes this paper.

II. Standard BP

Suppose a regular binary $(N,K)(dv,dc)$ LDPC code $\mathbf{C}$ is used for error control over an AWGN channel with zero mean and power spectral density $N_0/2$. Assume BPSK signaling with unit energy, which maps a codeword $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ into a transmitted sequence $\mathbf{q} = (q_1, q_2, \ldots, q_N)$, according to $q_n = 1 - 2w_n$, for $n = 1, 2, \ldots, N$. If $\mathbf{w} = [w_n]$ is a codeword in $\mathbf{C}$ and $\mathbf{q} = [q_n]$ is the corresponding transmitted sequence, then the received sequence is $\mathbf{q} + \mathbf{g} = \mathbf{y} = [y_n]$, with $y_n = q_n + g_n$, where for $1 \leq n \leq N$, g_n's are statistically independent Gaussian random variables with zero mean and variance $N_0/2$. Let $\mathbf{H} = [H_{mn}]$ be the parity check matrix which defines the LDPC code. We denote the set of bits that participate in check m by $\mathcal{N}(m) = \{n : H_{mn} = 1\}$ and the set of checks in which bit n participates as $\mathcal{M}(n) = \{m : H_{mn} = 1\}$. We also denote $\mathcal{N}(m)\backslash n$ as the set $\mathcal{N}(m)$ with bit n excluded, and $\mathcal{M}(n)\backslash m$ as the set $\mathcal{M}(n)$ with check m excluded. We define the following notations associated with $i - th$ iteration:

- $U_{ch,n}$: The log-likelihood ratios (LLR) of bit n which is derived from the channel output y_n. In BP decoding, we initially set $U_{ch,n} = \frac{4}{N_0} y_n$.
- $U_{mn}^{(i)}$: The LLR of bit n which is sent from check node m to bit node n.
- $V_{mn}^{(i)}$: The LLR of bit n which is sent from the bit node n to check node m.
- $V_n^{(i)}$: The *a posteriori* LLR of bit n computed at each iteration.

The standard LLR BP algorithm is carried out as follows [3]:

```
Initialization: Set $i = 1$, maximum number of iteration to $I_{Max}$. For each $m$,$n$, set $V_{mn}^{(0)} = U_{ch,n}$.
Step 1: (i)  Horizontal Step, for $1 \leq n \leq N$ and each $m \in \mathcal{M}(n)$, process:
             $U_{mn}^{(i)} = 2 \tanh^{-1} \prod_{n' \in \mathcal{N}(m)\backslash n} \tanh \frac{V_{mn'}^{(i-1)}}{2}$   (1)
        (ii) Vertical Step, for $1 \leq n \leq N$ and each $m \in \mathcal{M}(n)$, process:
             $V_{mn}^{(i)} = U_{ch,n} + \sum_{m' \in \mathcal{M}(n)\backslash m} U_{m'n}^{(i)}$
             $V_n^{(i)} = U_{ch,n} + \sum_{m \in \mathcal{M}(n)} U_{mn}^{(i)}$   (2)
Step 2: Hard decision and stopping criterion test:
        (i)  Create $\hat{\boldsymbol{w}}^{(i)} = [\hat{w}_n^{(i)}]$ such that $\hat{w}_n^{(i)} = 1$ if $V_n^{(i)} < 0$, and $\hat{w}_n^{(i)} = 0$ if $V_n^{(i)} \geq 0$.
        (ii) If $\boldsymbol{H}\hat{\boldsymbol{w}}^{(i)} = \mathbf{0}$ or the maximum iteration number $I_{Max}$ is reached, stop the decoding iteration and go to Step 3. Otherwise set $i := i + 1$ and go to Step 1.
Step 3: Output $\hat{\boldsymbol{w}}^{(i)}$ as the decoded codeword.
```

III. MS DECODING

The check node processing in the standard BP decoding may require considerable computational resource and may cause hardware implementation difficulties as well as high decoding delay. BP decoding can be simplified by approximating the calculation at the check nodes with a simple minimum operation, which results in MS decoding [6], [7].

A. MS algorithm

In MS decoding, the bit node operation is the same as in the standard BP. Taking advantage of the odd property of the function tanh(), MS simplifies the updating rule in check nodes by modifying (1) into

$$U_{mn}^{(i)} = \prod_{n' \in \mathcal{N}(m)\backslash n} \operatorname{sgn}\left(V_{mn'}^{(i-1)}\right) \cdot \min_{n' \in \mathcal{N}(m)\backslash n} |V_{mn'}^{(i-1)}| \quad (3)$$

The MS algorithm is much simpler than the BP decoding, since only comparisons and additions are needed in check and bit node processing. The product of the signs in (3) can be easily calculated by modulo 2 addition of the hard decision of all $\{V_{mn'}^{(i-1)} : n' \in \mathcal{N}(m)\backslash n\}$. The minimum magnitude can be found by comparison.

B. Density evolution of MS

Density evolution can be used to track the process of iterative decoding and determine the threshold of an iteratively decodable code when decoded by a certain algorithm [4]. The density evolution of MS has been presented in [10], [11]. By applying a similar analysis approach as [1, Lemma 4.1], in the following we derive it in a different way.

Let $f_V^{(i)}$ and $f_U^{(i)}$ be the pdf's of LLR delivered by bit and check node, respectively. In MS, the density evolution of bit node is the same as that in BP

$$f_V^{(i)} = \mathcal{F}^{-1}\left(\mathcal{F}(f_{U_{ch}}) \cdot \left(\mathcal{F}(f_U^{(i)})\right)^{dc-1}\right)$$

where $\mathcal{F}$ denotes Fourier transform.

To derive the density evolution of check node processor in MS, we can first prove a theorem following a similar spirit as the lemma 4.1 [1, p.41]. Consider J independent random variables $\{X_1, X_2, \ldots, X_J\}$. Let $p_{Xj}(x)$ be the pdf of X_j, for $j = 1, 2, \ldots, J$. Given a nonnegative value a, let $P_{Xj,+}$ and $P_{Xj,-}$ be the probabilities of X_j has magnitude greater or equal than a, and sign $+$ and $-$, respectively; i.e, $P_{Xj,+} = \int_a^{+\infty} p_{Xj}(x)dx$ and $P_{Xj,-} = \int_{-\infty}^{-a} p_{Xj}(x)dx$. Next consider a new random variable

$$Y = \prod_{j=1}^{J} \operatorname{sgn}(X_j) \cdot \min_{j=1}^{J} |X_j|. \quad (4)$$

Similarly define $P_{Y,+}$ and $P_{Y,-}$ as the probabilities that Y has magnitude greater or equal than a, and sign $+$ and $-$, respectively. We have

Lemma 1:

$$P_{Y,+} = \frac{\prod_{j=1}^{J}(P_{Xj,+} + P_{Xj,-}) + \prod_{j=1}^{J}(P_{Xj,+} - P_{Xj,-})}{2} \quad (5)$$

$$P_{Y,-} = \frac{\prod_{j=1}^{J}(P_{Xj,+} + P_{Xj,-}) - \prod_{j=1}^{J}(P_{Xj,+} - P_{Xj,-})}{2} \quad (6)$$

Proof: Consider the function

$$\prod_{j=1}^{J}(P_{Xj,+} + t \cdot P_{Xj,-}) \quad (7)$$

Observe that if this function is expanded as a polynomial in t, the coefficient of t^k is the probability that in

$\{X_1, X_2, \ldots, X_J\}$ there are k negative items and $J-k$ positive items, and each of them has a magnitude greater than or equal to a. The expansion of function

$$\prod_{j=1}^{J}(P_{Xj,+} - t \cdot P_{Xj,-}) \tag{8}$$

is identical except that all the coefficients of t^k with odd power k are the opposite of those in the expansion of (7). Adding (7) and (8), all coefficients of t^k with even power k are doubled, while all the coefficients of t^k with odd power k are cancelled out. Setting $t = 1$ and dividing the summation by 2, the result is the probability that in $\{X_1, X_2, \ldots, X_J\}$, there are even number of items with sign $-$ and magnitude greater than or equal to a. From (4), it is clear that this probability equals the probability that Y has sign $+$ and magnitude greater than or equal to a; therefore we obtain (5). Following a similar approach, it is straightforward to prove (6). ■

If the J independent random variables $\{X_1, X_2, \ldots, X_J\}$ have identical distribution $p_X(x)$, we have

Corollary 2:

$$\begin{aligned} P_{Y,+} &= \frac{(P_{X,+} + P_{X,-})^J + (P_{X,+} - P_{X,-})^J}{2} \\ P_{Y,-} &= \frac{(P_{X,+} + P_{X,-})^J - (P_{X,+} - P_{X,-})^J}{2} \end{aligned}$$

where $P_{X,+} = \int_a^{+\infty} p_X(x)dx$ and $P_{X,-} = \int_{-\infty}^{-a} p_X(x)dx$.

Based on the assumption of infinite codeword length and tree like structure of the graph representation of (dv, dc) regular LDPC codes, the LLR values delivered by check (bit) nodes have the same distribution. The check node processing (3) can be expressed as

$$U^{(i)} = \prod_{j=1}^{dc-1} \text{sgn}(V_j^{(i-1)}) \cdot \min_{j=1}^{dc-1} |V_j^{(i-1)}|. \tag{9}$$

where $(V_1, V_2, \ldots, V_{dc-1})$ are i.i.d. Following similar notations as in [10], for $a > 0$, we define $\psi_+^{(i)}(a) = \int_a^{+\infty} f_V^{(i)}(v)dv$ and $\psi_-^{(i)}(a) = \int_{-\infty}^{-a} f_V^{(i)}(v)dv$. Hence $\psi_+^{(i)}(a)$ and $\psi_-^{(i)}(a)$ are the probabilities that $V^{(i)}$ has magnitude no less than a, and sign $+$ and $-$, respectively. Applying Corollary 2, for $u > 0$, the PDF of $U^{(i)}$ is

$$\begin{aligned} F_U^{(i)}(u) &= 1 - P_r(\text{even number of negative values in} \\ &\quad \{V_j^{(i-1)}\}, \text{ and } |V_j^{(i-1)}| > u, \forall j) \\ &= 1 - \frac{1}{2}\Bigg[\Big(\psi_+^{(i-1)}(u) + \psi_-^{(i-1)}(u)\Big)^{dc-1} \\ &\quad + \Big(\psi_+^{(i-1)}(u) - \psi_-^{(i-1)}(u)\Big)^{dc-1}\Bigg] \end{aligned} \tag{10}$$

Similarly for $u < 0$, the PDF of $U^{(i)}$ is

$$\begin{aligned} F_U^{(i)}(u) &= P_r(\text{odd number of negative values in} \\ &\quad \{V_j^{(i-1)}\}, \text{ and } |V_j^{(i-1)}| > u, \forall j) \\ &= \frac{1}{2}\Bigg[\Big(\psi_+^{(i-1)}(|u|) + \psi_-^{(i-1)}(|u|)\Big)^{dc-1} \\ &\quad - \Big(\psi_+^{(i-1)}(|u|) - \psi_-^{(i-1)}(|u|)\Big)^{dc-1}\Bigg]. \end{aligned} \tag{11}$$

Differentiating (10) and (11), we obtain the pdf of $U^{(i)}$

$$\begin{aligned} f_U^{(i)}(u) &= \frac{dc-1}{2}\Bigg[\Big(f_V^{(i-1)}(u) + f_V^{(i-1)}(-u)\Big) \\ &\quad \cdot \Big(\psi_+^{(i-1)}(|u|) + \psi_-^{(i-1)}(|u|)\Big)^{dc-2} \\ &\quad + \Big(f_V^{(i-1)}(u) - f_V^{(i-1)}(-u))\Big) \\ &\quad \cdot \Big(\psi_+^{(i-1)}(|u|) - \psi_-^{(i-1)}(|u|)\Big)^{dc-2}\Bigg] \end{aligned} \tag{12}$$

IV. Conventional normalized MS

Although the MS algorithm greatly reduces the decoding complexity for implementation, it may cause a significant performance degradation compared to BP decoding. In [8], [9], normalized MS is proposed to improve the performance of standard MS.

A. Normalized MS algorithm

Let U^{BP} and U^{MS} denote the LLR values calculated by BP and MS decoding with (1) and (3), respectively. It was proved that $\text{sgn}(U^{BP}) = \text{sgn}(U^{MS})$ and $|U^{BP}| > |U^{MS}|$ [8], which implies that a linear post processing of U^{MS} could reduce the gap between BP and MS decoding. The normalized MS modifies the check node processing (3) as

$$U_{mn}^i \leftarrow \alpha \cdot U_{mn}^i \tag{13}$$

where α is a normalization factor, $0 < \alpha \leq 1$.

B. Density evolution of normalized MS

By keeping the normalization factor constant, it is straightforward to obtain the density evolution of normalized MS from that of standard MS. The only change is the density function $f_U^{(i)}(u)$ in (12) as

$$f_U^{(i)}(u) \leftarrow \frac{1}{\alpha} f_U^{(i)}\left(\frac{u}{\alpha}\right) \tag{14}$$

and the other procedures are the same as in MS [9].

V. Two-dimensional normalized MS

Another possible way to improve the performance of MS is to reprocess the delivered LLR from bit nodes by modifying (2) into

$$V_{mn}^{(i)} = U_{ch,n} + \beta \cdot \sum_{m' \in \mathcal{M}(n) \backslash m} U_{m'n}^{(i)} \tag{15}$$

where β is a normalization factor. For regular LDPC codes, the normalization factor for each check (bit) node is identical. In this case, bit node normalization processing (15) is equivalent to check node normalization. For irregular LDPC codes, the conventional normalized MS only reprocesses the LLRs outgoing from check nodes while keeping the bit node processor unchanged. For any given check node, the outgoing LLRs are normalized equally without considering the degree differences among the adjacent bit nodes in the graph representation of the code. However, intuitively, for bit nodes with different degrees, this same incoming LLR should play a different role in the outgoing LLRs. This implies it is possible to apply normalization post processing in both check and bit nodes. Because conventionally, only outgoing LLR from check nodes (horizontal step) are normalized, we may refer to it as one-dimensional normalized MS while we refer to this proposed approach as two-dimensional (both horizontal and vertical step) normalized MS decoding.

A. 2-D normalized MS algorithm

Consider an irregular LDPC code with degree distribution $\lambda(x) = \sum_{j=1}^{dv_{max}} \lambda_j x^{j-1}$ and $\rho(x) = \sum_{j=1}^{dc_{max}} \rho_j x^{j-1}$, which specify the degree distribution of bit nodes and check nodes, respectively. The parameters λ_j and ρ_j are the fractions of edges belonging to degree-j bit and check nodes, respectively. The limits dv_{max} and dc_{max} denote the maximum degree for bit and check nodes, respectively. Let α_j be the normalization factor for check nodes with degree j, for $j = 1, 2, \ldots, dc_{max}$. Let β_j be the normalization factor for bit nodes with degree j, for $j = 1, 2, \ldots, dv_{max}$. Let $dv(n)$ be the degree of bit node n, for $n = 1, 2, \ldots, N$. Let $dc(m)$ be the degree of check node m, for $m = 1, 2, \ldots, M$. The initialization and hard decision procedures in 2-dimensional normalized MS are the same as in MS decoding. Step 1 of normalized MS decoding is

Step 1: (i) Horizontal Step, for $1 \leq n \leq N$ and each $m \in \mathcal{M}(n)$, process:

$$U_{mn}^{(i)} = \alpha_{dc(m)} \cdot \prod_{n' \in \mathcal{N}(m) \backslash n} \operatorname{sgn}\left(V_{mn'}^{(i-1)}\right) \cdot \min_{n' \in \mathcal{N}(m) \backslash n} |V_{mn'}^{(i-1)}| \quad (16)$$

(ii) Vertical Step, for $1 \leq n \leq N$ and each $m \in \mathcal{M}(n)$, process:

$$V_{mn}^{(i)} = U_{ch,n} + \beta_{dv(n)} \cdot \sum_{m' \in \mathcal{M}(n) \backslash m} U_{m'n}^{(i)} \quad (17)$$

$$V_{n}^{(i)} = U_{ch,n} + \beta_{dv(n)} \cdot \sum_{m \in \mathcal{M}(n)} U_{mn}^{(i)} \quad (18)$$

It is worth mentioning that by using different normalization factors at different iterations (especially the first few iterations), the error performance of 2-D normalized MS decoding can be improved further.

B. Density evolution of 2-D normalized MS

Given the structure of a code and an iterative message-passing decoding algorithm, if the bit error rate (BER) after each iteration is independent of the transmitted codeword, the analysis of density evolution can be greatly simplified by assuming the all-zero codeword was transmitted. It has been shown in [4] that, if the channel and message-passing decoding satisfy three symmetry conditions, the entire behavior of the decoder can be predicted from its behavior assuming transmission of the all-zero codeword. Next we verify that the 2-D normalized MS decoding satisfies these symmetric conditions and therefore its density evolution can be analyzed based on the all-zero transmitted codeword.

Channel symmetry: For an AWGN channel, the output is symmetric,

$$\begin{aligned} f(y|q) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-q)^2}{2\sigma^2}\right) \\ &= f(-y|-q) \end{aligned}$$

where $\sigma^2 = 1/2R \cdot (E_b/N_o)$ is the variance of the noise and R is the code rate.

Check node symmetry: In check node processing, the sign of an outgoing message can be factored out:

$$\begin{aligned} U_{mn}^{(i)} &= \Psi_c^{(i)}(V_1, V_2, \ldots, V_{dc(m)-1}) \\ &= \alpha_{dc(m)} \cdot \prod_{n' \in \mathcal{N}(m) \backslash n} \operatorname{sgn}\left(V_{mn'}^{(i-1)}\right) \cdot \min_{n' \in \mathcal{N}(m) \backslash n} |V_{mn'}^{(i-1)}| \\ &= \prod_{j=1}^{dc(m)-1} \operatorname{sgn}(V_j) \cdot \Psi_c^{(i)}(|V_1|, |V_2|, \ldots, |V_{dc(m)-1}|) \end{aligned}$$

Bit node symmetry: In bit node processing, the sign of an outgoing message flips if the signs of all the corresponding incoming messages flip,

$$\begin{aligned} -V_{mn}^{(i)} &= -\Psi_v(U_{ch,n}, U_1, U_2, \ldots, U_{dv(n)-1}) \\ &= -U_{n,ch} - \beta_{dv(n)} \cdot \sum_{m' \in \mathcal{M}(n) \backslash m} U_{m'n}^{(i)} \\ &= \Psi_v(-U_{ch,n}, -U_1, -U_2, \ldots, -U_{dv(n)-1}) \end{aligned}$$

Moreover, at iteration 0, the initial belief message

$$\begin{aligned} U_{ch,n} &= \log \frac{P(w_n = 0|y_n)}{P(w_n = 1|y_n)} \\ &= \frac{2}{\sigma^2} y_n \end{aligned}$$

has density function which is also symmetric:

$$\begin{aligned} f_{U_{ch,n}}(u) &= \frac{\sigma}{2\sqrt{2\pi}} \exp\left(-\frac{(u - \frac{2}{\sigma^2})^2}{2(4/\sigma^2)}\right) \\ &= f_{U_{ch,n}}(-u)\exp(u) \end{aligned}$$

Although the original pdf of belief messages in 2-D normalized MS satisfies symmetry conditions, this symmetry property is not preserved during decoding iterations. Therefore there is no guarantee that the probability of error is non-increasing for 2-D normalized MS decoding [12]. Since the pdf of the outgoing messages of check nodes with different degree are distinct, the expectation of these pdf's is the overall pdf. Similarly to (14), the density evolution of degree-j check processor in 2-D normalized MS is based on

$$f_{U,j}^{(i)}(u) \leftarrow \frac{1}{\alpha_j} f_U^{(i)}\left(\frac{u}{\alpha_j}\right) \tag{19}$$

where $f_{U,j}^{(i)}(u)$ is the pdf of an outgoing message at a degree-j check. The overall pdf for all messages going out of check nodes is

$$\begin{aligned} f_U^{(i)}(u) &= \sum_{j=1}^{dc_{max}} \rho_j \cdot f_{U,j}^{(i)}(u) \\ &\leftarrow \sum_{j=1}^{dc_{max}} \frac{\rho_j}{\alpha_j} f_U^{(i)}\left(\frac{u}{\alpha_j}\right) \end{aligned}$$

Similarly the overall pdf for all messages going out of bit nodes is

$$f_V^{(i)}(v) = \sum_{j=1}^{dv_{max}} \frac{\lambda_j}{\beta_j} \mathcal{F}^{-}\left(\mathcal{F}(f_{U_{ch}}) \cdot \left(\mathcal{F}(f_U^{(i)})\right)^{j-1}\right)\left(\frac{v}{\beta_j}\right)$$

VI. Optimal 2-D normalization factors pair

Let $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_{dc_{max}}\}$ and $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_{dv_{max}}\}$ be the normalization factors (vectors) of check and bit nodes in 2-D normalization MS decoding, respectively. Given an irregular LDPC code with degree distribution pair (λ, ρ) and a normalization factor pair $(\boldsymbol{\alpha}$,$\boldsymbol{\beta})$, we can obtain the threshold of this code with 2-D normalized MS decoding, by using density evolution. We try to find the optimal normalization factor pair $(\boldsymbol{\alpha}$,$\boldsymbol{\beta})$ which yields the largest noise threshold. This is a nonlinear cost function minimization problem with continuous space parameters. The brute force search method becomes intractable when $dv_{max} \cdot dc_{max}$ is large. However an algorithm called parallel differential has been shown to be efficient and robust. In [5], this technique has been successfully applied to construct optimal irregular LDPC codes. In this paper parallel differential optimization is used to generate the optimal normalization factor pair of 2-D normalized MS decoding.

A. Parallel differential optimization

Parallel differential is a parallel direct search technique featured by evolution of variable vectors. Starting from an initial set of randomly generated vectors, the algorithm iteratively update each vector in the set simultaneously based on the currently best vector, which yields the smallest cost function value in the last iteration. The updating procedure is repeated until the iterations converge, and the resulting solution is the superior vector which has the best cost function value. Because after each iteration of updating, the new generation in general has a smaller average cost function value than that of the last generation, this algorithm is also called differential evolution. By updating all the vectors of the set in parallel, the algorithm can help the vectors escape local minima and prevent misconvergence.

B. Optimal normalization factor pair based on density evolution and parallel differential optimization

Given an irregular (dv_{max}, dc_{max}) LDPC code, the number of degrees of freedom with normalization factor pair is $dv_{max} \cdot dc_{max}$. In differential evolution optimization, the population of each generation is often selected to be multiple of the number of degrees of freedom. Let $L = T \cdot dv_{max} \cdot dc_{max}$ be the population in each generation, where T is a constant. Apparently, the larger T is, the higher the probability that a better optimal result could be found, and correspondingly the higher the needed complexity. Let $(\boldsymbol{\alpha}_l^{(g)}, \boldsymbol{\beta}_l^{(g)})$ denote the l-th member of generation g. The procedure to find the optimal factor pair of 2-D normalized MS based on density evolution and parallel differential optimization is carried out as follows

Step 1 Initialization: Set the maximum number of generations to G_{max} and the generation index $g = 0$. Randomly generate a set of normalization factor pairs $\{(\boldsymbol{\alpha}_l^{(0)}, \boldsymbol{\beta}_l^{(0)})\}$ with cardinality L. For $l = 1, 2, \ldots, L$, run density evolution of 2-D normalized MS based on the factor pair $(\boldsymbol{\alpha}_l^{(0)}, \boldsymbol{\beta}_l^{(0)})$ and obtain the corresponding threshold $(\frac{E_b}{N_o})_l^{(0)}$. Then compare and find the factor pair which yields the smallest $\frac{E_b}{N_o}$. We denote this best pair among the original generation as $(\boldsymbol{\alpha}_{l_{best}}^{(0)}, \boldsymbol{\beta}_{l_{best}}^{(0)})$, where $l_{best} = \arg\min_{l=1}^{L} \left(\frac{E_b}{N_o}\right)_l^{(0)}$.

Step 2 Mutation and test: Set $g \leftarrow g+1$. Update each factor pair in the candidate set according to the following mutation algorithm. For $l = 1, 2, \ldots, L$, generate J distinct random numbers $\{r_j | 1 \le r_j \le L, r_j \ne l\}$, and generate the test vector

$$\begin{aligned} (\boldsymbol{\alpha}_l^{(g+1)}, \boldsymbol{\beta}_l^{(g+1)})_t &= (\boldsymbol{\alpha}_{l_{best}}^{(g)}, \boldsymbol{\beta}_{l_{best}}^{(g)}) \\ &+\gamma \cdot \left(\sum_{\substack{1 \le j \le J \\ j:odd}} (\boldsymbol{\alpha}_{r_j}^{(g)}, \boldsymbol{\beta}_{r_j}^{(g)}) \right. \\ &\left. - \sum_{\substack{1 \le j \le J \\ j:even}} (\boldsymbol{\alpha}_{r_j}^{(g)}, \boldsymbol{\beta}_{r_j}^{(g)}) \right) \end{aligned} \tag{20}$$

where γ is a pre selected real constant which controls the amplification of the differential variation. The use of differences between candidate vectors increases the variation, therefore helping the iteration escaping from a local minimum. Then for $l = 1, 2, \ldots, L$, run density evolution of 2-D normalized MS based on the newly generated test factor pair $(\boldsymbol{\alpha}_l^{(g+1)}, \boldsymbol{\beta}_l^{(g+1)})_t$ and obtain the corresponding threshold $(\frac{E_b}{N_o})_{l,t}^{(g+1)}$.

Step 3 Compare and update: For $l = 1, 2, \ldots, L$, set

$$\begin{aligned} &(\boldsymbol{\alpha}_l^{(g+1)}, \boldsymbol{\beta}_l^{(g+1)}) \\ &= \begin{cases} (\boldsymbol{\alpha}_l^{(g+1)}, \boldsymbol{\beta}_l^{(g+1)})_t & \text{if } (\frac{E_b}{N_o})_{l,t}^{(g+1)} \le (\frac{E_b}{N_o})_l^{(g)} \\ (\boldsymbol{\alpha}_l^{(g)}, \boldsymbol{\beta}_l^{(g)}) & \text{otherwise} \end{cases} \end{aligned}$$

and

$$\left(\frac{E_b}{N_o}\right)_l^{(g+1)} = \begin{cases} \left(\frac{E_b}{N_o}\right)_{l,t}^{(g+1)} & \text{if } \left(\frac{E_b}{N_o}\right)_{l,t}^{(g+1)} \leq \left(\frac{E_b}{N_o}\right)_l^{(g)} \\ \left(\frac{E_b}{N_o}\right)_l^{(g)} & \text{otherwise} \end{cases}$$

Then update the best factor pair of generation $g+1$ to $(\boldsymbol{\alpha}_{l_{best}}^{(g+1)}, \boldsymbol{\beta}_{l_{best}}^{(g+1)})$, where $l_{best} = \arg\min_{l=1}^{L} \left(\frac{E_b}{N_o}\right)_l^{(g+1)}$.

Step 4 Stoping test and output: If G_{max} is reached or the smallest threshold of generations stop decreasing, output $(\boldsymbol{\alpha}_{l_{best}}^{(g+1)}, \boldsymbol{\beta}_{l_{best}}^{(g+1)})$. Otherwise, go to Step 2.

VII. Simulation results

Figure 1 depicts the word error rate (WER) performance of the standard BP, 2-D normalized MS, conventional normalized MS and MS algorithms for decoding a (16200,7200) irregular LDPC code. The check and bit node distributions of this code are $\rho(x) = 0.00006x^2 + 0.14917x^3 + 0.29851x^4 + 0.44777x^5 + 0.10449x^6$ and $\lambda(x) = 0.00002 + 0.38803x + 0.31344x^2 + 0.29851x^7$, respectively. We observe that 2-D normalized MS provides a comparable performance as BP and interestingly has a lower error floor than that of BP in the high SNR region. We also observe that 2-D normalized MS outperforms conventional normalized MS and MS by about 0.3dB. For this $(16200, 7200)$ irregular LDPC code, the thresholds computed by density evolution for standard BP, MS, conventional MS and 2-D normalized MS are 0.77dB, 0.93dB, 0.90dB and 0.85dB, respectively. The optimum normalization factor for 1-D normalized MS is $\alpha = 0.75$ and the optimum normalization vectors of 2-D normalized MS are $\boldsymbol{\alpha} = (1.00, 0.94, 0.92, 0.88, 0.86)$ and $\boldsymbol{\beta} = (1.00, 1.00, 0.91, 0.83)$.

VIII. Conclusion

In this paper a 2-D normalized MS decoding has been presented to improve the performance of standard MS and normalized MS decoding. The proposed method requires considerably less complexity than that of BP while introducing small performance degradation compared with BP. It is interesting to note that in the high SNR region, 2-D normalized MS can have a lower error floor than that of BP. With respect to conventional normalized MS which can be viewed as a simplified version of 2-D normalized MS decoding, the presented method offers a better performance with negligible increased complexity. Simulation and density evolution show that 2-D normalized MS provides a good performance versus complexity tradeoff for decoding irregular LDPC codes. The method discussed in this paper can be extended to offset MS decoding of irregular LDPC codes [13].

Acknowledgement

The authors would like to express their gratitude to Jinghu Chen for many interesting discussions and constructive comments.

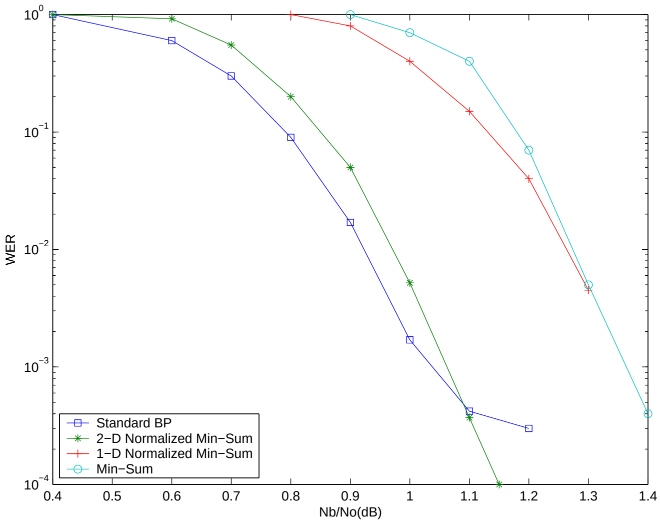

Fig. 1. WER of standard BP, 2-D normalized Min-Sum, conventional normalized Min-Sum and Min-Sum algorithms for decoding a (16200,7200) irregular LDPC code ($\mathrm{I_{max}}$=200).

References

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

[2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

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

[4] 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.

[5] T. J. Richardson, M. A. Shokrollahi, and R.L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inf Theory*, vol.47, pp. 619-637, Feb. 2001.

[6] J. Hagenauer, E. Offer and L. Papke, “Iterative decoding of block and convolutional codes,” *IEEE Trans. on Inform.*, vol. 42, pp. 429-445, March 1996.

[7] M. Fossorier, M. Mihaljevic 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.

[8] J. Chen and M. Fossorier, “Near optimum universal belief propagation based decoding of low-density parity-check codes,” *IEEE Trans. Commun.*, vol. 50, pp. 406-414, Mar. 2002.

[9] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier and X.-Y. Hu, “Reduced-Complexity Decoding of LDPC Codes,” *IEEE Trans. Commun.*, vol. 53, pp. 1288-1299, Aug. 2005.

[10] X. Wei and A. N. Akansu, “Density evolution for low-density parity-check codee under Max-Log-MAP decoding,”*Electron. Lett.*, vol.37, pp. 1225-1226, Aug. 2001.

[11] J. Chen and M. Fossorier, “Density evolution for two improved BP-sased decoding algorithms of LDPC codes,” *IEEE Commun. Lett.*, vol.6, pp. 208-210, May 2002.

[12] S. Y. Chung, “On the construction of some capacity-approaching coding schemes,” Ph.D. dissetation, MIT, Cambridge, MA, 2000.

[13] J. Zhang, M. Fossorier, D. Gu, and J.Y. Zhang, “Two-Dimensional Correction for Min-Sum Decoding of Irregular LDPC Codes,” submitted to *IEEE Commun. Lett.*