A Second Minimum Approximation Method for Layered Min-Sum Decoding of QC-LDPC Codes

Zijian Qin, Yangcan Zhou, and Zhongfeng Wang
School of Electronic Science and Engineering, Nanjing University, Jiangsu, P.R. China
{qingjian, yczhou}@smail.nju.edu.cn, zfwang@nju.edu.cn

***Abstract*—Low-density parity-check (LDPC) codes have attracted tremendous attention for their excellent error-correction performance and inherent fitness for high-parallelism implementation. With great complexity reduction and friendly for hardware implementation, the min-sum (MS) decoding algorithm for LDPC codes gains widespread applications in practical. In the MS decoding algorithm, finding the first two minima among input variable-to-check messages is the most complex part. To further reduce the complexity of the MS decoding algorithm, an efficient single-minimum (single-min) based layered MS decoding for quasi-cyclic LDPC (QC-LDPC) codes is proposed in this paper. It is found that statistically the difference between the first minimum and second minimum increases as the decoding becomes more convergent. Accordingly, in the proposed single-min scheme, the second minimum is approximately derived based on the first minimum and an index indicating the convergence degree. Besides, some other metrics that are easily obtained are used to refine the second minimum. Simulation results show that the proposed method can significantly improve the error-correction performance and lower the average number of iterations, compared with the state-of-the-art single-min based MS decoding algorithms.**

***Index Terms*—Low-density parity-check (LDPC) codes, min-sum (MS) decoding, layered schedule, single-minimum, approximate second minimum.**

I. Introduction

Low-density parity-check (LDPC) codes were invented by Gallager in 1962 [1]. Limited by the computing power at that time, LDPC codes had been silent for decades until its rediscovery by MacKay in 1999 [2]. From then on, LDPC codes have gained much attention from both industry and academia. With very good error-correction performance and the capability of offering high throughput, LDPC codes have been widely adopted by many communication standards, such as IEEE 802.11ad (Wigig) [3] and IEEE 802.11ax (WiFi 6) [4].

The belief propagation (BP) decoding algorithm [5], also known as sum-product (SP) decoding algorithm, shows the most excellent error-correction performance among the decoding algorithms of LDPC codes. However, the excessive computational complexity resulted from transcendental functions in BP decoding makes it undesirable for efficient hardware implementation. The min-sum (MS) decoding algorithm is an approximate version of BP decoding algorithm [6]. Since the magnitudes of check-to-variable (C2V) messages in MS decoding are overestimated compared with that in the origin BP decoding, the MS decoding may degrade the error-correction performance of LDPC codes to a certain degree. To alleviate the overestimation, the normalized MS (NMS) and the offset MS (OMS) decoding algorithms were proposed [7], which significantly narrows down the performance gap between the MS and BP decoding algorithms.

The schedules have a significant impact on the decoding of LDPC codes. The flooding schedule, also known as parallel schedule, was proposed originally by Gallager in his seminal work [1], where all variable-to-check (V2C) messages are simultaneously updated after the simultaneous update of C2V messages in one iteration. From the perspective of hardware implementation, the flooding schedule is unpopular due to its very complex routing. The layered schedule runs in a serial manner and has a much faster convergence speed than the flooding schedule [8], since the latest updated messages from the former layer can be utilized immediately to decode the following layer. Besides, the layered schedule is hardware-friendly, especially for the quasi-cyclic LDPC (QC-LDPC) codes.

In the MS decoding, the most complex part is to find the first and the second minima of input V2C message magnitudes. The minima finder usually forms the bottleneck to further improve the speed of MS decoders. Apparently, only finding the first minimum can drastically reduce the complexity. Hence, single-minimum (single-min) based schemes are promising solutions and various methods were proposed to derive an approximate second minimum with minor complexity. The single-min MS (smMS) [9] obtains the second minimum by adding a constant to the first minimum. It was shown in [10] that the difference between the first and the second minima gets larger as the iteration progresses. Thus, the simplified-variable-weight MS (svwMS) [10] adds a dynamic value that changes along with the number of iterations to the first minimum, compensating the error performance loss. The second minimum approximation MS (SAMS) [11] was proposed to determine the second minimum by adding a dynamic value related to the quantization word length to the first minimum. The dynamic smMS (dsmMS) [12] considered the convergence degree of the decoding to adjust the value adding to the first minimum. The number of non-match bits between a-posteriori probability messages and extrinsic messages was used as an

Zijian Qin and Yangcan Zhou contribute equally to this paper.

This work was supported in part by the National Key R&D Program of China under Grant 2022YFB4400604.

index indicating the convergence degree in [12]. The exMin-n and rExMin-n algorithm [13] divided input V2C messages into several groups. Then the first minimum and the pseudo-second minimum were obtained from the first minimum of each group. In [14], it was proposed to use the group method in [13] to obtain the first minimum and a pseudo-second minimum. Then a more accurate but still approximate second minimum was calculated as the linear combination of the first minimum and the pseudo-second minimum.

This paper focuses on designing the efficient single-min based MS decoding algorithm for layered schedule. An efficient second minimum approximation MS (esmaMS) decoding is proposed. First, based on the state-of-the-art single-min based MS decoding, a skewed estimation of second minimum is got. Second, taking the change trend of the difference between first and second minima along with convergence degree into account, we propose an adjustment scheme to the previously mentioned skewed second minimum. The convergence degree of the decoding is indicated by a very simple metric which is easily obtained. Third, the pseudo-second minimum derived by group method is used as the upper bound to make the estimated second minimum more reasonable. Simulation results show that the proposed esmaMS has better error-correction performance than the prior works. More specifically, the esmaMS has less than 0.05 dB performance degradation compared with the NMS when frame error rate (FER) is at 10^{-3}.

The rest of this paper is organized as follows. Section II briefly describes the basic knowledge of the LDPC codes and the layered MS decoding algorithm. Section III presents the proposed esmaMS decoding in detail. The simulation results and performance comparisons are shown in section IV. Finally, conclusions are drawn in section V.

$$\boldsymbol{H}=\begin{bmatrix}1&1&1&0&0\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}$$

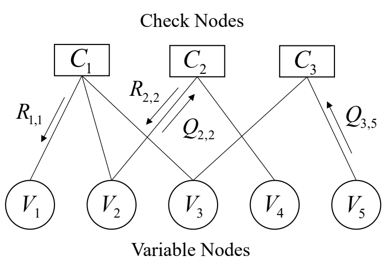


Fig. 1. The parity check matrix and Tanner graph.

II. Preliminaries

A. LDPC Codes

LDPC codes are a kind of linear block codes. For a binary (N, K) LDPC code, N is the code length, K is the number of information bits, and the code rate $R = K/N$. This code can be specified by a parity check matrix $\boldsymbol{H}$ with $M = N - K$ rows and N columns. As shown in Fig. 1, the code can be also represented by a Tanner graph with M check nodes (CNs) and N variable nodes (VNs) [15]. The CN C_i and the VN V_j correspond to the i-th row and j-th column of $\boldsymbol{H}$, respectively. There is a connection between C_i and V_j only if the element in the i-th row and j-th column of $\boldsymbol{H}$ is 1. The parity check matrix of QC-LDPC codes is constructed from a base matrix $\boldsymbol{H}_b$. A non-negative element s in $\boldsymbol{H}_b$ is expanded to a $z \times z$ sub-matrix which is obtained by cyclic shifting a $z \times z$ identity matrix s times. While the negative elements in $\boldsymbol{H}_b$ are expanded to a $z \times z$ zero matrix. Fig. 2 shows a 6×9 $\boldsymbol{H}$ which is constructed from a 2×3 $\boldsymbol{H}_b$ with $z = 3$.

$$\boldsymbol{H}_b=\begin{bmatrix}2&1&-1\\4&-1&0\end{bmatrix} \qquad \boldsymbol{H}=\left[\begin{array}{ccc:ccc:ccc}0&0&1&0&1&0&0&0&0\\1&0&0&0&0&1&0&0&0\\0&1&0&1&0&0&0&0&0\\\hdashline 0&1&0&0&0&0&1&0&0\\0&0&1&0&0&0&0&1&0\\1&0&0&0&0&0&0&0&1\end{array}\right]$$

Fig. 2. A 6×9 $\boldsymbol{H}$ constructed from a 2×3 $\boldsymbol{H}_b$ with $z = 3$.

B. Layered MS Decoding Algorithm

For QC-LDPC codes, rows of $\boldsymbol{H}$ can be divided into several layers. In layered schedule, the decoding is performed in parallel inside the layer and in sequence between layers, which reaches a faster convergence speed than the flooding schedule. The decoding of LDPC codes can be seen as message passing between CNs and VNs in the Tanner graph. The C2V message sent from C_i to V_j in the k-th iteration is denoted as $R_{i,j}^k$. The V2C message sent form V_j to C_i is denoted as $Q_{i,j}^k$. Let $L_j^{k,l}$ be the a-posteriori probability (APP) message of V_j in the k-th iteration input to the l-th layer. Let $\mathcal{N}(i)$ be the set of VNs connected to C_i and $\mathcal{M}(j)$ be the set of CNs connected to V_j, respectively. The layered MS decoding algorithm can be described as follows. At the beginning, the APP messages are initialized as the received channel log-likelihood ratio (LLR) while the C2V messages are initialized as zero, as shown in the following

$$L_j^{1,1} = LLR_j, \text{ for } 1 \leqslant j \leqslant N, \tag{1}$$

$$R_{i,j}^0 = 0, \text{ for } 1 \leqslant i \leqslant M, \ j \in \mathcal{N}(i). \tag{2}$$

In the k-th iteration, the V2C message $Q_{i,j}^k$ at l-th layer is calculated as

$$Q_{i,j}^k = L_j^{k,l} - R_{i,j}^{k-1}. \tag{3}$$

Then the C2V message $R_{i,j}^k$ is calculated as

$$R_{i,j}^k = \alpha \cdot \prod_{t \in \mathcal{N}(i) \backslash j} \operatorname{sign}(Q_{i,t}^k) \cdot \min_{t \in \mathcal{N}(i) \backslash j} \left|Q_{i,t}^k\right|, \tag{4}$$

where $\alpha \in (0, 1)$ is a scale factor. The APP message of V_j is updated as

$$L_j^{k,l+1} = R_{i,j}^k + Q_{i,j}^k. \tag{5}$$

Suppose that the parity check matrix has m layers. The codeword is denoted as $\boldsymbol{c} = (c_1, c_2, \cdots, c_N)$. After the k-th iteration is completed, the codeword $\boldsymbol{c}$ is estimated as

$$c_j = \begin{cases} 0, & \text{if } L_j^{k,m+1} \geqslant 0; \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

The decoding is terminated if a codeword satisfying all parity-check equations is found, i.e., $\boldsymbol{H}\boldsymbol{c}^T = \mathbf{0}$, or the maximum number of iterations is reached. Otherwise, $L^{k,m+1}$ is input to the first layer as $L^{k+1,1}$and the $(k+1)$-th iteration starts.

III. The Proposed esmaMS Decoding Algorithm

In this paper, we consider the finite word length implementation of the layered MS decoder. For the ease of description, denote the first and second minimum magnitudes of input V2C messages as m_1 and m_2, respectively. The SAMS decoding in [11] is the state-of-the-art work that takes the word length of C2V messages into account. Here em_2 is used to denote the estimated second minimum in [11]. em_2 is calculated by adding an offset γ on m_1 where γ is calculated as

$$\gamma = \left\lfloor ((2^{q-1} - 1) - m_1) \cdot \sigma/2 \right\rfloor, \tag{7}$$

where q is the word length of C2V messages and the value of σ depends on the code rate and the word length. Finally, em_2 is computed as

$$em_2 = m_1 + \gamma. \tag{8}$$

It will be shown that the SAMS decoding degrades error-correction performance visibly, especially in low to medium signal-to-noise ratio region. The performance degradation is caused by the difference between m_2 and em_2, which is denoted by d as follows

$$d = em_2 - m_2. \tag{9}$$

It is obvious that the larger the magnitude of d is, the severer the performance degradation is. Therefore, we pay our efforts to reduce the magnitude of d. In [12], the difference between m_1 and m_2 was studied (denoted as Δ). It was shown that Δ becomes larger as the decoding goes more convergent. In [12], the convergence degree is reflected by the number of non-matching bits between the signs of APP messages and external messages. However, it is inconvenient to compute the number of non-matching bits in the layered schedule. Inspired by [12], the relationship between d and the convergence degree is investigated in this section.

It is urgent to find a metric that can indicate the convergence degree and is undemanding to be observed for the layered decoder. Commonly, for various iterative decoding algorithms of LDPC codes, we say the decoding is completely convergent if $\boldsymbol{H}\boldsymbol{c}^T = \mathbf{0}$. There are M parity-check equations for $\boldsymbol{H}$ with M rows. These M parity-check equations are examined in parallel after each iteration. Intuitively, the number of parity-check equations that are not satisfied can reflect the convergence degree of the decoding. The less the number of unsatisfied parity-check equations is, the more convergent the decoding is.

As for QC-LDPC codes, $\boldsymbol{H}$ can be divided into multiple layers, and each layer contains z rows. It should be emphasized that the APP messages are continuously updated within one iteration since the layers are processed serially. Our goal is utilizing the latest information about the convergence degree to adjust γ in Eq. (7) when processing a certain layer. Denote

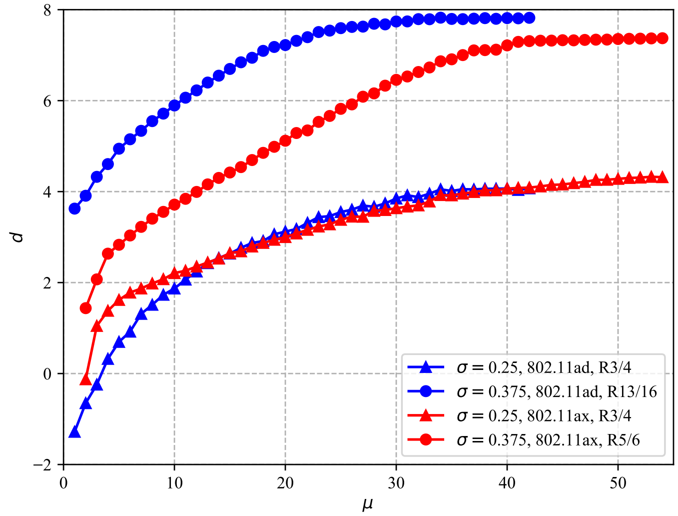


Fig. 3. Simulation results of the relationship between d and μ.

the partial parity-check matrix for the i-th layer as $\boldsymbol{H}_i$. After messages passing in the i-th layer, the codeword $\boldsymbol{c}$ is updated according to the latest APP messages. The latest $\boldsymbol{c}$ and $\boldsymbol{H}_i$ are used to count the number of unsatisfied parity equations in the i-th layer. Denote the number of unsatisfied check equations as μ, and it is apparent that the lower μ is, the more convergent the decoding is.

After choosing μ as a metric to measure the convergence degree during layered decoding, it is compulsory to establish the relationship between μ and d. The Monto-Carlo simulation is adopted and the corresponding results are shown in Fig. 3. Four codes, $(672, 504)$ and $(672, 544)$ IEEE 802.11ad [3] LDPC codes (code rate $3/4$ and $13/16$), $(1296, 972)$ and $(1296, 1080)$ IEEE 802.11ax [4] LDPC codes (code rate $3/4$ and $5/6$), are used as study cases. In Eq. (7), σ is set to 0.25 for two rate $3/4$ codes and σ is set to 0.375 for rate $13/16$ and $5/6$ codes. The simulation is performed under the additive white gaussian noise (AWGN) channel with the quadrature phase shift keying (QPSK) modulation. The E_b/N_0 varies from 2.5 dB to 4.0 dB, with a step of 0.5 dB. More than 10^8 frames are simulated for each code. The average value of d is calculated for each value of μ. As shown in Fig. 3, d becomes larger as μ increases. To bridge the gap between em_2 and m_2, we modify Eq. (7) as follows

$$\gamma = \left\lfloor ((2^{q-1} - 1) - m_1) \cdot \sigma/2 \right\rfloor + \beta, \tag{10}$$

where β is a dynamic factor related to μ. The expression of β depends on the used code and will be given in the next section.

Please note that β in Eq. (10) can be both positive and negative. Therefore, $em_2 = m_1 + \gamma$ is likely to be too large or too small. To make em_2 more accurate, we try to get the upper bound and the lower bound of em_2, denoted as m_u and m_l, respectively. Note that the real second minimum m_2 never be smaller than m_1, thus m_l is set to m_1. In [11], the upper bound m_u is set as the biggest value of a given word length, to be concrete, $m_u = 2^{q-1} - 1$. However, it is constant through the decoding and has nothing to do with the actual distribution of the V2C messages. Hence, setting $m_u = 2^{q-1} - 1$ is far from practical. In this paper, we pursue to find a dynamic and more accurate upper bound of em_2 during the procedure of finding m_1. The exMin-n MS decoding in [13], as shown in Fig. 4, divides the input V2C messages into two groups equally as G_1 and G_2 in stage1. Let m_{G_1} and m_{G_2} be the minimum of G_1 and G_2, respectively. In stage2, we have

$$m_1 = \min\{m_{G_1}, m_{G_2}\}, \tag{11}$$

$$pm_2 = \max\{m_{G_1}, m_{G_2}\}, \tag{12}$$

where pm_2 is pseudo-second minimum. In this method, m_1 is always the first minimum of input, but pm_2 is not necessarily the second minimum. Instead, pm_2 is always greater or equal to m_2, because m_2 may be in the same group of m_1 thus being excluded in stage1. In [13], em_2 is estimated by substracting a constant from pm_2. However, in our work, pm_2 is regarded as m_u, the upper bound of em_2. In this way, m_u is determined by the information on-the-fly and is always dynamic. According to the analysis above, our proposed esmaMS decoding can be summarized in Alg. 1.

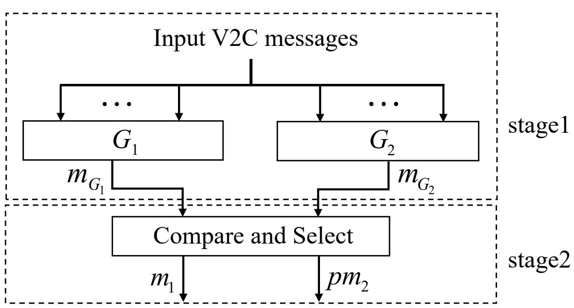


Fig. 4. Find m_1 and pm_2.

Algorithm 1: esmaMS Algorithm

Input: Input V2C message magnitudes, μ
Output: m_1 and em_2

Step. 1 Divide input V2C messages into two groups equally and find m_1 and pm_2 (as shown in Fig. 4).
Step. 2 Determine β according to the convergence degree μ (see Table I).
Step. 3 Calculate the offset γ using Eq. (10).
Step. 4 Determine em_2 as follows:
if m_1 == pm_2 **then**
    $em_2 = pm_2$;
else
    if $m_1 + \gamma \geq pm_2$ **then**
        $em_2 = pm_2$;
    else
        if $m_1 + \gamma \leq m_1$ **then**
            $em_2 = m_1$;
        else
            $em_2 = m_1 + \gamma$;
        end
    end
end

IV. Simulation Results

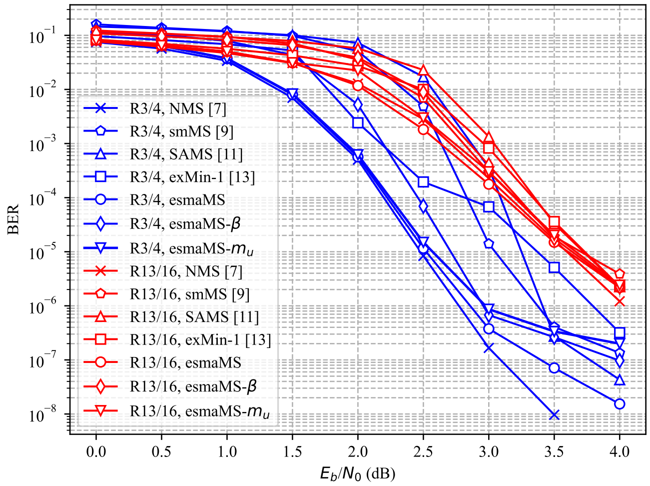
In this section, the error-correction performance of the proposed esmaMS decoding algorithm is exhibited and compared with NMS [7], exMin-n [13], smMS [9], SAMS [11] decoding algorithms, as shown in Fig. 5. And the average number of iterations of these decoding algorithms is shown in Fig. 6. The simulations are performed over the AWGN channel with the QPSK modulation. For all decoding algorithms, the maximum number of iterations is set to 30 and the scale factor is set to 0.75. The word lengths of channel LLR, APP messages, and C2V messages are 6 bits, 8 bits, and 7 bits, respectively. Other simulation parameters are shown in Table I and the specific expression of β for each code is shown in Eqs. (13a), (13b), (14a), and (14b). To show the effect of the single factor on the error-correction performance, esmaMS-β and esmaMS-m_u are also given. The former represents the esmaMS with modification in Eq. (10) but without the limitation of the upper bound m_u, while the latter represents esmaMS with the limitation of the upper bound m_u but without the modification in Eq. (10).

A. Error-Correction Performances

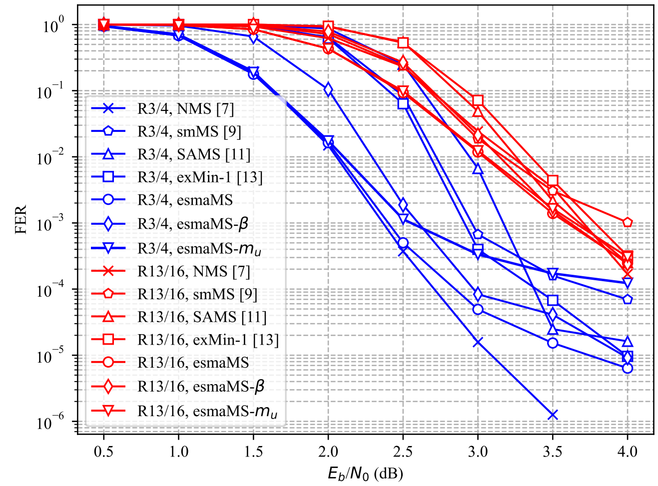
Fig. 5 (a) shows the bit error rate (BER) curves of various decoding algorithms for $(672, 504)$ and $(672, 544)$ IEEE 802.11ad LDPC codes (code rate $3/4$ and $13/16$). The corresponding FER curves are shown in Fig. 5 (b). Fig. 5 (c) shows the BER curves of various decoding algorithms for $(1296, 972)$ and $(1296, 1080)$ IEEE 802.11ax LDPC codes (code rate $3/4$ and $5/6$). The corresponding FER curves are shown in Fig. 5 (d). As can be seen, the proposed esmaMS decoding algorithm has about 0.2 dB to 1 dB performance gain at FER = 10^{-3} for different LDPC codes, compared with the state-of-the-art finite word length considering single-min based decoding algorithm, the SAMS [11]. Besides, the proposed esmaMS decoding algorithm shows better error-correction performance in a wide range of E_b/N_0 than other single-min based MS decoding algorithms. In addition, it shows about only 0.05 dB performance loss at FER = 10^{-3}, compared with the NMS decoding algorithm.

B. Average Number of Iterations

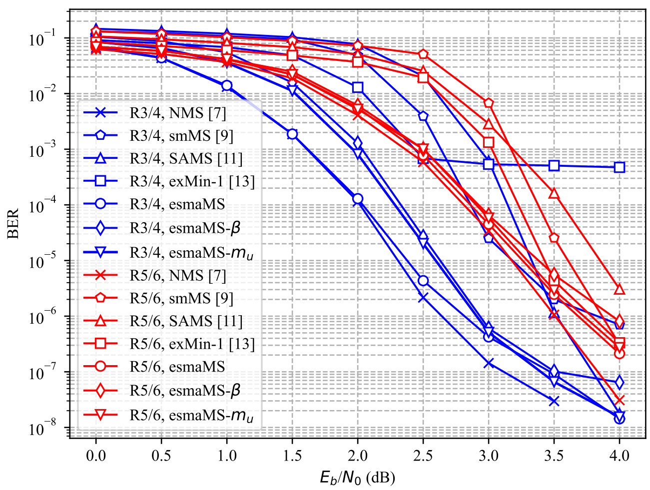
Fig. 6 (a) shows the average number of iterations of various decoding algorithms for IEEE 802.11ad rate $3/4$ and rate $13/16$ LDPC codes. Fig. 6 (b) shows the average number of iterations of various decoding algorithms for IEEE 802.11ax rate $3/4$ and rate $5/6$ LDPC codes. As can be seen, the proposed esmaMS decoding algorithm has a much faster convergence speed, compared with the SAMS decoding algorithm [11]. Besides, the proposed esmaMS decoding algorithm has lower average number of iterations in a wide range of E_b/N_0 than other single-min based MS decoding algorithms. Furthermore, it shows about only 3.4% higher average number



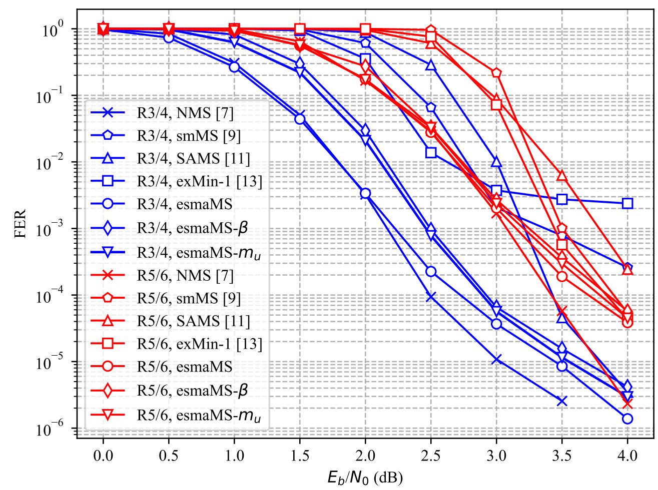
(a) BER results of the $(672, 504)$ and $(672, 544)$ IEEE 802.11ad LDPC codes (code rate $3/4$ and $13/16$).



(b) FER results of the $(672, 504)$ and $(672, 544)$ IEEE 802.11ad LDPC codes (code rate $3/4$ and $13/16$).



(c) BER results of the $(1296, 972)$ and $(1296, 1080)$ IEEE 802.11ax LDPC codes (code rate $3/4$ and $5/6$).



(d) FER results of the $(1296, 972)$ and $(1296, 1080)$ IEEE 802.11ax LDPC codes (code rate $3/4$ and $5/6$).

Fig. 5. Simulation results of error-correction performance.

of iterations at FER = 10^{-3}, compared with the NMS decoding algorithm.

TABLE I
SIMULATION PARAMETERS

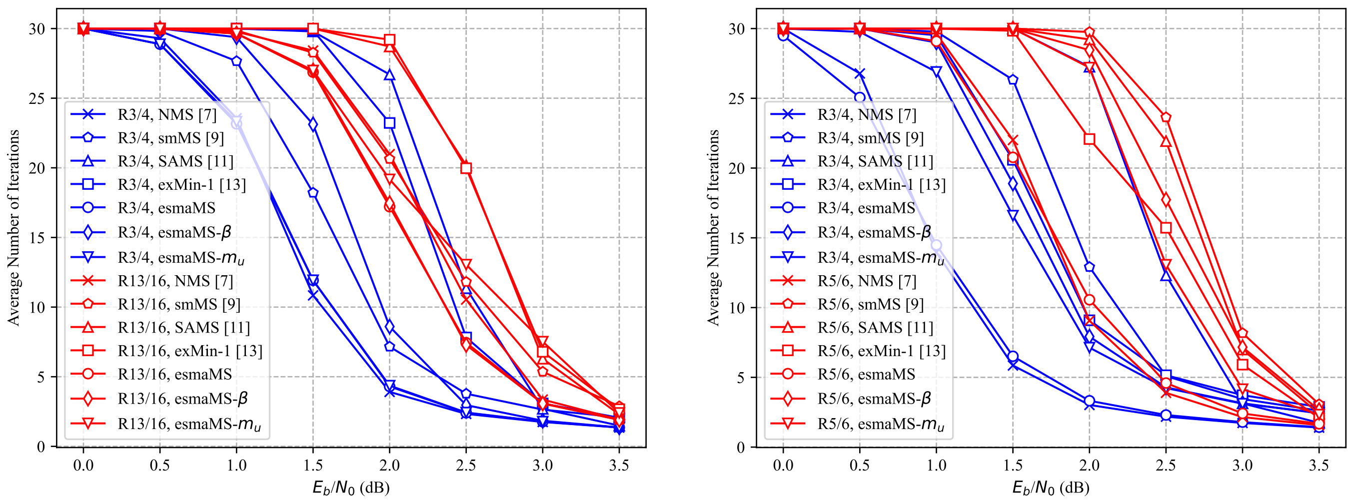
Standards	Code Rate	σ	β (for esmaMS)
IEEE 802.11ad	3/4	0.25	In Eq. (13a)
	13/16	0.375	In Eq. (13b)
IEEE 802.11ax	3/4	0.25	In Eq. (14a)
	5/6	0.375	In Eq. (14b)

$$\beta = \begin{cases} \lfloor -0.3\mu + 4.0 \rfloor, & \text{if } \mu \leqslant 20; \\ -2, & \text{otherwise.} \end{cases} \tag{13a}$$

$$\beta = \begin{cases} \lfloor -0.2\mu - 3.5 \rfloor, & \text{if } \mu \leqslant 20; \\ -7, & \text{otherwise.} \end{cases} \tag{13b}$$

$$\beta = \begin{cases} \lfloor -0.40\mu + 1.0 \rfloor, & \text{if } \mu \leqslant 5; \\ \lfloor -0.08\mu - 0.6 \rfloor, & \text{else if } 5 < \mu \leqslant 30; \\ -3, & \text{otherwise.} \end{cases} \tag{14a}$$

$$\beta = \begin{cases} -\mu, & \text{if } \mu \leqslant 3; \\ \lfloor -0.125\mu - 3.625 \rfloor, & \text{else if } 3 < \mu \leqslant 35; \\ -7, & \text{otherwise.} \end{cases} \tag{14b}$$



(a) Average number of iterations of the $(672, 504)$ and $(672, 544)$ IEEE 802.11ad LDPC codes (code rate $3/4$ and $13/16$).

(b) Average number of iterations of the $(1296, 972)$ and $(1296, 1080)$ IEEE 802.11ad LDPC codes (code rate $3/4$ and $13/16$).

Fig. 6. Simulation results of average number of iterations.

V. Conclusion

In this paper, a single-min based layered MS decoding algorithm named esmaMS is proposed for QC-LDPC codes. Based on the prior SAMS decoding algorithm, the estimated second minimum is modified according to the convergence degree during decoding, which makes the estimated second minimum more accurate. Besides, the estimated second minimum is limited by the upper and lower bound. The simulation results show that the proposed esmaMS decoding algorithm offers better error-correction performance and lower average number of iteratioins than prior arts. Compared with the NMS decoding algorithm, the proposed esmaMS decoding causes minor performance degradation and slight extra iteration at practical FER point. Based on the layered schedule, the proposed scheme provides a promising solution for the low-complexity decoder implementation of QC-LDPC codes.

References

[1] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[2] D. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[3] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band*, Std., 2012.

[4] *IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN*, Std., 2021.

[5] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[6] M. Fossorier, M. Mihaljevic, and H. Imai, “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation,” *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 673–680, 1999.

[7] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, “Reduced-complexity decoding of LDPC codes,” *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

[8] D. Hocevar, “A reduced complexity decoder architecture via layered decoding of LDPC codes,” in *IEEE Workshop on Signal Processing Systems (SIPS)*, 2004, pp. 107–112.

[9] A. Darabiha, A. Carusone, and F. Kschischang, “A bit-serial approximate min-sum LDPC decoder and FPGA implementation,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2006, pp. 149–152.

[10] F. Angarita, J. Valls, V. Almenar, and V. Torres, “Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 7, pp. 2150–2158, 2014.

[11] H. Lopez, H.-W. Chan, K.-L. Chiu, P.-Y. Tsai, and S.-J. J. Jou, “A 75-Gb/s/mm2 and energy-efficient LDPC decoder based on a reduced complexity second minimum approximation min-sum algorithm,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 926–939, 2020.

[12] Q. Zhang, S. Song, and Z. Wang, “Low-complexity dynamic single-minimum min-sum algorithm and hardware implementation for LDPC codes,” in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2022, pp. 1–5.

[13] I. Tsatsaragkos and V. Paliouras, “Approximate algorithms for identifying minima on min-sum LDPC decoders and their hardware implementation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 766–770, 2015.

[14] J. M. Català-Pérez, J. O. Lacruz, F. García-Herrero, J. Valls, and D. Declercq, “Second minimum approximation for min-sum decoders suitable for high-rate LDPC codes,” *Circuits, Systems, and Signal Processing*, vol. 38, no. 11, pp. 5068–5080, 2019.

[15] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on information theory*, vol. 27, no. 5, pp. 533–547, 1981.