TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering

A Weighted Finite-Alphabet Message Passing Decoder for High-Speed Optical Communication

by

L.M. Paulissen

MSC THESIS

Assessment committee	
Chair:	prof. dr. ir., J.W.M., Bergmans
Member 1:	dr., A., Alvarado
Member 2:	dr., A.K., Balatsoukas Stimming
Advisory member 1:	MSc, K., Wu

Graduation	
Program:	Electrical Engineering
Capacity group:	Signal Processing Systems Group
Supervisor:	dr., A., Alvarado
Date of defense:	April 4, 2022
Student ID:	0953065
Study load (ECTS):	45

A Weighted Finite-Alphabet Message Passing Decoder for High-Speed Optical Communication

Lotte Paulissen

***Abstract*—Finite-alphabet iterative decoders can significantly reduce the complexity of low-density parity-check (LDPC) decoders which is required to meet the strict throughput and latency requirements of future optical communication systems. The sign-preserving min-sum (SP-MS) decoder is a min-sum based finite alphabet message passing decoder that always preserves the sign of the message, even when a message is unreliable. Therefore, each message contributes to the convergence of the decoder and it shows significant performance and complexity improvements compared to the quantized offset min-sum (OMS) decoder. In this paper, we propose the weighted sign-preserving min-sum (WSP-MS) decoder that introduces iteration dependent weighting factors to the incoming messages of the SP-MS decoder. Monte Carlo simulations show that the WSP-MS decoder can achieve gains of up to 0.16 dB when compared to the SP-MS decoder with equal quantization bits. Moreover, the WSP-MS outperforms the floating-point BP decoder with a gain of up to 0.18 dB. Additionally, we applied the SP-MS decoder on an irregular LDPC code which shows the occurrence of an early error floor. This error floor can be mitigated by more than 2 orders of magnitude for the WSP-MS decoder.**

***Index Terms*—Low-complexity decoding, LDPC, quantization, forward error correction**

I. Introduction

The throughput of optical devices has increased at enormous speed over the past decades and devices with data rates of 400 Gbps are already available on the market. To provide reliable transmission at such high data rates, forward error correction (FEC) can be used to increase the performance of optical communication channels. Due to the strict throughput and latency requirements of the channel, it is a challenging task to find a FEC code that provides high performance at relatively low computational complexity. Soft-decision (SD) decoding can achieve performance close to the Shannon limit [1], but it requires a large amount of hardware and energy resources to enable the exchange of soft messages inside the decoder. The computational complexity can be reduced significantly when binary messages are exchanged inside the decoder. These hard-decision (HD) decoders can achieve higher throughput at the cost of decreased performance relatively far away from channel capacity. Product codes (PCs) and staircase codes (SCCs) are FEC coding schemes that are commonly used for HD decoding [2],[3].

An approach to lower the complexity of FEC schemes, while still achieving good performance is hybrid decoding. Hybrid HD-SD decoding uses an inner SD-FEC code and an outer HD-FEC code. The inner code lowers the bit-error rate (BER) below a predetermined threshold, while the outer code corrects the majority of the errors. In [4] and [5], the authors show that the decoding complexity of hybrid decoders can be reduced by 45% and 71%, respectively, at the same net coding gain (NCG) compared to the normalized min-sum (MS) algorithm in [6]. Hybrid decoding is also used in 400ZR [7], i.e., an optical communication standard that allows transmission of 400 Gbps over 100 km, which uses a concatenation of an inner SD double-extended Hamming code with an outer HD-SCC.

Low-density parity check (LDPC) codes invented by Gallager [8] are a popular area of research for high-speed FEC applications, due to their near-capacity performance and the capability of a high degree of parallelism in the decoder. The belief-propagation (BP) [9] algorithm for LDPC codes has excellent performance [1], yet has very high complexity due to the hyperbolic tangent update rule. Simplified BP-based algorithms like the min-sum (MS) and offset min-sum (OMS) algorithms proposed in [10], [11] reduce the complexity of the update rules, at the cost of slightly deteriorated performance in the waterfall region.

Various low-complexity BP-based algorithms that can achieve near-BP performance are presented in previous studies. However, the simplification of update rules is not sufficient to achieve high throughputs. The main reason for high complexity is the iterative exchange of soft messages in an SD decoder [12]. Therefore, a promising solution to achieve higher throughputs is quantized decoders, where the exchanged messages are discretized and presented by a small number of bits. Clipping and quantization of MS and OMS decoders have been studied in [11],[13]. These works show that six-bit quantized decoders approach, or even outperform the floating-point BP decoder in terms of BER while greatly reducing the complexity.

The idea of using binary message passing (BMP) decoders for LDPC codes was introduced by Gallager [8]. Decreasing the amount of bits used for each message is the most straightforward way to reduce the complexity in the decoder. This idea was extended by [14], where soft messages from the channel are combined with weighted binary messages from the check nodes (CNs) to perform the HD variable node (VN) update rule. Ternary message passing (TMP) and quaternary message passing (QMP) decoders which exploit soft information from the channel are described in [15]. Other decoders that only exchange binary messages are stochastic [16] and relaxed half-stochastic [17] decoders. The binary exchanged messages are probabilistic bitstreams and decoding is done in a bit-serial manner. These decoders can achieve better BER performance than the floating-point BP algorithm, at the cost of a higher number of iterations and therefore higher latency penalties.

The finite alphabet iterative decoding (FAID) algorithm proposed in [18] uses look-up tables instead of BP-based update rules for the VN update rule. The mapping of update rules reduces the complexity significantly while still being able to approach the BER performance of the BP algorithm with only 3 quantization bits. In [19], an FAID with full-parallel architecture is implemented with 3 quantization bits, that can achieve a throughput of 588 Gbps. However, a disadvantage of these FAIDs is that the message update rules usually cannot be represented by simple arithmetic circuits.

In [20], a Sign-Preserving Min-Sum (SP-MS) decoder is proposed which is a low-complexity finite precision iterative decoder that prevents erased messages from occurring. By adding a simple term to the VN update rule of the (O)MS decoder, a sign can always be assigned to unreliable messages that would typically be erased without increasing the number of quantization bits. Due to this additional available information in the decoder, the performance and convergence speed can be increased compared to the OMS decoder with little added complexity. Exchanged messages in the SP-MS decoder are defined within a finite alphabet of $q_m = 2, 3$ or 4 quantization bits and relatively simple update rules are used to keep the complexity of the decoder low. The channel messages are constructed from an alphabet of $q_c = 3$ or 4 bit quantized log-likelihood ratios (LLRs), such that $q_c \geq q_m$. Several studies on low-complexity quantized decoders [11], [13], [18], [21] indicate using an alphabet of size $2^{q_c} - 1$, while the SP-MS decoder exploits the full alphabet size 2^{q_c}.

In this paper, we propose the Weighted Sign-Preserving Min-Sum (WSP-MS) decoder that introduces iteration dependent weighting factors to the incoming messages before adding them to the quantized channel LLRs. Due to the iterative exchange of messages, more bits are likely to be correct at each iteration. Therefore, we can define the importance of incoming messages in the update rules by applying monotonically increasing weights to the incoming messages throughout the decoding process. We show that the $(3, 4)$-bit and $(4, 4)$-bit WSP-MS decoders can outperform the floating-point BP decoder in the high SNR regime. Also, the WSP-MS decoder shows gains up to 0.16 dB compared to the by density evolution optimized SP-MS decoder in [20] for a $(N = 2048)$-regular LDPC code. For an $(N = 17664)$-irregular LDPC code, we show that the WSP-MS decoder can mitigate the appearance of an early error floor by adding weighting factors that depend on both the VN degree and the iteration number.

The remainder of this paper is organized as follows. Sec. II gives a general introduction to LDPC codes, the system model and quantized OMS decoders. The quantization and update rules for the SP-MS decoder and WSP-MS decoder are presented in Sec. III. Sec. IV shows the simulation results of the (W)SP-MS decoder for regular and irregular LDPC codes and a conclusion is given in Sec. V.

II. PRELIMINARIES

An LDPC code $\mathcal{C}$ is a special type of linear block code that can be represented by a $M \times N$ sparse parity check matrix $\boldsymbol{H}$, where M is the number of parity check bits. The code rate of an LDPC code is given by $R = \frac{K}{N}$, where $K = N - M$. A graphical representation of an LDPC code can be given by a Tanner graph, which is a bipartite graph consisting of M check nodes $\mathrm{CN}_m,\ m = 1, ..., M$ and N variable nodes $\mathrm{VN}_n,\ n = 1, ..., N$. CN_m is connected to VN_n if and only if $\boldsymbol{H}_{mn} = 1$. Each check node is connected to d_c variable nodes and each variable node is connected to d_v check nodes. When d_c and d_v are constant among all CNs and VNs, the LPDC code is said to be regular, otherwise the LDPC code is irregular. Irregular LDPC codes are characterized by the degree distribution polynomials $\lambda(X) = \sum_{d=1}^{d_v} \lambda_d X^{d-1}$ and $\rho(X) = \sum_{d=1}^{d_c} \rho_d X^{d-1}$, where λ_d and ρ_d are the fraction of all edges connected to degree-d VNs and CNs, respectively.

In general, an LDPC code is a concatenated code where the VNs represent the inner code and the CNs can be seen as the outer code, because the CNs are not directly connected to the channel output. Each CN performs a single parity check operation on the incoming extrinsic messages before sending to the connected VNs according to the Tanner graph. The VNs represent a repetition code, which computes the extrinsic information from both the incoming messages and the channel and then sends it back to the CNs. The process of iterative exchange of information between CNs and VNs occurs until one of the stopping criteria occurs. The first criterion is when the maximum amount of iterations L_{tot} is reached. The second criterion is when a valid codeword is detected at the VN.

A. System model

Let $\boldsymbol{x} = (x_1, x_2, ..., x_N) \in \{0, 1\}^N$ be a codeword of $\mathcal{C}$ such that $\boldsymbol{H}\boldsymbol{x}^T = \boldsymbol{0}$. Every bit x_n $(n = 1, 2, \dots, N)$ is modulated with binary-phase shift keying (BPSK) and transmitted over the binary-input additive white Gaussian noise (bi-AWGN) channel with noise variance σ^2. The channel output is given by

$$y_n = (-1)^{x_n} + z_n, \tag{1}$$

where $z_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = (2E_s/N_0)^{-1}$ and E_s/N_0 is the signal-to-noise ratio (SNR). A hard-decision is applied on the channel output to calculate the syndrome of the codeword. This hard-decision on y_n can be written as $\hat{\boldsymbol{x}} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_N\} = \{0, 1\}^N$. The codeword is a valid codeword of $\mathcal{C}$ when the syndrome vector is all-zero ($\boldsymbol{H}\hat{\boldsymbol{x}}^T = \boldsymbol{0}$). When decoding is done using binary messages only, the hard-decision outputs $\hat{\boldsymbol{x}}$ are used for iterative decoding. Otherwise, the demapper calculates the log-likelihood ratio (LLR) of the received bits according to

$$L_n = \ln\left(\frac{\Pr(y_n|x_n = 0)}{\Pr(y_n|x_n = 1)}\right) = \frac{2y_n}{\sigma^2}. \tag{2}$$

B. LLR and message quantization

In order to reduce the complexity of the decoder, the floating-point LLRs L_n are quantized to a certain number of precision bits. A unique alphabet can be defined that lists all possible values of the output LLR. For classical LDPC decoders, this alphabet is defined as $\mathcal{A} = \{-N_a, ..., -1, 0, +1, ..., +N_a\}$, where $N_a = 2^{q_c - 1} - 1$ and q_c is the number of precision bits. So, when the LLRs are quantized to $q_c = 3$ bits, the LLR alphabet consists of 7 possible values $\mathcal{A} = \{-3, -2, -1, 0, +1, +2, +3\}$. The quantized LLRs $I_n \in \mathcal{A}$ can be derived from the channel output LLRs as

$$I_n = \mathcal{S}(\lfloor \alpha L_n + 0.5 \rfloor, N_a), \; n = 1, ..., N, \tag{3}$$

where $\mathcal{S}(\cdot)$ is the saturation function clipping the values to $\{-N_a, +N_a\}$ when $|\lfloor \alpha L_n + 0.5 \rfloor|$ is larger than N_a. The channel gain factor α determines the scaling of the channel outputs to the alphabet values. This factor is a constant parameter that can be optimized to increase the performance of the decoder. The exchanged messages between VNs and CNs are often quantized to the same number of precision bits ($q_m = q_c$) and therefore the message and the LLR share the same alphabet.

C. Offset Min-Sum decoding

The quantized LLRs $\boldsymbol{I} = \{I_1, I_2, ..., I_N\}$ are used to initialize each VN of the OMS decoder. The messages are iteratively exchanged between VNs and CNs that perform update rules in order to converge towards a valid codeword. A VN-to-CN message at iteration ℓ is denoted as $m^{(\ell)}_{v_n \to c_m}$ and $m^{(\ell)}_{c_m \to v_n}$ denotes a CN-to-VN message at iteration ℓ. $v \in \mathcal{N}(c_m)$ represents all VNs that are connected to CN m and $v \in \mathcal{N}(c_m) \setminus \{v_n\}$ denotes all VNs connected to CN m except for VN n. The CN update rule of the OMS decoder is a commonly used BP-based approximation defined as

$$\begin{aligned} m^{(\ell)}_{c_m \to v_n} = &\left(\prod_{v \in \mathcal{N}(c_m) \setminus \{v_n\}} \operatorname{sign}\left(m^{(\ell)}_{v \to c_m} \right) \right) \cdot \\ &\left(\min_{v \in \mathcal{N}(c_m) \setminus \{v_n\}} \left(\left| m^{(\ell)}_{v \to c_m} \right| \right) \right), \end{aligned} \tag{4}$$

where $m^{(\ell)}_{c_m \to v_n} \in \mathcal{A}$ is automatically satisfied when $m^{(\ell)}_{v \to c_m} \in \mathcal{A}$. The VN update rule of the OMS decoder is based on the VN update rule of the BP decoder. It contains a quantization function $\Upsilon(\cdot)$ to ensure the outgoing messages belong to the alphabet $\mathcal{A}$. The VN update rule is given by

$$m^{(\ell+1)}_{v_n \to c_m} = \Upsilon \left(I_n + \sum_{c \in \mathcal{N}(v_n) \setminus \{c_m\}} m^{(\ell)}_{c \to v_n} \right), \tag{5}$$

where the quantization function is defined as

$$\Upsilon(m_s) = \operatorname{sign}(m_s) \cdot \mathcal{S}\left(\max\left(|m_s| - \varphi_v, 0 \right), N_a \right). \tag{6}$$

An offset factor φ_v is subtracted from the magnitude of the total message m_s to compensate for the overestimation of the update rules [11]. When $\varphi_v = 0$, the OMS decoder becomes the regular MS decoder.

The binary codeword can be derived from the CN-to-VN messages by first calculating the tentative update rule

$$\gamma^{(\ell)}_n = I_n + \sum_{c \in \mathcal{N}(v_n)} m^{(\ell)}_{c \to v_n}, \; n = 1, ..., N. \tag{7}$$

Subsequently, the decoded codeword $\hat{\boldsymbol{x}}$ is determined by applying a hard-decision on the sign of $\boldsymbol{\gamma}$ given by

$$\hat{x}_n = \begin{cases} (1 - \operatorname{sign}(I_n))/2, & \text{if } \gamma^{(\ell)}_n = 0 \\ (1 - \operatorname{sign}(\gamma^{(\ell)}_n))/2, & \text{otherwise} \end{cases} \tag{8}$$

for $n = 1, ..., N$.

III. Sign-Preserving Min-Sum decoding

The most important difference between the classical OMS decoder and the SP-MS decoder is the alphabet used for channel outputs and exchanged messages. The alphabet of the classical OMS decoder described in the previous section consists of $2^{q_c - 1}$ possible values, where one message is erased. The erased message carries no information and therefore is not contributing to the convergence of the decoder. The alphabet of the SP-MS decoder can take 2^{q_c} possible values. A sign is always assigned to a message in the SP-MS decoder, even when the magnitude of the message is zero. To achieve this, a sign preserving factor is added to the VN update rule, which ensures that the VN update rule never outputs an erased message. In this section, the quantization and message alphabets of the SP-MS decoder are described. Then, we show the update rules.

A. Quantization and message alphabet SP-MS

The channel LLR alphabet of the SP-MS decoder is defined as $\mathcal{B}_c = \{-N_{bc}, ..., -1, -0, +0, +1, ..., +N_{bc}\}$, with $N_{bc} = 2^{(q_c - 1)} - 1$ and q_c the precision bits for the channel LLRs. The quantization of the channel LLRs is slightly different from (3) due to the modified alphabet and is given by

$$I_n = (\operatorname{sign}(L_n), \mathcal{S}(\lfloor \alpha |L_n| \rfloor, N_{bc})), \tag{9}$$

so $I_n \in \mathcal{B}_c$ for $n = 1, ..., N$. The quantized LLRs I_n are used to initialize the VN-to-CN messages $m^{(\ell)}_{v_n \to c_m}$ at iteration $\ell = 0$. For the SP-MS decoder, we also consider the possibility that q_m - the number of precision bits used for the exchanged messages between VNs and CNs - is smaller than q_c. Therefore, the exchanged message alphabet is separately defined as $\mathcal{B}_m = \{-N_{bm}, ..., -1, -0, +0, +1, ..., +N_{bm}\}$. When $q_c = q_m$, both alphabets are equivalent and each VN is initialized as $m^{(0)}_{v_n \to c_m \in \mathcal{N}(v_n)} = I_n$. When $q_c > q_m$, the quantized LLRs need to be mapped to the alphabet of the messages for the initialization of the VNs according to $m^{(0)}_{v_n \to c_m \in \mathcal{N}_m} = \mathcal{S}(I_n, N_{bm})$. Furthermore, all messages in the SP-MS decoder are written in binary sign-magnitude representation such that the first bit is used for the sign of the message and the remaining bits are used for the magnitude of the message. For example, given $q_m = 3$, the messages $+0$, -0 and -3 are written in binary as 000, 100 and 111, respectively.

B. Update rules SP-MS decoder

The CN update rule of the SP-MS decoder is given by

$$\begin{aligned} m^{(\ell)}_{c_m \to v_n} = \Bigg(&\prod_{v \in \mathcal{N}(c_m) \setminus \{v_n\}} \operatorname{sign}\left(m^{(\ell)}_{v \to c_m} \right), \\ &\min_{v \in \mathcal{N}(c_m) \setminus \{v_n\}} \left(\left| m^{(\ell)}_{v \to c_m} \right| \right) \Bigg). \end{aligned} \tag{10}$$

When the incoming messages are in $\mathcal{B}_m$, the outgoing message $m^{(\ell)}_{c_m \to v_n}$ will automatically be in $\mathcal{B}_m$ as well. Note that the

CN update rule for the SP-MS decoder is similar to the CN update rule in (4).

To preserve the sign of the messages in the SP-MS decoder, an additional factor called the sign-preserving factor is added to the VN update rule. This sign-preserving factor $\mu^{(\ell)}_{v_n\to c_m}$ can be calculated according to

$$\mu^{(\ell)}_{v_n\to c_m} = \xi \operatorname{sign}(I_n) + \sum_{c\in\mathcal{N}(v_n)\setminus\{c_m\}} \operatorname{sign}\left(m^{(\ell)}_{c\to v_n}\right), \quad (11)$$

where ξ depends on the amount of incoming messages to ensure $\mu^{(\ell)}_{v_n\to c_m}$ always takes an odd value. ξ is defined as

$$\xi = \begin{cases} 0, & \text{if } d_v = 2 \\ 1, & \text{if } d_v > 2 \text{ and } \operatorname{mod}_2(d_v) = 1 \\ 2, & \text{if } d_v > 2 \text{ and } \operatorname{mod}_2(d_v) = 0 \end{cases} \quad (12)$$

The sign preserving factor is added to the update rule of the OMS decoder in (5), which results in the VN update rule of the SP-MS decoder given by

$$m^{(\ell+1)}_{v_n\to c_m} = \Psi\left(I_n + \frac{\mu^{(\ell)}_{v_n\to c_m}}{2} + \sum_{c\in\mathcal{N}(v_n)\setminus\{c_m\}} m^{(\ell)}_{c\to v_n}\right). \quad (13)$$

Because $I_n \in \mathcal{B}_c$, $m^{(\ell)}_{c\to v_j} \in \mathcal{B}_m$ and $\mu^{(\ell)}_{v_n\to c_m}$ is always an odd number, the total sum will never be an integer number. As a result, a sign can always be assigned to an outgoing message. To guarantee each VN-to-CN message takes a value in the alphabet $\mathcal{B}_m$, the quantization function $\Psi(\cdot)$ is defined as

$$\Psi(m_s) = \left(\operatorname{sign}(m_s), \mathcal{S}\left(\max\left(\lfloor |m_s| \rfloor - \varphi_v, 0\right), N_{bm}\right)\right), \quad (14)$$

where φ_v is an offset factor to be elaborated next in Sec. III-C. After performing the CN update rule, the binary codeword can be obtained by first calculating the tentative update rule given by

$$\begin{aligned} \gamma^{(\ell)}_n =& I_n + \frac{1}{2}\xi \operatorname{sign}(I_n) + \\ & \sum_{c\in\mathcal{N}(v_n)} \left(m^{(\ell)}_{c\to v_n} + \frac{1}{2}\operatorname{sign}\left(m^{(\ell)}_{c\to v_n}\right)\right). \end{aligned} \quad (15)$$

Given $\gamma^{(\ell)}_n$, the decoded codeword $\hat{\boldsymbol{x}}$ is determined according to (8). Subsequently, the decoder checks if one of the stopping criteria is met.

C. Offset model SP-MS decoder

When quantizing the VN update rule, an offset φ_v is subtracted from the sum of the factors in (6) and (14). This offset value is often a single constant value that is used for all messages, e.g., $\varphi_v = \{0, 1\}$. In [20], an offset model is proposed for the SP-MS decoder in which the offset value depends on the magnitude of the sum of the factors $|m_s|$. The offset model consists of three offset values $\boldsymbol{\varphi} = (\varphi_s, \varphi_a, \varphi_0)$, which are defined as

$$\varphi = \begin{cases} \varphi_s, & \text{if } N_{bm} < |m_s| \le N_{bm} + 1 \\ \varphi_a, & \text{if } 2 < |m_s| \le N_{bm}, \\ \varphi_0, & \text{if } 1 < |m_s| \le 2 \end{cases} \quad (16)$$

For messages with a magnitude larger than $N_{bm}+1$, no offset value is required because these messages are saturated to N_{bm} in (14). The offset values $\boldsymbol{\varphi} = (\varphi_s, \varphi_a, \varphi_0)$ together with the channel gain factor α can be optimized by density evolution to improve the decoding performance.

For irregular LDPC codes, the offset values depend on the degree distribution of the VNs. Therefore, an offset model that consists of different offset values for each VN degree is proposed in [20]. A distinction is made for VNs of degree $d_v = 2$, $d_v = 3$ and $d_v \ge 4$.

D. Weighted SP-MS decoder

The performance and convergence speed of the SP-MS decoder can be improved by applying a weighting factor to the incoming messages of the SP-MS decoder. This method is already used for 1- and 2-bit message passing LDPC decoders in [14], [15]. The weighting factor is used to adjust the importance of the magnitude of the incoming messages compared to the channel LLRs in the VN and tentative update rule. For the full precision decoder, by performing the update rules and exchanging messages iteratively, more bits are likely to be correct at each iteration. Therefore the importance of the incoming messages increases compared to the channel LLRs as the iteration number grows. For quantized decoders, such an increase of message importance can be realized by applying a weighting factor to the incoming messages, which is dependent on the current iteration number $w^{(\ell)}$. Hence, the performance of the decoder can be improved. The SP-MS decoder is modified to include the weighting factor, and is called the weighted sign-preserving min-sum (WSP-MS) decoder.

The VN update rule of (13) is adjusted for the WSP-MS decoder and can be written as

$$m^{(\ell+1)}_{v_n\to c_m} = \Psi\left(I_n + w^{(\ell)}_n\left(\frac{\mu^{(\ell)}_{v_n\to c_m}}{2} + \sum_{c\in\mathcal{N}(v_n)\setminus\{c_m\}} m^{(\ell)}_{c\to v_n}\right)\right). \quad (17)$$

The weighting factors are also applied to the incoming messages in the tentative update rule according to

$$\begin{aligned} \gamma^{(\ell)}_n =& I_n + \frac{1}{2}\xi \operatorname{sign}(I_n) + \\ & w^{(\ell)}_n \sum_{c\in\mathcal{N}(v_n)} \left(m^{(\ell)}_{c\to v_n} + \frac{1}{2}\operatorname{sign}\left(m^{(\ell)}_{c\to v_n}\right)\right). \end{aligned} \quad (18)$$

For irregular LDPC codes, the amount of incoming messages at each VN is not constant. A large amount of incoming messages usually requires a lower scaling factor than for a smaller d_v. Therefore, the weights depend on both the iteration number (ℓ) and the VN degree d_v. When an irregular LDPC code consists of VNs with degree $d_v = \{3, 6, 11, 12\}$ and $L_{tot} = 12$, a total of 48 different weights need to be optimized. To simplify the optimization of the weighting factors in this study, a weight vector $\boldsymbol{w_c}$ is determined where each value depends on the iteration number only. This vector is scaled

TABLE I
SIMULATION PARAMETERS FOR IEEE 802.3 ETHERNET CODE

Decoder	(q_m, q_c)	α	$(\varphi_s, \varphi_a, \varphi_0)$	L_{tot}
(W)SP-MS	(2, 3)	0.74	(1, –, –)	20
SP-MS	(3, 3)	0.74	(1, 1, 1)	20
WSP-MS			(0, 1, 0)	
(W)SP-MS	(3, 4)	1.22	(1, 1, 1)	14
(W)SP-MS	(4, 4)	1.18	(1, 1, 1)	14

with a constant factor $s^{(d_v)}$ according to the VN degree resulting in the weight matrix $\boldsymbol{w}$ given by

$$\boldsymbol{w} = \begin{bmatrix} s^{(3)} \\ s^{(6)} \\ s^{(11)} \\ s^{(12)} \end{bmatrix} \cdot \left[w_c^{(0)}, w_c^{(1)}, \dots w_c^{(11)} \right]. \quad (19)$$

In this case, the number of weight parameters is reduced from 48 to 16. For regular LDPC codes, the weight matrix reduces to a weight vector due to the constant VN degree, so the weights are solely dependent on the iteration number. The weights need to be chosen carefully, because non-optimal weights can significantly decrease the performance of the decoder. In this paper, the weight vectors are determined by selecting the one resulting in the lowest BER/FER performance from a set of random candidate weight vectors. Furthermore, to simplify parameter optimization and avoid overestimation, the weights satisfy the constraint $0 < w^{(\ell)} \leq w^{(\ell+1)}$.

IV. SIMULATION RESULTS WSP-MS DECODER

In this section, the BER and frame-error rate (FER) performance of the (q_m, q_c)-bit WSP-MS decoder will be presented and compared to the performance of the SP-MS decoder for $q_m \in \{2, 3, 4\}$ and $q_c \in \{3, 4\}$. The results are obtained from Monte Carlo simulations for (A) the $(d_v = 6, d_c = 32)$-regular $N = 2048$, $R = 0.841$ LDPC code according to the IEEE 802.3 standard for ETHERNET [22] and (B) irregular $N = 17664$, $R = 0.826$ LDPC code specified by the IEEE 802.3ca 25G/50G EPON standard [23].

The performance of the floating-point BP decoder is shown as a benchmark for both LDPC codes. In addition, the performance of the hard-decision on the LLR values from the channel (PreFEC) is given. A minimum of 500 frames are sent with at least 10 frame errors occurring for each SNR point.

A. Simulation results regular IEEE 802.3 ETHERNET code

The parameters of the SP-MS decoder for the $N = 2048$-regular IEEE 802.3 ETHERNET code are estimated with density evolution in [20]. In order to apply density evolution, the channel and the decoder must satisfy the symmetry constraints specified in [9], which is the case for the SP-MS decoder. The channel gain factor α and the offset values $(\varphi_s, \varphi_a, \varphi_0)$ are jointly optimized for each precision (q_m, q_c). The optimized parameters for the (W)SP-MS decoder are shown in Table I. Note that only the offset factors used for the $(3, 3)$-bit WSP-MS decoder are different from the parameters of the SP-MS decoder.

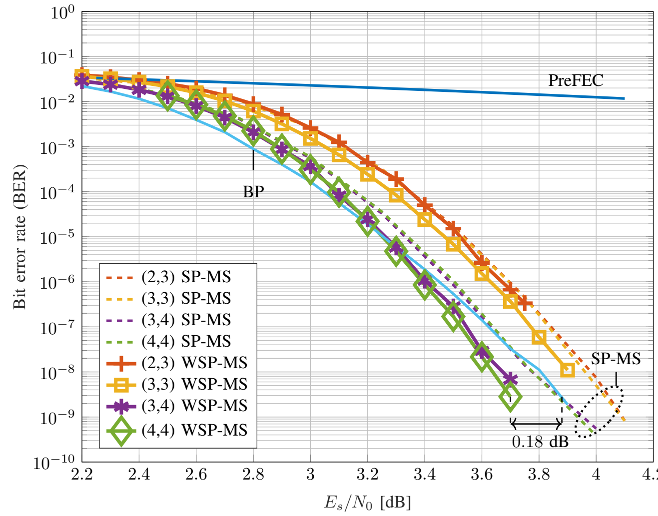

Fig. 1. BER performance WSP-MS decoder for (6,32)-regular LDPC code according to IEEE 802.3 ETHERNET code, $N = 2048$.

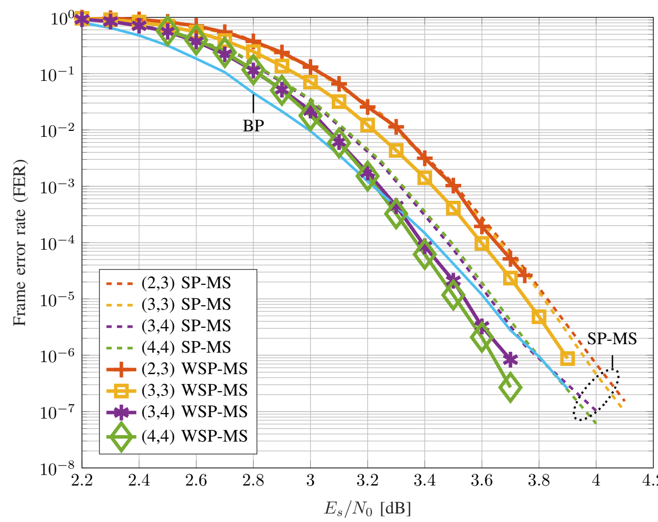

Fig. 2. FER performance WSP-MS decoder for (6,32)-regular LDPC code according to IEEE 802.3 ETHERNET code, $N = 2048$.

Fig. 1 shows the BER performance of the WSP-MS decoders, SP-MS decoders and the floating-point BP decoder with $L_{tot} = 14$. At BER of 10^{-7}, the $(3, 3)$-bit, $(3, 4)$-bit and $(4, 4)$-bit WSP-MS decoders show gains of 0.06 dB, 0.11 dB and 0.16 dB, respectively, compared to the SP-MS decoders with the same precision. As the SNR increases over 3.3 dB, the $(q_m, q_c = 4)$-bit WSP-MS decoders outperform the floating-point BP decoder and a gain up to 0.18 dB is obtained for the $(4, 4)$-bit WSP-MS decoder. The FER performance of the WSP-MS decoder is shown in Fig. 2. It can be seen in Fig. 2 that no early error floor is observed for BER up to 10^{-8} and FER up to 10^{-6} for the WSP-MS decoders.

Fig. 3 gives a comparison of the FER convergence of different decoders. The $(3, 3)$-bit $(3, 4)$-bit and $(4, 4)$-bit WSP-MS decoders converge faster than the SP-MS decoders with equal precision. After 8 iterations, the $(q_m, q_c = 4)$-bit WSP-MS decoders achieve the same FER performance as the $(3, 3)$-

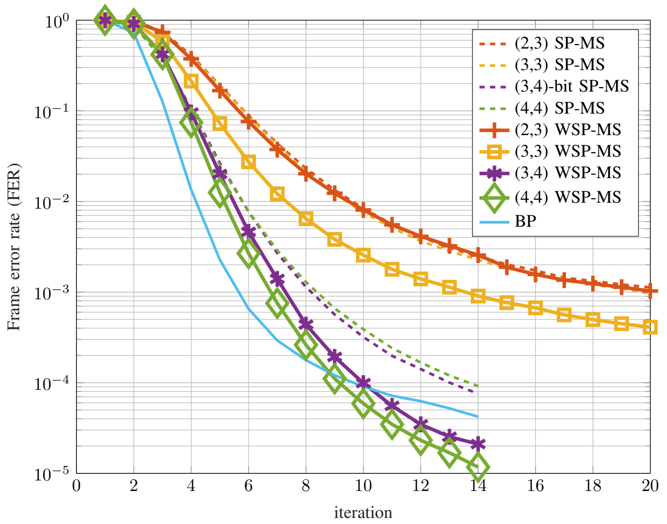

Fig. 3. FER convergence comparison WSP-MS decoder at $E_s/N_0 = 3.5$ dB for (6,32)-regular LDPC code according to IEEE 802.3 ETHERNET code, $N = 2048$.

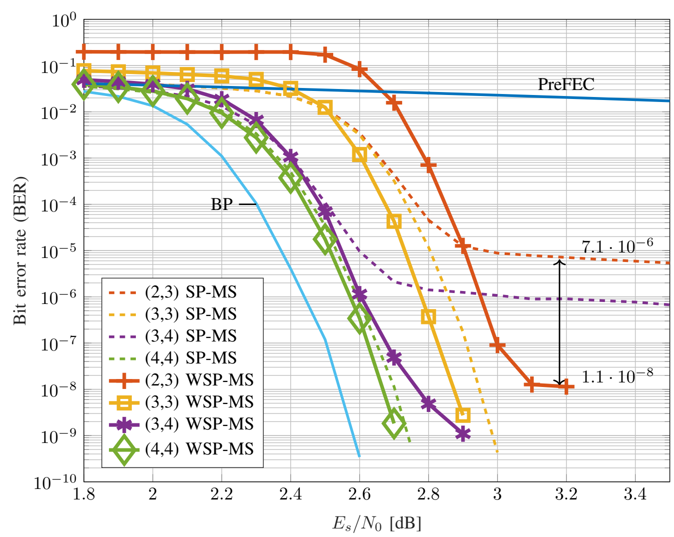

Fig. 4. BER performance WSP-MS decoder for $N = 17664$-irregular LDPC code according to IEEE 802.3ca standard, $L_{tot} = 12$.

TABLE II

SIMULATION PARAMETERS FOR IRREGULAR IEEE 802.3CA LDPC CODE

decoder	(q_m, q_c)	α	d_v	$(\varphi_s, \varphi_a, \varphi_0)$	L_{tot}
(W)SP-MS	(2, 3)	0.45	3	(0, –, –)	12
			≥ 4	(1, –, –)	
SP-MS	(3, 3)	0.45	3	(1, 1, 0)	
			≥ 4	(1, 1, 1)	
WSP-MS			3	(0, 0, 0)	
			≥ 4	(0, 0, 0)	
(W)SP-MS	(3, 4)	0.95	3	(1, 1, 0)	
			≥ 4	(1, 1, 1)	
(W)SP-MS	(4, 4)	1.15	3	(0, 1, 0)	
			≥ 4	(1, 1, 1)	

bit WSP-MS decoder after 20 iterations. Furthermore, we can observe that the $(q_m, q_c = 4)$-bit WSP-MS decoders outperform the floating-point BP decoder after 10 iterations.

B. Simulation results irregular IEEE 802.3ca code

The performance of the SP-MS decoder on irregular LDPC codes is not explicitly explained in [20]. Therefore, the parameters of the SP-MS decoder for the irregular IEEE 802.3ca code are first determined with Monte-Carlo simulations before presenting the simulation results of the WSP-MS decoder.

1) Parameter selection SP-MS decoder: The channel gain factor α and the offset values $\boldsymbol{\varphi}$ are important parameters of the SP-MS decoder. As mentioned before in Sec. III-C, the offset values depend on the VN degree d_v of a given LDPC code. The IEEE 802.3ca code is an irregular code with degree distribution polynomial $\lambda(X) = \frac{12800}{17664}X^2 + \frac{4352}{17664}X^5 + \frac{256}{17664}X^{10} + \frac{256}{17664}X^{11}$. DE conclusions drawn in [20] suggest a few guidelines for the selection of offset values for irregular LDPC codes:

- The optimal offset values are always 1 for $d_v \geq 4$,
- The offset value $\varphi_0 = 0$ for VNs with $d_v = 3$ and for precision $q_c = q_m \in \{3, 4\}$.

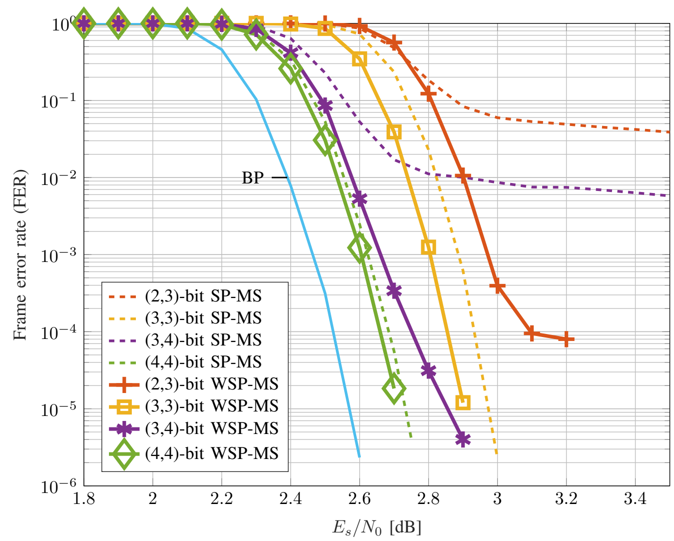

Fig. 5. FER performance WSP-MS decoder for $N = 17664$-irregular LDPC code according to IEEE 802.3ca standard, $L_{tot} = 12$.

These guidelines result in only two unknown offset values φ_s and φ_a for $d_v = 3$. By performing Monte Carlo simulations, the offset values φ_s and φ_a for $d_v = 3$ together with the channel gain factor α are chosen such that the lowest BER/FER is reached without the appearance of an early error floor. All these parameters for (W)SP-MS decoding of the irregular 802.3ca LDPC code are shown in Table II.

2) Performance analysis of WSP-MS decoder: Fig. 4 shows the BER performance of the WSP-MS decoder for the irregular IEEE 802.3ca code. We can observe that the WSP-MS decoder outperforms the SP-MS decoder for all precisions (q_m, q_c). The highest gain can be observed for the $(3, 3)$-bit WSP-MS decoder, where the WSP-MS decoder improves by 0.08 dB compared to the SP-MS decoder. For the $(2, 3)$-bit and $(3, 4)$-bit SP-MS decoders, we can observe an early error floor. The appearance of the error floor is caused by the small message magnitudes compared to the channel LLR magnitude. Hence, VNs of degree $d_v = 3$ struggle to overcome erroneous high channel LLR messages. By applying large weighting factors to the incoming messages at higher iteration numbers, the $(2, 3)$-bit and $(3, 4)$-bit WSP-MS decoders mitigate the early appearance of the error floor by more than two orders of magnitude.

The FER performance of the WSP-MS decoder is shown in Fig. 5. It can be seen that the error floors of the $(2, 3)$-bit and $(3, 4)$-bit SP-MS decoders are reduced by more than two orders of magnitude.

V. Conclusions

Strict throughput and latency requirements of optical communication channels require low-complexity FEC algorithms such as the WSP-MS decoder proposed in this paper. In this work, we applied iteration dependent weighting factors to the incoming messages of the low-complexity SP-MS decoder described in [20]. The performance of the SP-MS and WSP-MS decoders with $q_m \in \{2, 3, 4\}, q_c \in \{3, 4\}$ quantization bits is compared to the performance of the floating-point BP decoder for regular and irregular LDPC codes. We have shown that the WSP-MS decoder can achieve a gain up to 0.16 dB compared to the SP-MS decoder with equal precision (q_m, q_c). Monte Carlo simulations for the $(N = 2048)$-regular LDPC code show that the $(3, 4)$-bit and $(4, 4)$-bit WSP-MS decoders outperform the BP decoder, with a gain up to 0.18 dB. Furthermore, we have shown that the WSP-MS decoder holds the potential to mitigate the appearance of an early error floor. Therefore, the results show it is evident that the WSP-MS decoder can achieve large BER/FER performances, faster convergence speeds and a mitigation of the error floor. Moreover, there is still room for improvement, since the weighting vectors we used are obtained by simulations. Higher gains can be achieved if we analytically determine the optimal weighting vectors. Further research into optimization of the weight vectors, i.e. using density evolution, exit graphs or neural networks could be used to further optimize the performance of the WSP-MS decoder.

References

[1] S.-Y. Chung, G. 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. 2012.

[2] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, “Staircase Codes: FEC for 100 Gb/s OTN,” *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.

[3] A. Sheikh, A. Graell I Amat, and G. Liva, “Binary Message Passing Decoding of Product-Like Codes,” *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8167–8178, Dec. 2019.

[4] L. M. Zhang and F. R. Kschischang, “Low-Complexity Soft-Decision Concatenated LDGM-Staircase FEC for High-Bit-Rate Fiber-Optic Communication,” *J. Lightw. Technol.*, vol. 35, no. 18, pp. 3991–3999, Sep. 2017.

[5] M. Barakatain and F. R. Kschischang, “Low-Complexity Concatenated LDPC-Staircase Codes,” *J. Lightw. Technol.*, vol. 36, no. 12, pp. 2443–2449, Jun. 2018.

[6] D. A. Morero, M. A. Castrillon, F. A. Ramos, T. A. Goette, O. E. Agazzi, and M. R. Hueda, “Non-concatenated fec codes for ultra-high speed optical transport networks,” in *Proc. IEEE Glob. Telecommun. Conf.* Houston, TX, USA, Dec. 2011.

[7] “Implementation Agreement 400ZR,” *OIF Optical Internetworking Forum*, Mar. 2020.

[8] R. G. Gallager, “Low-Density Parity-Check Codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[9] T. J. Richardson and R. L. Urbanke, “The Capacity of Low-Density Parity-Check Codes under Message-Passing Decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[10] M. P. C. Fossorier, M. Mihaljević, and H. Imai, “Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes based on Belief Propagation,” *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

[11] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, “Reduced-Complexity Decoding of LDPC Codes,” *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.

[12] A. Graell I Amat, G. Liva, and F. Steiner, “Coding for optical communications - can we approach the shannon limit with low complexity?” in *Proc. Eur. Conf. Opt. Commun.* Dublin, Ireland, Sep. 2019.

[13] 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.

[14] G. Lechner, T. Pedersen, and G. Kramer, “Analysis and Design of Binary Message-Passing Decoders,” *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 601–607, Mar. 2010.

[15] F. Steiner, E. Ben Yacoub, B. Matuz, G. Liva, and A. G. I. Amat, “One and Two Bit Message Passing for SC-LDPC Codes with Higher-Order Modulation,” *J. Lightw. Technol.*, vol. 37, no. 23, pp. 5914–5925, Dec. 2019.

[16] S. S. Tehrani, A. Naderi, G.-A. Kamendje, S. Hemati, S. Mannor, and W. J. Gross, “Majority-Based Tracking Forecast Memories for Stochastic LDPC Decoding,” *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4883–4896, Sep. 2010.

[17] F. Leduc-Primeau, S. Hemati, S. Mannor, and W. J. Gross, “Relaxed Half-Stochastic Belief Propagation,” *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1648–1659, May 2013.

[18] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasic, “Finite Alphabet Iterative Decoders-Part I: Decoding Beyond Belief Propagation on the Binary Symmetric Channel,” *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4033–4045, Oct. 2013.

[19] R. Ghanaatian, A. Balatsoukas-Stimming, T. C. Müller, M. Meidlinger, G. Matz, A. Teman, and A. Burg, “A 588-Gb/s LDPC decoder based on finite-alphabet message passing,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 2, pp. 329–340, Feb. 2018.

[20] F. Cochachin, E. Boutillon, and D. Declercq, “Sign-Preserving Min-Sum Decoders,” *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6439–6454, Oct. 2021.

[21] X. Zhang and P. H. Siegel, “Quantized Iterative Message Passing Decoders with Low Error Floor for LDPC Codes,” *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 1–14, Jan. 2014.

[22] “IEEE Standard for Ethernet,” *IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012)*, pp. 1–4017, Mar. 2016.

[23] “IEEE Standard for Ethernet Amendment 9 Physical Layer Specifications and Management Parameters for 25 Gb/s and 50 Gb/s Passive Optical Networks.” Jul. 2020.

APPENDIX A
WEIGHT VECTORS WSP-MS DECODERS

TABLE III
WEIGHT VECTORS $\boldsymbol{w}^{(\ell)}$ FOR REGULAR IEEE 802.3 ETHERNET CODE

(q_m, q_c)	$w^{(0)}$	$w^{(1)}$	$w^{(2)}$	$w^{(3)}$	$w^{(4)}$	$w^{(5)}$	$w^{(6)}$	$w^{(7)}$	$w^{(8)}$	$w^{(9)}$	$w^{(10)}$	$w^{(11)}$	$w^{(12)}$	$w^{(13)}$	$w^{(14)}$	$w^{(15)}$	$w^{(16)}$	$w^{(17)}$	$w^{(18)}$	$w^{(19)}$
$(2,3)$	1	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	1.05	1.1	1.1	1.1	1.1	1.1
$(3,3)$	1	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.7	0.7	0.7	0.7
$(3,4)$	1	0.7	0.72	0.73	0.73	0.76	0.76	0.81	0.82	0.83	0.84	0.85	0.89	0.92	–	–	–	–	–	–
$(4,4)$	1	0.65	0.66	0.67	0.67	0.68	0.69	0.72	0.73	0.74	0.74	0.92	0.93	0.93	–	–	–	–	–	–

TABLE IV
SCALING FACTORS $s^{(d_v)}$ FOR IRREGULAR IEEE 802.3CA CODE

(q_m, q_c)	$s^{(3)}$	$s^{(6)}$	$s^{(11)}$	$s^{(12)}$
$(2,3)$	0.97	0.93	0.92	0.87
$(3,3)$	1.05	0.97	0.88	0.87
$(3,4)$	1	0.93	0.86	0.78
$(4,4)$	1	0.93	0.87	0.81

TABLE V
WEIGHT VECTORS $\boldsymbol{w_c}^{(\ell)}$ FOR IRREGULAR IEEE 802.3CA CODE

(q_m, q_c)	$w_c^{(0)}$	$w_c^{(1)}$	$w_c^{(2)}$	$w_c^{(3)}$	$w_c^{(4)}$	$w_c^{(5)}$	$w_c^{(6)}$	$w_c^{(7)}$	$w_c^{(8)}$	$w_c^{(9)}$	$w_c^{(10)}$	$w_c^{(11)}$
$(2,3)$	1	0.8	0.82	0.92	1	1.1	1.2	1.3	1.4	1.6	1.8	2.5
$(3,3)$	1	0.57	0.62	0.63	0.64	0.66	0.72	0.75	0.75	0.75	0.76	1.01
$(3,4)$	1	0.72	0.9	0.93	0.94	1.05	1.09	1.13	1.21	1.21	1.21	1.43
$(4,4)$	1	0.8	0.84	0.87	0.9	0.94	0.98	1.01	1.05	1.08	1.12	1.16