J Electr Eng Technol.2017; 12(3): 1262-1270
http://doi.org/10.5370/JEET.2017.12.3.1262

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Simplified 2-Dimensional Scaled Min-Sum Algorithm for LDPC Decoder

Keol Cho*, Wang-Heon Lee and Ki-Seok Chung†**

Abstract – Among various decoding algorithms of low-density parity-check (LDPC) codes, the min-sum (MS) algorithm and its modified algorithms are widely adopted because of their computational simplicity compared to the sum-product (SP) algorithm with slight loss of decoding performance. In the MS algorithm, the magnitude of the output message from a check node (CN) processing unit is decided by either the smallest or the next smallest input message which are denoted as min1 and min2, respectively. It has been shown that multiplying a scaling factor to the output of CN message will improve the decoding performance. Further, Zhong et al. have shown that multiplying different scaling factors (called a 2-dimensional scaling) to min1 and min2 much increases the performance of the LDPC decoder. In this paper, the simplified 2-dimensional scaled (S2DS) MS algorithm is proposed. In the proposed algorithm, we figure out a pair of the most efficient scaling factors which multiplications can be replaced with combinations of addition and shift operations. Furthermore, one scaling operation is approximated by the difference between min1 and min2. The simulation results show that S2DS achieves the error correcting performance which is close to or outperforms the SP algorithm regardless of coding rates, and its computational complexity is the lowest comparing to modified versions of MS algorithms.

Keywords: Error-correction code, Low-density parity-check code, LDPC decoder, Min-sum algorithm, Normalized min-sum algorithm

1. Introduction

Low-density parity-check (LDPC) codes were introduced by Gallager [1], but hardly attracted attention due to its high hardware implementation complexity. However, Mackay and Neal rediscovered advantages of LDPC codes in 1996 [2], and various researches, such as improving error-correcting performance, implementing efficient LDPC decoders, and lowering power consumption of LDPC decoders, have been conducted [3-5]. Due to their near Shannon limit performance [6], easily parallelizable characteristics, and linear decoding complexity, LDPC codes have become popular error correcting codes in many modern communication systems which require faster and higher data rates without any error. LDPC codes have been adopted as forward-error correction (FEC) codes in several emerging communication standards, such as IEEE 802.11n/ac (Wi-Fi) [7, 8], IEEE 802.11ad (WiGig) [9], and IEEE 802.3an (10 Gbase-T Ethernet) [10].

An LDPC code is uniquely defined by an M by N parity check matrix **H**, where M is the number of the parity checks and N is the length of the codeword. The matrix **H** for a binary LDPC code is very sparse with few nonzero elements. As shown in Fig. 1, the **H** matrix is also described by a bipartite graph [11], which is composed of variable nodes (VNs) for columns of the **H** matrix in one partite and check nodes (CNs) for rows of the **H** matrix in the other. An edge is connected between VN i and CN j if the element of the i-th column and the j-th row is one. The number of VNs connected to a CN is the degree of CN, d_c, and the number of CNs connected to a VN is the degree of VN, d_v. An LDPC code with constant d_v and d_c is called (d_v, d_c)-regular code; otherwise, it is called an irregular code.

The LDPC codes are typically decoded by a message-passing algorithm, which iteratively exchanges messages through the edges between the CNs and the VNs. In the sum-product (SP) algorithm, also known as the belief-propagation (BP) algorithm, messages are exchanged in the form of log-likelihood ratios (LLRs) between CNs and VNs [1]. The SP algorithm achieves a powerful decoding performance close to the Shannon limit, but suffers from high computational complexity. The computational complexity of the SP algorithm can be greatly reduced by using the min-sum (MS) approximation [12], but slight performance loss is incurred. To resolve the performance loss of the MS algorithm, many modified versions of the MS algorithm have been proposed. Most of them have tried to multiply the check to variable node (CTV) messages by a scaling factor to compensate for over-estimated belief messages in comparison to the SP algorithm, and thus, these approaches are commonly called normalized MS (NMS) algorithms [13, 14]. In [15], the CTV messages are adjusted by an offset based on the number of VNs connected to the CNs, and the CTV

† Corresponding Author: Dept. of Electronic Engineering, Hanyang University, Seoul, Korea. (kchung@hanyang.ac.kr)
* Dept. of Electronics and Computer Engineering, Hanyang University, Seoul, Korea. (keolman2@gmail.com)
** Dept. of IT, Hansei University, Gunpo-si, Gyeonggi-do, Korea. (whlee@hansei.ac.kr)
Received: July 28, 2016; Accepted: December 26, 2016

Copyright ⓒ The Korean Institute of Electrical Engineers
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

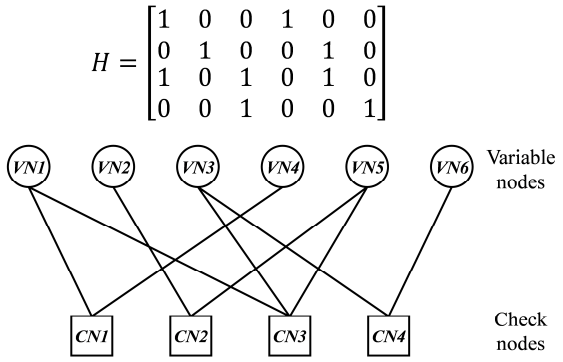


Fig. 1. An example of the H matrix for LDPC code and its bipartite graph representation

messages are adaptively scaled based on the iteration count [16, 17]. In [18], the first two smallest CTV messages are scaled by different scaling factors using density evolution to improve the decoding performance. Even though the existing algorithms enhance the performance of LDPC decoders, [15, 16], and [18] did not take the hardware implementation cost into account, and [17] suffered from the increased average iteration count until the decoding process is completed.

To estimate the hardware cost of the LDPC decoder, the number of bits to quantize the exchanged messages should be considered because it directly affects both the error-correcting capability and the hardware cost of the decoder. The hardware implementation cost typically includes the decoder circuit size and the amount of memory usage. Studies have shown that there is slight performance loss when 5 or 6 bit fixed point representation is used to quantize the message compared to floating point representation [19, 20]. [21] aggressively quantized the CTV message using only 2 bits, but the performance loss was about 0.3 dB. Finding the first two minima among VN to CN (VTC) messages is usually adopted when MS-based LDPC decoders are implemented in hardware due to the efficiency in memory usage [22]. B. Xiang et al. [23] showed that instead of sending the two found minima, the interconnection complexity and memory usage could be reduced by compressing CTV messages using the difference of the two minima (Δmin) which will need a smaller quantization bit width. By this compression, the memory usage is reduced by 5.64% while the performance loss is up to 0.15 dB.

In this paper, a new decoding algorithm which both the decoding performance and the complexity of hardware implementation of LDPC decoder are taken into account based on the MS algorithm is proposed. As reported in [18], applying two dimensional (2D) scaling factors to the first two minima improves the decoding performance. In the proposed algorithm of this paper, the 2D scaling factors are determined so that the hardware cost is minimized without losing the decoding performance. In addition, these scaling factors are further optimized by using the Δmin of the CTV message, and achieves 0.2 to 0.5 dB coding gain with the least computational complexity compared to modified versions of MS algorithms.

The remainder of this paper is organized as follows. In section 2, representative decoding algorithms of the LDPC code are briefly introduced. The proposed decoding algorithm is explained in detail in section 3, and the experimental results are described with the complexity analysis in section 4. Finally, our conclusions are presented in section 5.

2. Decoding Algorithms of the LDPC Code

In this section, representative decoding algorithms of the LDPC code will be described. Let us suppose that the LDPC code is defined by an M by N parity check matrix. The set of VNs neighboring CN j ($j = 1,2,\dots,M$) is denoted as V_j, and the set of CNs neighboring VN i ($i = 1,2,\dots,N$) is denoted as C_i. Also, $V_j \backslash i$ denotes the subset of VNs excluding the i-th VN, and $C_i \backslash j$ represents the subset of CNs excluding the j-th CN. The LDPC decoder iteratively updates the belief messages and estimates codewords using the following information.

- F_i : The initial LLR value (*a priori* LLR) of the i-th bit. It is derived from the received vector, y_i.
- $L_{j\to i}^{(l)}$: The CTV message; the message sent from CN j to VN i at the l-th iteration count. It is obtained from the extrinsic VTC messages $L_{i'\to j}$, where $i' \in \mathrm{V_j} \backslash i$.
- $L_{i\to j}^{(l)}$: The VTC message; the message sent from VN i to CN j at the l-th iteration count. It is obtained from F_i and the extrinsic CTV message $L_{j'\to i}$, where $j' \in \mathrm{C_i} \backslash j$.
- z_i : The *a posteriori* LLR of the i-th bit computed at each iteration. It is obtained from F_i and the information $L_{j\to i}^{(l)}$.

2.1 Sum-product algorithm

The SP algorithm assuming that codewords are modulated by binary phase shift keying (BPSK) and they are transmitted over additive Gaussian noise channel (AWGN) with noise variance, δ^2 can be described as follows.

- *Initialization*: For each i, *a priori* LLR and the initial VTC message are updated by

$$L_{i\to j}^{(0)} = F_i = \frac{2y_i}{\delta^2}. \tag{1}$$

- *Iterative steps*: For each iteration count l (l = *1, ..., max_iteration*) the three following steps are processed.
  1) Check node process: For each j, i, update CTV message by

$$L_{j\to i}^{(l)} = 2\tanh^{-1}\left(\prod_{i'\in V_j\setminus i}\tanh\left(\frac{L_{i'\to j}^{(l-1)}}{2}\right)\right). \quad (2)$$

2) Variable node process: For each j, i, update VTC message by

$$L_{i\to j}^{(l)} = F_i + \sum_{j'\in C_i\setminus j} L_{j'\to i}^{(l)}, \quad (3)$$

and update a posteriori LLR by

$$z_i = F_i + \sum_{j\in C(i)} L_{j\to i}^{(l)} \quad (4)$$

3) Tentative decision and stopping criterion test:

i) In the tentative decision, the estimated codeword $\hat{\boldsymbol{c}} = \{\hat{c}_1, \hat{c}_2, ..., \hat{c}_N\}$ is constructed based on z_i by

$$\hat{\boldsymbol{c}} = \{\hat{c}_1, \hat{c}_2, ..., \hat{c}_N\}, \text{ where } \hat{c}_i = \begin{cases} 0, & z_i \geq 0 \\ 1, & z_i < 0 \end{cases} \quad (5)$$

ii) If either the syndrome check, $\mathbf{H}\cdot\hat{\boldsymbol{c}}^T = 0$, or the number of iterations reaches the predefined maximum count, $\hat{\boldsymbol{c}}$ becomes the output of the decoder. When the syndrome check is not satisfied, the decoder goes back to 2) and increments the iteration count.

2.2 Min-sum algorithms

In the MS algorithm, (3), (4), and (5) are equivalent to the SP algorithm. Instead of (1), the MS algorithm initializes F_i and $L_{j\to i}^{(0)}$ with

$$L_{i\to j}^{(0)} = F_i = y_i \quad (6)$$

and the CN process (3) is approximated by the minimum finding function as follows:

$$L_{j\to i}^{(l)} = \prod_{i'\in V_j\setminus i}\operatorname{sign}\left(L_{i'\to j}^{(l-1)}\right)\min_{i'\in V_j\setminus i}\left|L_{i'\to j}^{(l-1)}\right| \quad (7)$$

The approximation of the CTV message in the MS algorithm is known to be overestimated compared to the CTV message of the SP algorithm, [13] and [14] normalize the CTV message using scaling factor, α. With the normalization, (7) is rewritten by

$$L_{j\to i}^{(l)} = \alpha\prod_{i'\in V_j\setminus i}\operatorname{sign}\left(L_{i'\to j}^{(l-1)}\right)\min_{i'\in V_j\setminus i}\left|L_{i'\to j}^{(l-1)}\right| \quad (8)$$

It is proven that the optimal α varies according to the code rate, the signal-to-noise (SNR) ratio, and the codeword length [24]. Therefore, various researches have followed in order to figure out α for the best error correcting performance or for the most efficient hardware implementation [15-18, 20].

In [15], degree match two-step MS (DM2S) has been proposed. DM2S compensates CTV messages by subtracting positive correction factors, which is derived by logarithmic calculation of d_c. In DM2S, CNs have different correction factors according to their own d_c's, and the steps are decided by the magnitude of the smallest LLR and the distance between the first two smallest magnitudes of LLRs.

[16] and [17] adaptively adjust α according to the iteration count of the decoding process, and the adjustment is based on the fact that the reliability of the LLRs is improved as the iteration count goes up. Generalized simplified variable-scaled MS (GSVS) algorithm [17] divides the iteration count into four steps and the initial scaling factor α_0 is increased as follows: α_0, $0.5 + 0.5\cdot\alpha_0$, $0.75 + 0.25\cdot\alpha_0$, and $0.875 + 0.125\cdot\alpha_0$.

In (7), the minimum should be found for all of the neighboring VNs excluding the i-th VN, which implies that the magnitude of CTV message is either the smallest LLR or the next smallest LLR. Thus, the minimum finding part in (7) can be replaced by

$$\min_{i'\in V_j\setminus i}\left|L_{i'\to j}^{(l)}\right| = \begin{cases} \min 2, & \text{if the index of } \min 1 = i \\ \min 1, & \text{otherwise} \end{cases} \quad (9)$$

where min1 and min2 are the first and the second minimum, respectively. Zhong et al. [18] have shown that 2-dimensional scaling (2DS) of min1 and min2 by α_1 and α_2 ($0 < \alpha_1 < \alpha_2 < 1.0$), respectively, achieves almost 0.4 dB coding gain compared to a single scaling factor. However, the scaling factors of 2DS and DM2S do not consider the hardware implementation cost of the LDPC decoder.

In order to reduce the implementation complexity, a modified 2DS MS algorithm, called simplified 2-dimensional scaled (S2DS) MS algorithm, is proposed in this paper. First of all, scaling factors for 2-dimensional MS which is called hardware considered 2D scaling factors (H-2DS) are chosen to achieve less complex decoder implementation. Further, the computational complexity of the scaling operation is reduced in S2DS using the Δmin approximation, which has been proposed in [23].

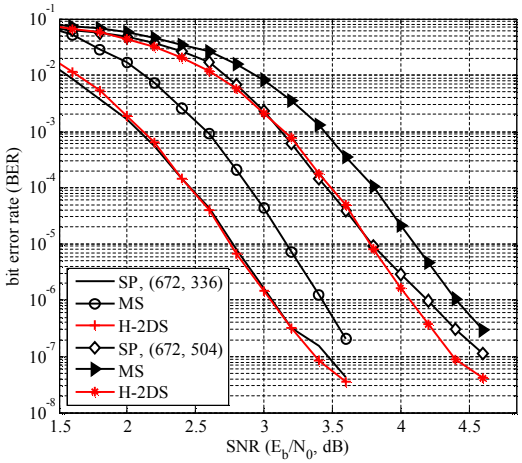
3. Simplified 2-Dimensional Scaled Min-sum Algorithm

3.1 Low complexity 2D scaling factors

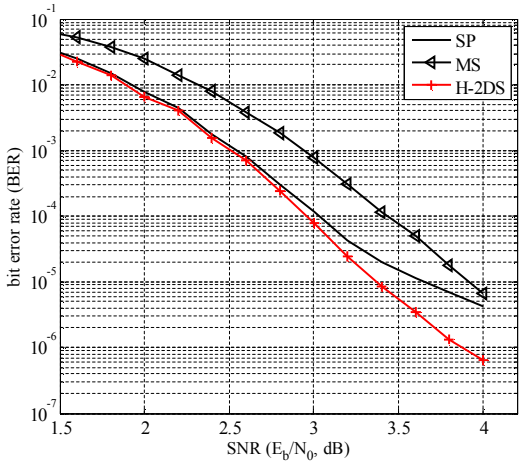
The optimal scaling factors of 2DS vary according to the SNR [18]. Without considering the cost of the hardware implementation of LDPC decoder, 2DS requires multiplying min1 and min2 by $\alpha_1 = 0.4902$ and $\alpha_2 = 0.9174$, respectively. It is obvious that multiplying the two minima

by these scaling factors is quite complicated in aspect of hardware implementation. To figure out the best H-2DS which will require less complex computational complexity, we have carried out simulations with varying α_1 and α_2 from 0.5 to 1.0 with increments of 0.125. It should be noted that multiplication the two minima by these scaling factors can be implemented with the combination of add and shift operations. For the simulation, the subsets of LDPC codes defined in IEEE 802.11ad applications [9], which are irregular and have block length N and dimension K, (N, K) = (672, 336) and (672, 504), is chosen. For a regular LDPC code, (408, 204) code with (d_v, d_c) = (3, 6) is chosen. The maximum iteration count is set to 20, and *a priori* LLRs and the exchanging LLRs have been simulated using the floating point arithmetic. The simulation with the BPSK transmission of the all-zero codeword over the AWGN channel was carried out until the number of frame errors reached at least 100.

Through the simulations, it turns out that the best performance is achieved with $\alpha_1 = 0.75$ and $\alpha_2 = 0.875$, and



(a) (672, 336) and (672, 504) irregular codes



(b) (408, 204) regular code (d_v, d_c) = (3, 6)

Fig. 2. BER performance of hardware-considered 2DS

the corresponding results of the bit-error rate (BER) performance comparison with the SP and MS algorithms are depicted in Fig. 2. The solid lines and the dashed lines in Fig. 2(a) represent the BER simulation results of the (672, 336) code and the (672, 504) code, respectively, and Fig. 2(b) shows the result of the (408, 204) regular LDPC code. Regardless of the degree regularity and the code rate of the LDPC code, H-2DS achieves coding gains from 0.4 dB to 0.6 dB compared to the MS algorithm at BER of 10^{-5}, and H-2DS even outperforms the SP algorithm in high SNR regions. As aforementioned, multiplications with 0.75 and 0.875 can be implemented by a combination of add and shift operations. The scaling x with 0.75 can be implemented as $0.75 \cdot x = x/2 + x/4$ and the scaling x with 0.875 can be implemented as $0.875 \cdot x = x/2 + x/4 + x/8$. According to (8) and (9), the magnitude of the CTV message of the normalized MS algorithm is calculated as follows

$$\left|L_{j\to i}^{(l)}\right| = \begin{cases} \alpha_2 \cdot \min 2, & \text{if the index of } \min 1 = i \\ \alpha_1 \cdot \min 1, & \text{otherwise} \end{cases} \quad (10)$$

and the CTV message consists of {signs, index of min1, $\alpha_1 \cdot$min1, $\alpha_2 \cdot$min2}. When a single scaling factor is used, α_1 of (10) is equal to α_2 ($\alpha_1 = \alpha_2 = \alpha$), which means that it is still required two multiplications (or few shifts and addition) for scaling in the CN unit (min1 and min2 with α). On the other hand, H-2DS requires only shift and add operations, and due to the fact that the scaling factor 0.875 can be computed with only one more add and shift operations in addition to computing the scaling factor 0.75, only one arithmetic circuit can be used for both scaling operations in a serial manner to save the hardware implementation cost.

3.2 Simplified 2D scaling with Δmin

Based on the pair of scaling factors of H-2DS, the proposed S2DS decoding algorithm further reduces the computational complexity using Δmin information [23]. In [23], the CTV message is compressed using the difference between min1 and min2, Δmin (=min2 – min1). Thus, a CTV message consists of {signs, index of min1, min1, Δmin}, which reduces the memory usage and the inter-connection complexity of the LDPC decoder compared to sending min1 and min2.

Utilizing Δmin information, S2DS replaces the scaling operation, 0.875·min2 by the following computation under the assumption that the magnitudes of min1 and Δmin are similar:

$$\begin{aligned} 0.875 \cdot \min 2 &= 0.875 \cdot (\min 1 + \Delta\min) \\ &= 0.875 \cdot \min 1 + 0.875 \cdot \Delta\min \\ &= 0.75 \cdot \min 1 + 0.125 \cdot \min 1 + 0.875 \cdot \Delta\min \\ &\approx 0.75 \cdot \min 1 + \Delta\min \ (\textit{iff}.\ \min 1 \approx \Delta\min) \end{aligned} \quad (11)$$

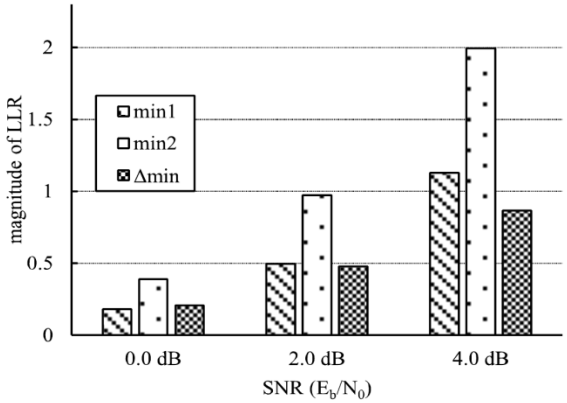


Fig. 3. The average magnitudes of min1, min2, and Δmin

Under this approximation, the CTV message of S2DS consists of {signs, index of min1, 0.75·min1, Δmin}, and S2DS requires only one scaling operation in the CN unit instead of two.

To prove that the above assumption is valid, we have carried out an analysis that compares the magnitudes of min1, min2, and Δmin. The (672, 546) code in IEEE 802.11ad [9] over AWGN channel was chosen for the analysis, and the average magnitudes of min1, min2, and Δmin of 500,000 codewords were calculated varying SNRs. The analysis results are summarized in Fig. 3, which shows that the magnitude of min1 is similar to that of Δmin.

3.3 Fixed-point implementation

To reduce the complexity of a hardware implementation, many digital circuits are designed to handle only the fixed point numbers. However, such fixed point implementations should be concerned about the amount of the quantization error that will result in a performance degradation. In this paper, the BER performance of the S2DS algorithm is estimated with a fixed-point Q*m.f*, where an $(m + f)$-bit fixed-point LLR message consists of m integer bits and f fractional bits. The exchanged LLR messages are quantized with 5 bits excluding the sign bit because 5 bit quantization gives the best tradeoff between the performance and the hardware cost [24]. By varying m and f, we have carried out simulations to figure out which Q*m.f* within 5 bits (Q1.4, Q2.3, Q3.2, and Q4.1) shows the best performance.

The simulations have been conducted under the same conditions in section 3.1. The (672, 336) code and the (672, 504) code are modulated in BPSK and are transmitted over the AWGN channel with 20 as the maximum iteration count. The BER performance comparison results of the S2DS algorithm with 5 bit quantization are shown in Fig. 4, where the solid lines are for the (672, 336) code and the dashed lines are for the (672, 504) code. For both codes, Q1.4 and Q4.1 suffer from significant performance degradation, so the results are not depicted in Fig. 4, whereas Q2.3 shows the best decoding performance and does not suffer from an error floor at high SNR regions

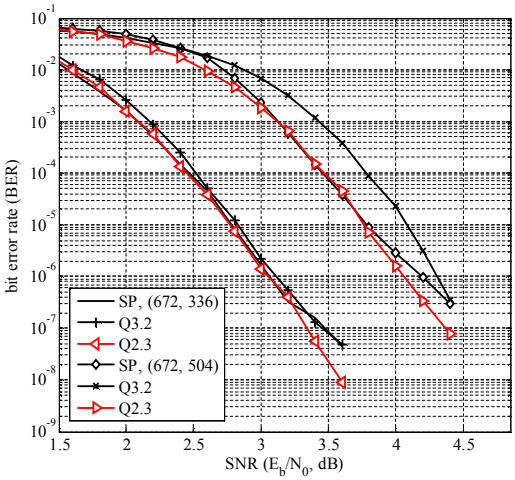


Fig. 4. BER performance of S2DS algorithm in Q3.2 and Q2.3 with SP algorithm in floating point arithmetic

compared to SP and Q3.2.

4. Simulation Results

The simulation results are presented to show the performance of the S2DS algorithm by comparing with the simulation results are presented to show the performance of the S2DS algorithm by comparing with the other MS-based algorithms that were briefly mentioned in the paper: SP, MS, GSVS, and DM2S algorithms. The complexity of the proposed algorithm is also discussed in this section.

All of four **H** matrices defined in IEEE 802.11ad applications [9] are used for the simulation. These four LDPC codes are irregular and consist of rates 1/2, 3/4, 5/8, and 13/16 with a common length of 672 bits: (672, 336), (672, 504), (672, 420), and (672, 546) codes, respectively. BPSK transmission of the all-zero codeword over the AWGN channel was used. Simulations were running until at least 400 frame errors were counted at low and middle SNR simulation points and 100 frame errors for high SNR points. The maximum allowable iteration count was set to 20.

The SP and DM2S algorithms were simulated using a floating-point arithmetic, and the MS and GSVS were simulated using a Q4.6 fixed-point arithmetic. The initial scaling factor, α_0, of GSVS was set to 0.5. For each iteration step, the following scaling factors were used: 0.5, 0.75, 0.875, and 0.9375.

4.1 Decoding performance

The simulation results are compared with the S2DS with Q2.3. As shown in Fig. 5, S2DS algorithm shows the best performance among all the MS-based decoding algorithms. Fig. 5(a), (c), and (d) shows that S2DS achieves the performance close to the SP performance and does not

Table 1. Computational complexity of a check node within a single iteration

Decoding algorithms	Multiplication	Division	Comparison	Addition	Subtraction	Bit shift	Remarks
SP	d_c	d_c - 1	0	0	0	0	tanh, $\tanh^{-1}$
MS	0	0	$d_c + \log_2 d_c$ - 2	0	0	0	-
NMS (0.75)	0	0	$d_c + \log_2 d_c$ - 2	2	0	4	-
DM2S	0	0	$d_c + \log_2 d_c$ + 2	2	3	4	logarithm
GSVS	0	0	$d_c + \log_2 d_c$ - 1	7	0	9	iteration counter
S2DS	0	0	$d_c + \log_2 d_c$ - 2	1	1	2	-

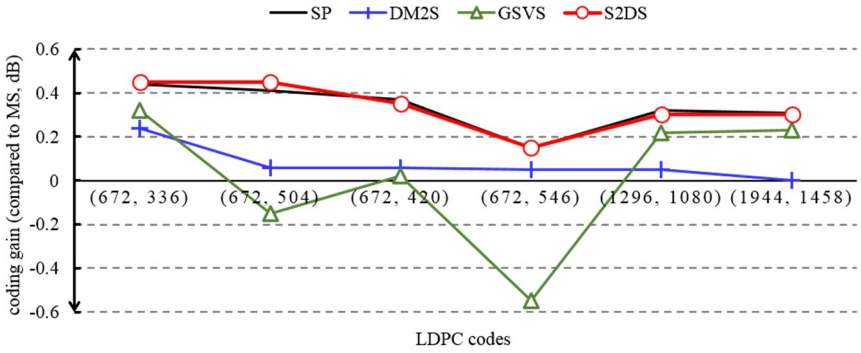


Fig. 6. Coding gain compared to MS at a BER of 10^{-5} with various LDPC codes: N=672 codes in [9], N=1296 and N=1944 in [7]

suffer from an error floor at high SNR points. In the case of high coding rates as shown in Fig. 5 (b), S2DS outperforms SP. It is proven that the SP algorithm does not provide the optimal decoding method in short code lengths [13].

For more precise comparison of the decoding performance, the coding gains compared to MS at a BER of 10^{-5} have been analyzed with two more LDPC codes in [7] which have different code lengths (N=1296 and N=1944) and dimensions with those of in Fig.5. As shown in Fig. 6 which depicts the coding gains for various LDPC codes, S2DS achieves overall coding gain close to SP (less than 0.02 dB) and more than DM2S (from 0.07 dB to 0.3 dB). However, the coding gains of GSVS fluctuate with codes. It implies that it is crucial to figure out the best performing scaling factor sets for GSVS, which requires more computational logics for sets of scaling factors. In contrast, S2DS uses a constant scaling factor and achieves much better performance.

More aggressive quantization of Δmin information was also examined. However, Q1.2 and Q2.1 for Δmin showed significant performance degradation. Q2.2 for Δmin showed a performance drop about 0.05 dB, but suffered from early error floors at high SNR points.

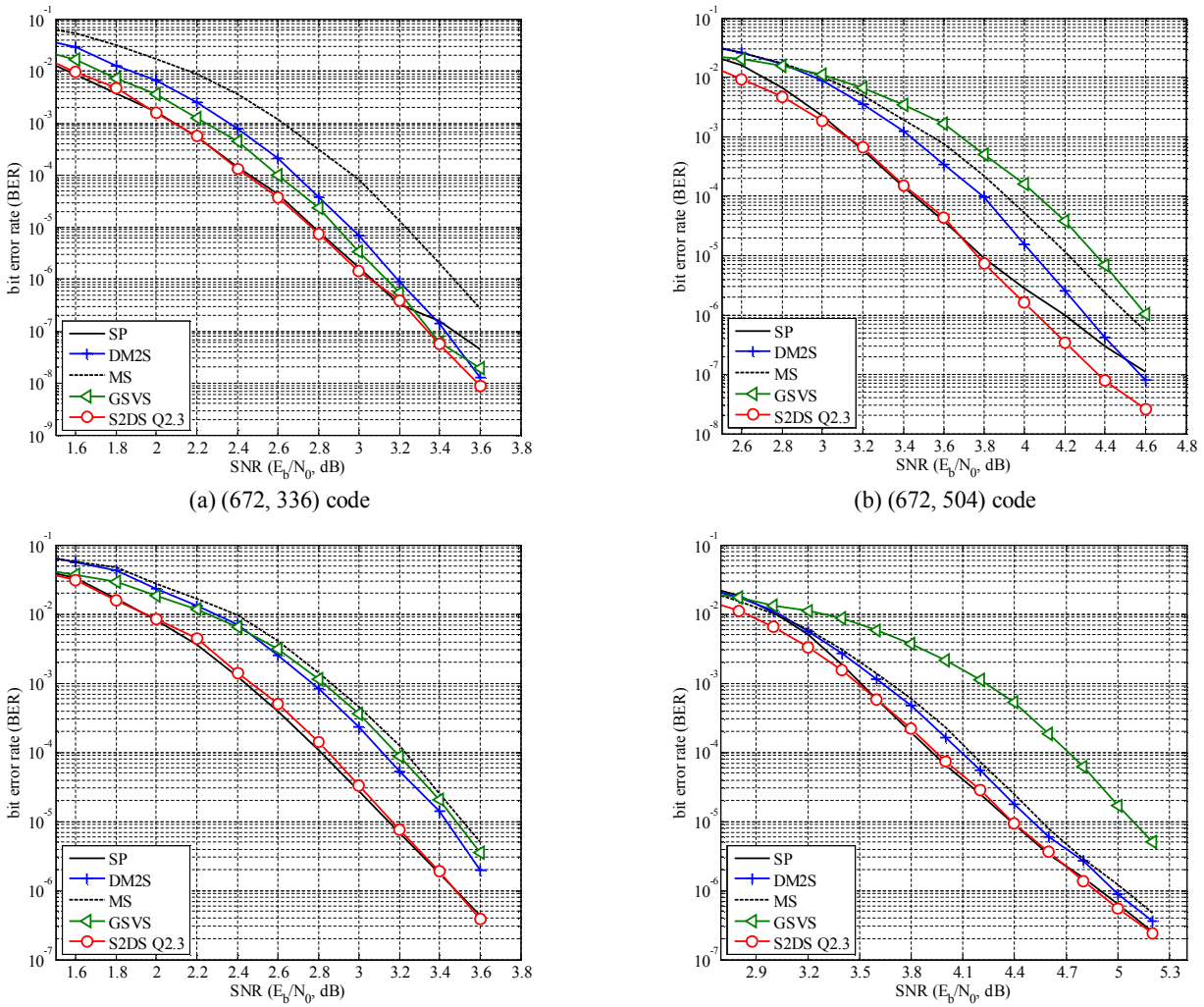
4.2 Computational complexity

In this section, the computational complexity of various decoding algorithms is analyzed. Table 1 summarizes the computational complexity of the CN processing in each iteration. The set of the basic arithmetic operations is listed with extra operations besides the basic operations in the remarks. For example, the SP algorithm requires floating point multiplications and divisions including the hyperbolic tangent calculation. The comparison column shows the required number of comparisons to find the first two minimum values ($d_c + \log_2 d_c$ - 2) for MS algorithm [22].

The MS algorithm requires the least amount of computation among all of the decoding algorithms, and S2DS requires the second least. DM2S requires more computations including natural logarithms for the thresholds and their subjects. GSVS requires quite a few combinations of shift and addition operations to compute the scaling factor sets, and an additional counter and a comparator are required for choosing the step that decides which scaling factors will be used. However, S2DS requires only a single scaling factor that is computed by two shift operations and one addition, and a Δmin information obtained from one subtraction. Furthermore, since the scaling operation of S2DS remains the same for different coding rates, only one scaling unit is required when a multi-rate LDPC decoder is implemented. Considering that SP requires multiplications and divisions and DM2S requires divisions and logarithmic computations, S2DS can be claimed to the best when both the computational complexity and the decoding performance are taken into account.

5. Conclusion

This paper proposed a simple yet powerful 2-dimensional scaled min-sum algorithm called S2DS min-sum algorithm. We figure out scaling factors with which



(a) (672, 336) code

(b) (672, 504) code

(c) (672, 420) code

(d) (672, 546) code

Fig. 5. Decoding performance of Q2.3 S2DS compared with SP, DM2S, MS, and GSVS

scaling operation can be simplified. Further, we show that one scaling operation is approximated by the difference between min1 and min2. Therefore, our proposed algorithm reduced the complexity of the check node computation significantly. In spite of the simplicity, the proposed S2DS algorithm achieves coding gains from 0.2 dB to 0.4 dB compared to the other min-sum based decoding algorithms, and its performance is consistently good regardless of the coding rate or the irregularity of the LDPC codes.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A09061079).

References

[1] Robert G. Gallager, "Low-Density Parity-Check Codes," *IRE Trans. Information Theory,* vol. 8, no. 1, pp. 21-28, Jan. 1962.

[2] D. J. C. MacKay and R. M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes," *IET Electron. Lett.*, vol. 32, no. 18, pp. 1645, Aug. 1996.

[3] L. Li, D. Qu, and T. Jiang, "Partition Optimization in LDPC-Coded OFDM Systems with PTS PAPR Reduction," *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, pp. 4108-4113, Oct. 2014.

[4] A. J. Wong, S. Hemati, and W. J. Gross, "Efficient Implementation of Structured Long Block-Length LDPC Codes," *in 2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors* (*ASAP*), Toronto, Canada, July 2015.

[5] Y. S. Park, D. Blaauw, D. Sylvester, and Z. Zhang, "Low-Power High-Throughput LDPC Decoder Using Non-Refresh Embedded DRAM," *IEEE J. Solid-State Circuits*, vol. 49, no. 3, pp. 783-794, Mar. 2014.
[6] Sae-Young Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58-60, Feb. 2001.
[7] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: enhancements for higher throughput, *IEEE Std. P802.11n/D7.0*, 2008.
[8] IEEE 802.11ac-Enhancements for Very High Throughput for operation in bands below 6 GHz, *IEEE P802.11ac/D5.0*, 2013.
[9] IEEE P802.11ad: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Enhancements for Higher Throughput in the 60 GHz Band, IEEE, 2012.
[10] Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Std. 802.3an, 2006.
[11] R. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Trans. Inform. Theory,* vol. 27, no. 5, pp. 533-547, Sep. 1981.
[12] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996.
[13] J. Chen and M. Fossorier, "Near Optimum Universal Belief Propagation Based Decoding of Low Density Parity Check Codes," *IEEE Trans. Commun.,* vol. COM-50, no. 3, pp. 406-414, Mar. 2002.
[14] J. Heo, "Analysis of Scaling Soft Information on Low Density Parity Check Code," *IET Electron. Lett.*, vol. 39, no. 2, pp. 219, Jan. 2003.
[15] S. L. Howard, C. Schlegel, and V. C. Gaudet, "Degree-Matched Check Node Decoding for Regular and Irregular LDPCs," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 53, no. 10, pp. 1054-1058, Oct. 2006.
[16] Y. Xu, L. Szczecinski, B. Rong, F. Labeau, D. He, Y. Wu, and W. Zhang, "Variable LLR Scaling in Min-Sum Decoding for Irregular LDPC Codes," *IEEE Trans. Broadcast.*, vol. 60, no. 4, pp. 606-613, Dec. 2014.
[17] Ahmed A. Emran and Maha Elsabrouty "Generalized Simplified Variable-Scaled Min Sum LDPC Decoder for Irregular LDPC Codes," *Personal, Indoor, and Mobile Radio Communication* (*PIMRC*), *2014 IEEE 25th Annual International Symposium on*, Washington DC, USA, Sep. 2014.
[18] Zhou Zhong, Shuming Guo, Xiangyang Xu, and Huiqing Bai, "A Classified Normalized BP-Based Algorithm with 2-Dimensional Correction for LDPC Codes," *Journal of Communications*, vol. 8, no. 5, May 2013
[19] R. Zarubica, R. Hinton, S. G. Wilson, and E. K. Hall, "Efficient Quantization Schemes for LDPC Decoders," *in IEEE Military Communications Conference*, San Diego, Nov. 2008.
[20] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, "On Implementation of Min-Sum Algorithm and Its Modifications for Decoding Low-Density Parity-Check (LDPC) Codes," *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 549-554, Apr. 2005.
[21] V. A. Chandrasetty and S. M. Aziz, "FPGA Implementation of High Performance LDPC Decoder Using Modified 2-Bit Min-Sum Algorithm," *in 2010 Second International Conference on Computer Research and Development*, Kuala Lumpur, May 2010.
[22] C. L. Wey, M. D. Shieh and S. Y. Lin, "Algorithms of Finding the First Two Minimum Values and Their Hardware Implementation," *IEEE Trans. Circuits Syst. I,* vol. 55, no. 11, pp. 3430-3437, Dec. 2008.
[23] B. Xiang, R. Shen, A. Pan, D. Bao and X. Zeng, "An Area-Efficient and Low-Power Multirate Decoder for Quasi-Cyclic Low-Density Parity-Check Codes," *IEEE Trans. Very Large Scale Integration Systems,* vol. 18, no. 10, pp. 1447-1460, Sep. 2010.
[24] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and Xiao-Yu Hu, "Reduced-Complexity Decoding of LDPC Codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288-1299, Aug. 2005.

Keol Cho He received his B.S. degree in Media Communication Engineering from Hanyang University, Seoul, Korea in 2009 and his Ph.D. degree in Electronics and Computer Engineering from Hanyang University, Seoul, Korea in 2017. His research interests include system-on-chip architecture, error-correction codes, and hardware implementation of error-correction codes.

Wang-Heon Lee He received his B.S. degree in Control and Instrumentation from Seoul National University in 1985, his M.S. and Ph.D. degrees in Automation and Design Engineering from Korea Advanced Institute of Science and Technology in 1992, 2001,respectively. In 2006, he joined the Department of Information Technology at the Hansei University, now an associate professor and became a chairman of technical committee of

Machine Vision from 2011 of ICROS. His research interests include Deep Learning, Mobile Robot Control and Control system design of Drone.

Ki-Seok Chung He received his B.E. degree in Computer Engineering from Seoul National University, Seoul, Korea, in 1989 and his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 1998. Currently, he is a Professor at Hanyang University, Seoul, Korea. His research interests include low-power embedded system design, multi-core architecture, image processing, reconfigurable processor and DSP design, SoC-platform-based verification, and system software for MPSoC.