



LUND UNIVERSITY

Generalized Two-Magnitude Check Node Updating with Self Correction for 5G LDPC Codes Decoding

Zhou, Wei; Lentmaier, Michael

Published in:

SCC 2019 - 12th International ITG Conference on Systems, Communications and Coding

DOI:

[10.30420/454862047](https://doi.org/10.30420/454862047)

2019

Document Version:

Peer reviewed version (aka post-print)

[Link to publication](#)

Citation for published version (APA):

Zhou, W., & Lentmaier, M. (2019). Generalized Two-Magnitude Check Node Updating with Self Correction for 5G LDPC Codes Decoding. In *SCC 2019 - 12th International ITG Conference on Systems, Communications and Coding* (pp. 274-279). Article S8-3 VDE Verlag gmbh Berlin. <https://doi.org/10.30420/454862047>

Total number of authors:

2

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Generalized Two-Magnitude Check Node Updating with Self Correction for 5G LDPC Codes Decoding

Wei Zhou, Michael Lentmaier

Department of Electrical and Information Technology,

Lund University, Sweden

Email: wei.zhou@eit.lth.se, michael.lentmaier@eit.lth.se

Abstract—The min-sum (MS) and approximate-min* (a-min*) algorithms are alternatives of the belief propagation (BP) algorithm for decoding low-density parity-check (LDPC) codes. To lower the BP decoding complexity, both algorithms compute two magnitudes at each check node (CN) and pass them to the neighboring variable nodes (VNs). In this work we propose a new algorithm, ga-min*, that generalizes the MS and a-min* in terms of number of incoming messages to a CN. We analyze and demonstrate a condition to improve the performance when applying self-correction to the ga-min*. Simulations on 5G LDPC codes show that the proposed decoding algorithm yields comparable performance to the a-min* with a significant reduction in complexity, and it is robust against LLR mismatch.

I. INTRODUCTION

Low-density parity-check (LDPC) codes have become a channel coding solution in 5G New Radio (NR). There are a multitude of services in 5G NR. Enhanced mobile broadband (eMBB), being one of the services, demands high data rate with moderate reliability.

The standardization process of LDPC codes in the 3rd generation partnership project (3GPP) has focused on the trade-off between decoding performance and complexity. The sum-product (SP) algorithm [1], also referred to as belief propagation (BP) algorithm, provides near-optimal decoding performance. However, the check node (CN) processor in the SP algorithm computes a large number of non-linear functions, which results in high hardware complexity.

The min-sum (MS) [3] and approximate-min* (a-min*) algorithms [4] are alternatives to the SP decoding with lower complexity. Unlike the SP algorithm, where the CN processor generates a unique magnitude for each neighboring variable node (VN), the CN processor of the MS and a-min* decoder computes only two magnitudes in the outgoing messages. The a-min* algorithm, in which one of the messages in CN processor is identically computed to that in the SP algorithm, has negligible performance loss compared to the SP algorithm. While the MS algorithm, the CN processor of which approximates both magnitudes, can achieve more complexity reduction.

The approximated messages that CNs deliver in the MS algorithm degrade the performance [3]. To overcome this loss, the self-corrected MS (SCMS) algorithm is proposed

[5], where unreliable messages are erased at VNs in the MS algorithm. It is a powerful yet simple technique. In [6] the SCMS algorithm has a good performance in the error floor region, and it is robust on noisy hardware [7].

In the wireless communication scenario, due to the noise estimation mismatch in the detector, automatic gain control (AGC) and other front-end components, the log-likelihood ratios (LLRs) fed into the LDPC decoder can be modeled as a scaled version of the true received ones [13]. This situation, which we call LLR mismatch, can cause severe performance degradation.

Despite being a powerful technique, to our best knowledge self-correction (SC) has been only applied in the VNs of the MS algorithm to improve the performance. In this work, we identify a condition in the CN processing under which applying the SC can potentially benefit the performance. That is, if the critical messages of CNs in an iterative algorithm is over-estimated, applying the SC at the VNs can lead to a performance boost. As an example fitting this condition, we propose a new decoding algorithm, referred to as ga-min*, which generalizes the MS and a-min* algorithms in terms of the number of incoming messages to CNs. The performance of the proposed ga-min* and SC ga-min* (scga-min*) is close to the SP algorithm and comparable to the a-min* with reduced complexity. We also evaluate the robustness of various iterative algorithms. It turns out that the scga-min* with two incoming messages per CN shows superiority in performance in the presence of LLR mismatch.

The paper is organized as follows: Section II introduces LDPC codes and the finalized design in the 5G NR eMBB scenario. Section III gives an overview of the box-plus operator and different LDPC decoders. We propose the new algorithm, ga-min*, in Section IV. The performance of the (sc)ga-min*, in the absence and presence of LLR mismatch, is simulated and compared with other decoding algorithms in Section V. A complexity comparison between the ga-min* and a-min* is also presented there. Section VI concludes the paper.

II. LDPC CODES

An LDPC code \mathcal{C} can be described by a sparse parity-check matrix (PCM) \mathbf{H} . The codewords \mathbf{v} of \mathcal{C} is a set of vectors, the null space of which is \mathbf{H} , i.e., $\mathbf{v}\mathbf{H}^T = \mathbf{0}$.

This work at EIT, Lund University was jointly executed with and funded by Huawei Technologies Sweden AB, Lund, Sweden

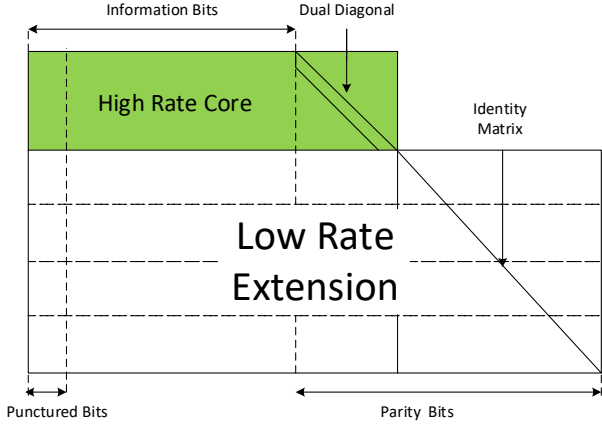


Fig. 1. A sketched PCM for finalized NR eMBB LDPC codes

LDPC codes can also be represented by Tanner graphs. A Tanner graph is a bipartite graph which consists of VNs and CNs. The VNs and CNs correspond to columns and rows of \mathbf{H} , respectively. Label the VNs with index n and the CNs with index m where $n, m = 1, 2, \dots$. The design of an LDPC code can start from the protograph, or base graph (BG), which is a small Tanner graph allowing multiple edges between VNs and CNs.

A. 5G NR eMBB LDPC codes

The 3GPP has finalized the design of 5G NR LDPC codes for eMBB service [10] [11]. To support a broad range of code lengths and rates, two BGs, BG1 and BG2 are considered. BG1 is used for codes of lengths $N = 512$ to 8448 and of rates $R = 8/9$ to $1/3$, while BG2 is used for codes of lengths $N = 40$ to 2560 and of rates $R = 2/3$ to $1/5$. Both BG1 and BG2 consist of two parts, a core graph representing the high rate codes, as well as an extension graph which includes more parity bits for lower rate codes. Fig. 1 shows a sketched PCM for the BG. The first two information bits are punctured to enhance the block error rate. The core graph adopts a dual diagonal structure for parity bits to simplify the encoding process. The extension graph has an equal amount of VNs and CNs. The VNs are degree one, forming an identity matrix.

B. Notations

Throughout the paper, we use the following notations regarding the LDPC code with a Tanner graph \mathcal{T} and message passing algorithms.

- N : number of VNs,
- M : number of CNs,
- K : number of information bits,
- R : code rate,
- $n \in \{1, 2, \dots, N\}$: an index of a VN in \mathcal{T} ,
- $m \in \{1, 2, \dots, M\}$: an index of a CN in \mathcal{T} ,
- d_c : CN degree,
- $\mathcal{M}(n)$: the index set of CNs in the neighborhood of VN n ,
- $\mathcal{N}(m)$: the index set of VNs in the neighborhood of CN m ,
- $T_{m \rightarrow n}$: the LLR from CN m to VN n .
- $L_{n \rightarrow m}$: the LLR from VN n to CN m .
- F_n : the LLR from channel for VN n .

III. ITERATIVE DECODING ALGORITHMS AND BOX-PLUS OPERATOR

In this section we give an overview of different decoding algorithms for LDPC codes. For the sake of clarity, all the CN processors in these algorithms are formulated with the box-plus operator [8]. We end this section by reviewing and proving some properties of box-plus operator, which do not commonly appear in coding textbooks or research papers. These properties can be used to analyze the approximated CN messages in the MS and a-min* algorithms.

A. Sum-Product Algorithm

Define a two-input operator \boxplus taking L_1 and L_2 as

$$L_1 \boxplus L_2 = \log \left(\frac{1 + e^{L_1 + L_2}}{e^{L_1} + e^{L_2}} \right),$$

here we call \boxplus the box-plus operator. The CN processing in the SP algorithm is

$$T_{m \rightarrow n} = \boxplus_{n' \in \mathcal{N}(m) - n} L_{n' \rightarrow m}, \quad (1)$$

where \boxplus is the summation operator of box-plus \boxplus .

B. Min-Sum Algorithm and Normalized Min-Sum Algorithm

The CN processing of the min-sum (MS) algorithm is given as follows:

$$T_{m \rightarrow n} = \left(\prod_{n' \in \mathcal{N}(m) - n} \text{sign}(L_{n' \rightarrow m}) \right) \cdot \min_{n' \in \mathcal{N}(m) - n} |L_{n' \rightarrow m}|. \quad (2)$$

The advantage of the MS algorithm lies in its simple operation. Unlike CN processing of the SP algorithm, which computes a unique value for every neighboring VN, in the MS algorithm only two different magnitudes are passed to VNs per CN. By introducing a normalizing factor α , the normalized min-sum (NMS) [3] can improve the performance of the MS algorithm.

C. The Approximate Min* Algorithm

The approximate min* (a-min*) decoder [4] has a negligible performance loss to the SP algorithm, and it generates two messages per CN as in the MS algorithm. The two messages, called *critical* and *non-critical message*, are defined and generated in Algorithm 1.

D. Self-Corrected Min-Sum

Self-corrected min-sum (SCMS) [5] performs the same CN processing as that of the MS algorithm and differs in the VN processing. Specifically, VN messages get erased, i.e., $L_{n \rightarrow m}^{(i)} = 0$, if there is a sign change between current iteration i and the previous one $i - 1$, as formalized below.

$$\tilde{L}_{n \rightarrow m}^{(i)} = F_n + \sum_{m' \in \mathcal{M}(n) - m} T_{m' \rightarrow n}^{(i)}$$

$$L_{n \rightarrow m}^{(i)} = \begin{cases} 0 & : \text{sign}(\tilde{L}_{n \rightarrow m}^{(i)}) \text{sign}(L_{n \rightarrow m}^{(i-1)}) < 0 \text{ and } L_{n \rightarrow m}^{(i-1)} \neq 0, \\ \tilde{L}_{n \rightarrow m}^{(i)} & : \text{otherwise.} \end{cases}$$

Algorithm 1 The CN processing of the a-min*

1. At CN m , find the incoming message of minimum magnitude and label VN n_1 where minimum message is from.
2. The message sent to VN n_1 , called *critical message*, is the same as the one computed by the SP algorithm rule in (1),

$$T_{m \rightarrow n_1} = \boxplus_{n' \in \mathcal{N}(m) - n_1} L_{n' \rightarrow m}. \quad (3)$$

3. For the rest of neighboring VNs of CN m , i.e., $n \in \mathcal{N}(m) - n_1$, the magnitude of *non-critical* messages is computed by applying the box-plus operator to all incoming messages, i.e., $L_{n' \rightarrow m}$, $n' \in \mathcal{N}(m)$, with an extrinsic sign that is the product of signs of $L_{n' \rightarrow m}$, $n' \in \mathcal{N}(m) - n$.

$$T_{m \rightarrow n} = \left(\prod_{n' \in \mathcal{N}(m) - n} \text{sign}(L_{n' \rightarrow m}) \right) \cdot \left| \boxplus_{n' \in \mathcal{N}(m)} L_{n' \rightarrow m} \right|. \quad (4)$$

In this way, unreliable messages are detected by sign fluctuation. By simply erasing these unreliable messages, SCMS can achieve close-to-optimal decoding performance [5].

E. Some Properties of the Box-Plus Operator

Property 1. The box-plus operator $L_1 \boxplus L_2$ is symmetric about point $(0, 0)$, i.e., $L_1 \boxplus L_2 = -L_1 \boxplus -L_2$.

Property 2. The box-plus operator $L_1 \boxplus L_2$ is odd with respect to one of its input, L_1 or L_2 , i.e., $L_1 \boxplus -L_2 = -(L_1 \boxplus L_2) = -L_1 \boxplus L_2$.

Proof of Property 1 and 2: From [2] the box-plus can be written as

$$L_1 \boxplus L_2 = \text{sign}(L_1)\text{sign}(L_2) \min(|L_1|, |L_2|) + s(L_1, L_2), \quad (5)$$

where $s(L_1, L_2) = \log(1 + e^{-|L_1+L_2|}) - \log(1 + e^{-|L_1-L_2|})$. Since $s(L_1, L_2) = s(-L_1, -L_2)$ and $s(-L_1, L_2) = -s(L_1, L_2) = s(L_1, -L_2)$, Property 1 and 2 can immediately follow. ■

Property 3. The following three equities hold for the box-plus operator $L_1 \boxplus L_2$.

$$\begin{aligned} L_1 \boxplus L_2 &= \text{sign}(L_1)\text{sign}(L_2) [\min(|L_1|, |L_2|) + s(|L_1|, |L_2|)] \\ &= \text{sign}(L_1)\text{sign}(L_2) (|L_1| \boxplus |L_2|) \\ &= \text{sign}(L_1)\text{sign}(L_2) |L_1 \boxplus L_2|. \end{aligned}$$

Proof: The first equality follows from (5) and by the fact that $s(L_1, L_2) = \text{sign}(L_1)\text{sign}(L_2)s(|L_1|, |L_2|)$. The second equality follows by substituting $|L_1|$ and $|L_2|$ as the inputs of (5). The third equality follows from the fact that the box-plus operator with two non-negative inputs always gives a non-negative value. ■

Property 4.

$$|L_1 \boxplus L_2| \leq \min(|L_1|, |L_2|). \quad (6)$$

Proof:

$$\begin{aligned} |L_1 \boxplus L_2| &= |L_1| \boxplus |L_2| = \min(|L_1|, |L_2|) + s(|L_1|, |L_2|) \\ &\leq \min(|L_1|, |L_2|), \end{aligned}$$

where the first two equalities follow from Property 3 and last equality follows from $s(|L_1|, |L_2|) \leq 0$. ■

It follows from (6) we have the two propositions.

Proposition 1. The CN message of the MS algorithm over-estimates, i.e., has a larger value in magnitude than, the one in the SP algorithm.

Proof: Following (1), the magnitude of CN message $|T_{m \rightarrow n}|$ in the SP algorithm, is

$$|T_{m \rightarrow n}| = \left| \boxplus_{n' \in \mathcal{N}(m) - n} L_{n' \rightarrow m} \right| \leq \min_{n' \in \mathcal{N}(m) - n} |L_{n' \rightarrow m}|,$$

which is the magnitude of CN message of the MS algorithm (2), the inequality follows by repeated use of (6). ■

Proposition 2. [4] The CN message to VN n , $n \neq n_1$, in the a-min* under-estimates the one in the SP algorithm.

Proof: By (4), the magnitude of CN message $|T_{m \rightarrow n}|$ of a-min*, is

$$\begin{aligned} |T_{m \rightarrow n}| &= \left| \boxplus_{n' \in \mathcal{N}(m)} L_{n' \rightarrow m} \right| \\ &= \left| \left(\boxplus_{n' \in \mathcal{N}(m) - n} L_{n' \rightarrow m} \right) \boxplus L_{n \rightarrow m} \right| \\ &\leq \min \left(\left| \boxplus_{n' \in \mathcal{N}(m) - n} L_{n' \rightarrow m} \right|, |L_{n \rightarrow m}| \right) \\ &\leq \left| \boxplus_{n' \in \mathcal{N}(m) - n} L_{n' \rightarrow m} \right|, \end{aligned}$$

which is the magnitude of CN message of the SP algorithm, the first inequality follows from (6). ■

IV. GENERALIZED A-MIN* DECODER

In this section a generalized a-min* decoder (ga-min*) in terms of number of messages to CNs is proposed. The intuition is based on the property of the box-plus operator that, if $|L_1| \ll |L_2|$, then $|L_1 \boxplus L_2| \approx |L_1|$. For example, $|1 \boxplus -5| = 0.9843 \approx 1$. Therefore, the smaller magnitude is dominating in the box-plus operator. By finding subsets of incoming messages with smallest magnitudes before the CN processing, we can approximate (3) and (4) in the a-min* to process a smaller number of VN messages.

At a CN m of d_c VN neighbors, let s and s' be two integers where $s, s' \leq d_c$ and $s_{\max} = \max(s, s')$. Find s_{\max} smallest incoming messages in magnitude and denote the index of the VN with i th smallest message by n_i for $i = 1, 2, \dots, s_{\max}$. In the following, write L_{n_i} instead of $L_{n_i \rightarrow m}$ for simplicity. Let $\mathcal{S}_m = \{L_{n_1}, L_{n_2}, \dots, L_{n_s}\}$ and $\mathcal{S}'_m = \{L_{n_2}, L_{n_3}, \dots, L_{n_{s'}}\}$ be the resulting two sets of the VN messages with cardinalities $|\mathcal{S}_m| = s$, $|\mathcal{S}'_m| = s' - 1$. The CN processing for the ga-min* decoder is formalized in Algorithm 2.

Algorithm 2 The CN processing of the ga-min*

1. The magnitude of the critical CN message to VN n_1 is computed by applying the box-plus operator to messages $L_{n_i} \in \mathcal{S}'_m$, with an extrinsic sign that is the product of signs of $L_{n'}, n' \in \mathcal{N}(m) - n_1$.

$$T_{m \rightarrow n_1} = \left(\prod_{n' \in \mathcal{N}(m) - n_1} \text{sign}(L_{n'}) \right) \cdot \left| \boxplus_{L_{n_i} \in \mathcal{S}'_m} L_{n_i} \right|. \quad (7)$$

2. For the rest of VNs in the neighborhood of CN m , the magnitude of non-critical messages to VN $n, n \neq n_1$ is computed by applying the box-plus operator to messages $L_{n_i} \in \mathcal{S}_m$, with an extrinsic sign that is the product of signs of $L_{n'}, n' \in \mathcal{N}(m) - n$.

$$T_{m \rightarrow n} = \left(\prod_{n' \in \mathcal{N}(m) - n} \text{sign}(L_{n'}) \right) \cdot \left| \boxplus_{L_{n_i} \in \mathcal{S}_m} L_{n_i} \right|. \quad (8)$$

Remark 1. When $s = s' - 1 = 1$, the ga-min* reduces to the MS algorithm. When $s = s' = d_c$, the ga-min* becomes the a-min*.

Remark 2. For the a-min*, the non-critical CN message is non-extrinsic, i.e., the computation of message $T_{m \rightarrow n}$, $n = n_2, n_3, \dots, n_{d_c}$, involves the corresponding incoming message L_n . While for the ga-min*, only CN messages to VNs n_2, n_3, \dots, n_s will be non-extrinsic.

Unlike the a-min* algorithm, which computes the same value for the critical message as that in the SP algorithm, ga-min* computes an over-estimated critical message.

Proposition 3. If $s < d_c$, then the CN message to VN n_1 of the ga-min* decoder over-estimates the one in the SP algorithm.

Proof: The proof is similar to the one given in Proposition 2 and by the fact that the index set of VNs with messages $L_{n_i} \in \mathcal{S}'_m$ is a proper subset of $\mathcal{N}(m) - n_1$ if $s' < d_c$. ■

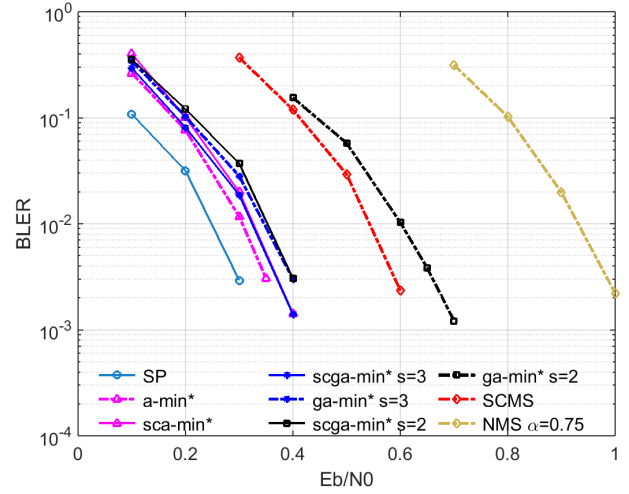
V. SIMULATIONS

A. Performance

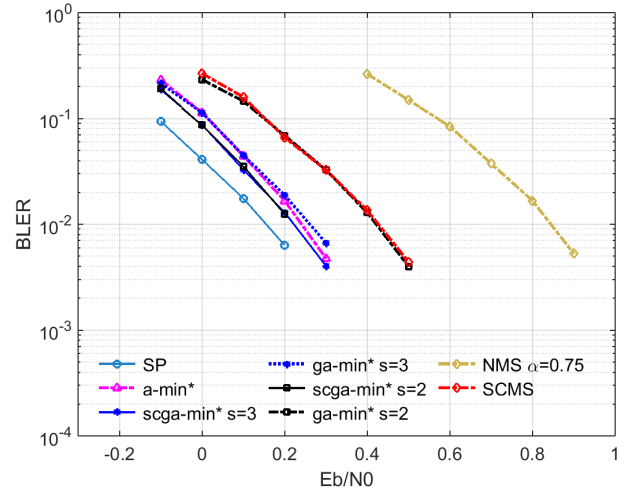
We perform simulations in Fig. 2(a) and 2(b) on the AWGN channel with BPSK modulated transmission. The block error rate (BLER) is simulated down to around 10^{-2} since it is the primary target rate in the NR eMBB scenario [12]. An LDPC code in each BG is selected with the lowest rate. Note that given information length K , the lowest rate, i.e., $1/3$ in BG1 and $1/5$ in BG2, will lead to biggest performance gaps between SP algorithm and other iterative decoding algorithms discussed in the paper. For simplicity, we let $s = s' - 1$ in the (sc)ga-min* and choose $s = 2$ or 3 in the simulations.

With $s = 2$, the ga-min* has 0.3dB and 0.2dB loss to the SP algorithm in these two codes, respectively. With $s = 3$, the gap reduces to 0.1dB to the SP algorithm. The performance gap is less than 0.05dB in the first code and almost overlapped to the a-min* in the second. Meaning that with $s = 3$, the CN processings of both critical and non-critical messages are good approximations of those in the a-min*.

The performance of the sca-min* and the a-min* is fairly close, in which the CN processing is exact in (3) and under-estimated



(a)



(b)

Fig. 2. BLER performance of various decoders (a) $K = 4224$, $R = 1/3$; (b) $K = 960$, $R = 1/5$

in (4) compared to that of the SP algorithm. Under this condition, SC technique has no prominent effect in improving the performance. A similar condition can account for the close performance between the scga-min* and the ga-min* with $s = 3$, where the CN message is close enough to that of the a-min*, as mentioned above.

However, with $s = 2$ the scga-min* outperforms the ga-min* by 0.2dB and also slightly better than the performance of the a-min*. This is showing that the SC technique can achieve a performance boost when the critical message in the CN of the ga-min* is over-estimated.

An even bigger performance gain, i.e., 0.4dB, can be observed between the SCMS and the NMS, where the scaling factor $\alpha = 0.75$ is optimized heuristically for this code. In this case, the SC technique is also powerful in improving the performance. Note that performance of the MS is further degraded from that of the NMS by a few tenth of dB and is not shown in Fig. 2(a) and 2(b).

B. Robustness

Let LLRs from the channel fed into the decoder be scaled by the constant η . If $\eta > 1$ (< 1), we refer to it as LLR over (under)-estimation. Fig. 3 shows the performance under LLR mismatch. The (sc)ga-min* shows superiority in performance compared to other iterative algorithms for both situations. Also note that the SP and a-min* decoder are very sensitive to both SNR under-estimation ($\eta = 0.5$) in Fig. 3(a) and over-estimation ($\eta = 2$) in Fig. 3(b). There is at least 1dB loss to the performance in the absence of LLR mismatch in Fig. 2(a). The SCMS and scga-min* with $s = 2$ are robust in both LLR under-estimate and over-estimate. Specifically, the performance of the SCMS is similar in both cases, whereas only 0.2dB and 0.1dB degradation for LLR under-estimate and over-estimate, respectively, is in the scga-min* with $s = 2$. For LLR under-estimation, the ga-min* even outperforms the one in the absence of LLR mismatch by 0.1dB, but it degrades drastically for LLR over-estimation. We conclude that when SC is applied to the ga-min* with $s = 2$, the robustness of the performance can be enhanced greatly against LLR mismatch.

C. Complexity Comparison

In the ga-min*, the box-plus operator is used only s times per CN, compared to $d_c - 1$ times in the a-min*. The saving in the box-plus operator is at the cost of finding out $s + 1$ smallest incoming messages in magnitude, which is relatively inexpensive to implement in hardware for the case of $s = 2$ or 3. A hardware implementation of finding two smallest values in a set is presented in [9]. Table I provides a complexity comparison with respect to the box-plus operator savings for various code rates of NR LDPC codes. In Table I, the highest rate has the highest average row weight d_c , due to the high CN degree in the core graph, resulting in greatest savings, namely 86% for $s = 2$ and 80% for $s = 3$ in BG1, and 69% for $s = 2$ and 53% for $s = 3$ in BG2. As the rate goes lower more CNs in the extension graph are included, which lowers d_c , and the complexity saving is 66% for $s = 2$ and 49% for $s = 3$ in BG1, and 46% for $s = 2$ and 19% for $s = 3$ in BG2. We also notice there is slight increase in iteration numbers when SC is applied to the ga-min*, but the extra numbers of iterations can be well accommodated by the complexity savings in ga-min* itself.

TABLE I
Box-plus savings of the ga-min* for various code rates compared to the a-min*

R	BG	$d_c - 1$	savings ($s = 2$)	savings ($s = 3$)
8/9	1	14.80	86%	80%
11/15	1	11.20	82%	73%
1/2	1	7.75	74%	61%
1/3	1	5.87	66%	49%
2/3	2	6.43	69%	53%
1/2	2	5.42	63%	45%
1/3	2	4.50	55%	33%
1/5	2	3.69	46%	19%

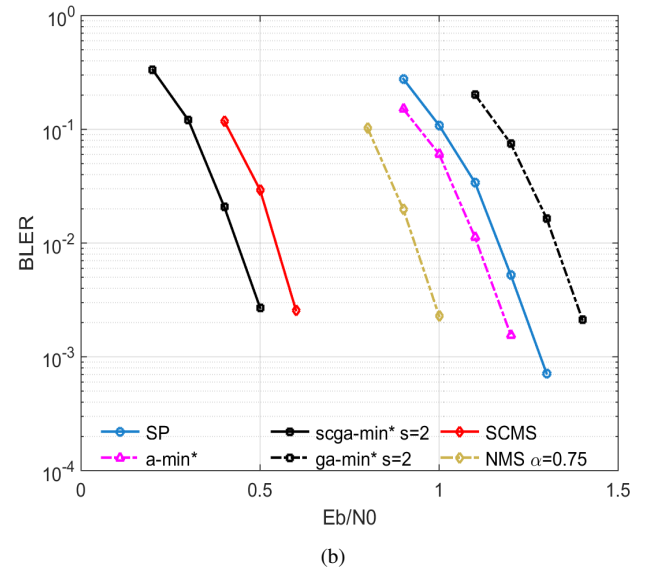
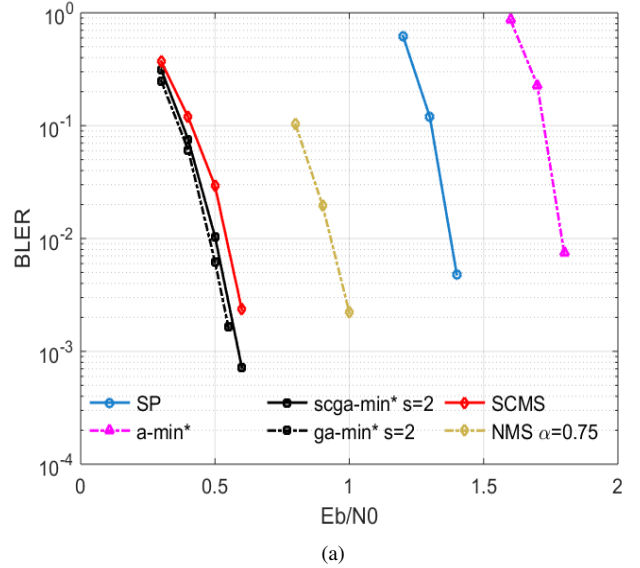


Fig. 3. BLER performance of various decoders under LLR mismatch for the code of $K = 4224$, $R = 1/3$ (a) $\eta = 1/2$; (b) $\eta = 2$

VI. CONCLUSION

In this work we propose a new iterative decoding algorithm, ga-min*. We identify conditions for which the SC technique can be a good performance boost when is applied to the ga-min*. Simulations on 5G LDPC codes for various decoding algorithms are conducted. While the SP has the best performance, it is not robust and degrades drastically under LLR mismatch. The NMS is robust with low complexity, but gives sub-optimal performance for low rate codes. The performance of the scga-min* with $s = 2$ is close to the SP algorithm and comparable to the a-min* but with much complexity savings, and it is robust against LLR mismatch.

REFERENCES

- [1] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inf. Theory*, vol. 8, pp. 21-28, Jan. 1962.
- [2] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementation of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Global Telecommun. Conf.*, San Antonio, TX, Nov. 2001, pp. 1036-1036E vol.2.
- [3] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406-414, Mar. 2002.
- [4] C. Jones, E. Valles, M. Smith, and J. Villasenor, "Approximate-Min* constraint node updating for LDPC code decoding," in *Proc. IEEE Mil. Commun. Conf.*, Boston, MA, Oct. 2004, vol. 1, pp. 157-162.
- [5] V. Savin, "Self-corrected min-sum decoding of LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Toronto, ON, Jul. 2008, pp. 146-150.
- [6] J. Andrade, G. Falcao, V. Silva, J.P. Baretto, N. Goncalves, V.Savin, "Near- LSPA Performance at MSA Complexity," in *Proc. IEEE Int. Conf. Commun.*, Budapest, Hungary, Jun. 2013, pp. 3281-3285.
- [7] C. K. Ngassa, V. Savin, and D. Declercq, "Min-sum-based decoders running on noisy hardware," in *Proc. IEEE Global Telecommun. Conf.*, Atlanta, GA, Dec. 2013, pp. 1879-1884.
- [8] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, pp. 429-445, Mar. 1996.
- [9] 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, Reg. Papers*, vol. 55, no. 11, pp. 3430-3437, Dec. 2008.
- [10] 3GPP TSG RAN WG1 NR Ad-Hoc #2, R1-1711545, "Summary of [89-24] Email Discussion on LDPC Code Base Graph #1 for NR," Nokia, Alcatel-Lucent Shanghai Bell, Qingdao, China, 27-30 Jun. 2017.
- [11] 3GPP TSG RAN WG1 NR Ad-Hoc #2, R1-1711546, "Summary of [89-25] Email Discussion on LDPC code Base Graph #2 for NR," Nokia, Alcatel-Lucent Shanghai Bell, Qingdao, China, 27-30 Jun. 2017.
- [12] 3GPP TSG RAN WG1 meeting 91, R1-1720759, "Channel coding for URLLC", Huawei, HiSilicon, Reno, USA, Dec. 2017.
- [13] 3GPP TSG-RAN WG1 meeting #86bis, R1-1610140, "LDPC decoding with adjusted min-sum," Qualcomm Incorporated, Lisbon, Portugal, Oct. 2016.