The Box-Minus Operator and its Application to Low-Complexity Belief Propagation Decoding

Thorsten Clevorn and Peter Vary
Institute of Communication Systems and Data Processing (ind), RWTH Aachen University, Germany
{clevorn,vary}@ind.rwth-aachen.de

***Abstract*— For iterative decoding of Turbo codes and low-density parity check (LDPC) codes quite often log-likelihood ratios called L-values are used. A major role in the decoding algorithms plays extrinsic information which is obtained as L-value of a modulo 2 combination of several bits. This leads in the L-domain to the so-called box-plus operation. In this paper we introduce a complementary box-minus operation. Based on it paper a low-complexity but powerful belief propagation algorithm for decoding of LDPC codes is described. This lookup-sum algorithm uses lookup-tables for correction functions to efficiently approximate box-plus and box-minus operations and its capabilities are demonstrated by complexity comparisons and simulation results.**

I. Introduction

With the discovery of Turbo codes [1],[2],[3] and the (re)discovery of low-density parity check (LDPC) codes [4],[5],[6] channel coding close to the Shannon limit becomes possible with moderate computational complexity. To further simplify the computations *log-likelihood ratios* (LLR), i.e., L-values [3], are frequently used, e.g., in decoding of Turbo processes [1],[2],[3],[7] and belief propagation decoding of LDPC codes [5],[8],[9],[10]. Using L-values the *multiplicative* decoding algorithms based on probabilities are transformed into usually less complex *additive* algorithms in the L-value domain.

We present the new arithmetic operation *box-minus* $\boxminus$ for L-values, which is the complementary to the *box-plus* operation $\boxplus$ [3]. For each operation two approximations are presented, the better ones using a lookup-table for a correction function. *Box-plus*, *box-minus* and their approximations can be, e.g., used for low-complexity belief propagation decoding of LDPC codes by the *lookup-sum* algorithm [10]. Simulation results show a significant reduction of the computational complexity, approaching the complexity of the *min-sum* algorithm, with none or only a marginal loss in performance with respect to the non-approximated *sum-product* algorithm.

Belief propagation decoding is the most common decoding technique for LDPC codes. Very recently LDPC codes have been standardized as channel codes for new wireless communication systems, e.g., for the second generation of digital video broadcasting via satellite (DVB-S2) [11],[12] and as option in the IEEE 802.16 standard for wireless metropolitan area networks (WirelessMAN) [13]. This indicates the importance of LDPC codes in future communication systems and the resulting demand for powerful but low-complex decoding algorithms.

II. The Box-Plus and the Box-Minus Operator

A. L-values and Soft-Bits

An L-value L_i is the natural logarithm of the ratio of the probabilities for the two realizations of a binary random variable x_i, possibly conditioned on other variables [3]:

$$L_i \triangleq L(x_i|\dots) = \log \frac{P(x_i=+1|\dots)}{P(x_i=-1|\dots)} \quad . \tag{1}$$

The sign of an L-value $\operatorname{sgn}(L_i)$ is the hard decision for the variable while the magnitude $|L_i|$ indicates the reliability of the decision.

In the context of L-values also soft-bits [14],[15] $\breve{x}_i$ can be defined:

$$\breve{x}_i = E\{x_i\} = P(x_i{=}+1) - P(x_i{=}-1) = \tanh(L_i/2) \tag{2}$$

$$\text{and} \quad P(x_i{=}\pm 1) = (1 + x_i \breve{x}_i)/2 \quad . \tag{3}$$

B. The Box-Plus Operator $\boxplus$ and its Approximations $\tilde{\boxplus}$, $\overset{\approx}{\boxplus}$

Besides the regular addition in $\mathbb{R}$, using L-values often requires the second arithmetic operation *box-plus* $\boxplus$ [3]:

$$\begin{aligned} L_1 \boxplus L_2 &= \log \frac{1 + e^{L_1} e^{L_2}}{e^{L_1} + e^{L_2}} & (4)\\ &= 2 \operatorname{atanh}\big(\tanh(L_1/2) \cdot \tanh(L_2/2)\big) \quad , & (5) \end{aligned}$$

with $L_1 \boxplus \infty = L_1$, $L_1 \boxplus {-\infty} = -L_1$, and $L_1 \boxplus 0 = 0$. In [3] also the well known approximation, denoted by $\overset{\approx}{\boxplus}$, is presented:

$$L_1 \overset{\approx}{\boxplus} L_2 = \operatorname{sgn}(L_1) \operatorname{sgn}(L_2) \min(|L_1|, |L_2|) \tag{6}$$

In [7] it was shown that the correction function $f_+(x) = \log(1 + e^{-x})$ of the Jacobian logarithm can be efficiently implemented by a lookup-table $\tilde{f}_+$. Using $\tilde{f}_+$ (4) can be approximated [10] by

$$\begin{aligned} L_1 \tilde{\boxplus} L_2 = &\operatorname{sgn}(L_1) \operatorname{sgn}(L_2) \min(|L_1|, |L_2|) \\ &+ \tilde{f}_+(|L_1 + L_2|) - \tilde{f}_+(|L_1 - L_2|) \quad , \end{aligned} \tag{7}$$

which improves the approximation in (6).

C. The Box-Minus Operator $\boxminus$ and its Approximations $\tilde{\boxminus}$, $\overset{\approx}{\boxminus}$

As complementary operation to *box-plus* $\boxplus$ we introduce the *box-minus* operation $\boxminus$

$$\begin{aligned} L_3 \boxminus L_4 &= \log \frac{1 - e^{L_3} e^{L_4}}{e^{L_3} - e^{L_4}} & (8)\\ &= 2 \operatorname{atanh}\left(\frac{\tanh(L_3/2)}{\tanh(L_4/2)}\right) , & (9) \end{aligned}$$

0-7803-8887-9/05/$20.00 (c)2005 IEEE

with $L_3 \boxminus \infty = L_3$, $L_3 \boxminus -\infty = -L_3$, and $0 \boxminus L_4 = 0$. Thus, ∞ is the identity element and 0 the "infinity" element for $\boxminus$ as well as for $\boxplus$. Note that $\boxminus$ is only defined if

$$|L_3| < |L_4| \quad . \tag{10}$$

This is somewhat similar to the regular subtraction in $\mathbb{R}^+$ where the minuend has to be larger than the subtrahend and it can be interpreted such that with $\boxminus$ only a more reliable L-value can be subtracted from a less reliable one. However, if L_4 was part of a, e.g., recursive *box-plus* summation to obtain L_3, the condition of (10) is always fulfilled. Further note that, e.g., in consequence of (10), $\boxminus$ is not commutative, i.e.,

$$L_3 \boxminus L_4 \neq L_4 \boxminus L_3 \quad . \tag{11}$$

There exists an approximation $\overset{\approx}{\boxminus}$ for $\boxminus$ similar to (6)

$$L_3 \overset{\approx}{\boxminus} L_4 = \mathrm{sgn}(L_3)\,\mathrm{sgn}(L_4)\min(|L_3|, |L_4|) \tag{12}$$
$$= \mathrm{sgn}(L_4) L_3 \quad . \tag{13}$$

(13) results from (12) due to (10).

An approximation $\tilde{\boxminus}$ comparable to (7) using a lookup-table $\tilde{f}_-$ for a correction function is also available for $\boxminus$:

$$L_3 \overset{\sim}{\boxminus} L_4 = \mathrm{sgn}(L_4) L_3 + \tilde{f}_-(|L_3 + L_4|) - \tilde{f}_-(|L_3 - L_4|). \tag{14}$$

The lookup-table $\tilde{f}_-$ represents the correction function $f_-(x) = \log(1 - e^{-x})$.

D. *Combination of Box-Plus $\boxplus$ and Box-Minus $\boxminus$*

When combining the $\boxplus$ and $\boxminus$ operation in a single equation the order may make the result defined or undefined as a consequence of (10), e.g., $(L_5 \boxplus L_6) \boxminus L_6 = L_5$ in any case while $(L_5 \boxminus L_6) \boxplus L_6 = L_5$ is only possible for $|L_5| < |L_6|$.

Using the soft-bits $\breve{x}_i = \tanh(L_i/2)$ in (5) and (9), equations with multiple $\boxplus$ and $\boxminus$ can be simplified or rearranged, e.g.,

$$L_1 \ldots \boxplus L_i \ldots \boxplus L_{I_1} \boxminus L_{I_1+1} \ldots \boxminus L_{I_2}$$
$$= 2\,\mathrm{atanh}\left((\prod_{i=1}^{I_1} \breve{x}_i)/(\prod_{i=I_1+1}^{I_2} \breve{x}_i) \right) \tag{15}$$
$$= \left(\sum_{i=1}^{I_1} \boxplus L_i \right) \boxminus \left(\sum_{i=I_1+1}^{I_2} \boxplus L_i \right) \quad , \tag{16}$$

if the condition (10), i.e., $|\sum_{1}^{I_1} \boxplus L_i| < |\sum_{I_1+1}^{I_2} \boxplus L_i|$, is fulfilled.

III. Application to Belief Propagation Decoding

For belief propagation (BP) [16] decoding we consider an LDPC code with a $P \times N$ parity-check matrix $\underline{B}$ containing elements $B_{pn} \in \{0, 1\}$, $p = 1, \ldots P$ and $n = 1, \ldots N$. The rows of $\underline{B}$ are the P parity-check vectors $\underline{b}_p$. For each column n of $\underline{B}$ there exists a sub-matrix $\underline{\mathcal{B}}_n$ of parity-check vectors $\underline{b}_p$ with $B_{pn} = 1$. Thus, $\underline{\mathcal{B}}_n$ comprises all rows of $\underline{B}$ with a one in column n. No other column of $\underline{\mathcal{B}}_n$ has more than a single non-zero element. For example if we use

$$\underline{B} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

we get

$$\underline{\mathcal{B}}_2 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

We assume the transmission of the bipolar code word $\underline{y} = (y_1, \ldots y_N)$, $y_n \in \{+1, -1\}$, and the reception of $\underline{z} = (z_1, \ldots z_N)$, $z_n \in \mathbb{R}$, which is distorted by AWGN of zero mean and variance $N_0/2$.

A. *Belief Propagation Decoding with L-Values*

When using L-values for BP decoding [8], the elements z_n of the received $\underline{z}$ are converted to L-values $r_n^{(0)} := (4/N_0) \cdot z_n$. Next, a $P \times N$ matrix $\underline{Z}$ is initialized according to $\underline{B}$ as $Z_{pn}^{(0)} = r_n^{(0)}$ if $B_{pn} = 1$, and $Z_{pn}^{(0)} = 0$ otherwise.

The first step of each iteration i is the computation of a matrix $\underline{E}^{(i)}$ with $E_{pn}^{(i)} = 0$ for $B_{pn} = 0$, and

$$E_{pn}^{(i)} = 2\,\mathrm{atanh}\left(\prod_{j \in \mathrm{supp}(\underline{b}_p) \backslash n} \tanh(Z_{pj}^{(i)}/2) \right) = \sum_{j \in \mathrm{supp}(\underline{b}_p) \backslash n} \boxplus Z_{pj}^{(i)} \tag{17}$$

for $B_{pn} = 1$, with $\mathrm{supp}(\underline{b}_p) := \{j | B_{pj} = 1\}$.

In the next step of each iteration the obtained *extrinsic* information $E_{pn}^{(i)}$ is summed for each n

$$e_n^{(i)} = \sum_{p,\, \underline{b}_p \in \underline{\mathcal{B}}_n} E_{pn}^{(i)} \quad , \tag{18}$$

and then added to $r_n^{(0)}$

$$r_n^{(i+1)} = r_n^{(0)} + e_n^{(i)} \quad . \tag{19}$$

Afterwards a component-wise hard decision is made $\underline{\hat{y}}^{(i+1)} = \mathrm{sgn}(\underline{r}^{(i+1)})$, $\underline{\hat{y}}^{(i+1)} \in \mathrm{GF}(2)$. If the maximum number of iterations is reached or if $\underline{\hat{y}}^{(i+1)}$ fulfills the terminating condition $(\underline{B} \cdot \underline{\hat{y}}^{(i+1)}) \bmod 2 = \underline{0}$, the iterative process is stopped and $\underline{\hat{y}}^{(i+1)}$ serves as the decoding output.

Otherwise, if the terminating condition is not fulfilled, the elements Z_{pn} with $B_{pn} = 1$ are updated in the last step by

$$Z_{pn}^{(i+1)} = Z_{pn}^{(0)} + \sum_{l,\, \underline{b}_l \in \underline{\mathcal{B}}_n \backslash \underline{b}_p} E_{ln}^{(i)} \quad , \tag{20}$$

and after increasing the iteration index, $i = i + 1$, the next iteration starts with (17). In (20) $E_{pn}^{(i)}$ is excluded from the summation for the new $Z_{pn}^{(i+1)}$ to mitigate error propagation.

B. Efficient Decomposition

In (17) and (20) always just one element is excluded from the summation. Thus, it is obviously more efficient, at least for rows or columns with a not too small weight, to first compute the result for the complete row or column and afterwards extract the particular element [10]. Since (20) contains a regular summation the decomposition is straightforward. Actually, the sum of the complete column $e_n^{(i)}$ is already given by (18), and consequently (20) is modified to

$$Z_{pn}^{(i+1)} = Z_{pn}^{(0)} + e_n^{(i)} - E_{pn}^{(i)} \quad . \tag{21}$$

The decomposition of the *box-plus* summation in (17) requires the complementary operation to *box-plus*, i.e., the *box-minus* operation $\boxminus$ introduced in Section II-C. Based on (16) we can write for a *box-plus* summation excluding a single element

$$\sum_{k=1,k\neq j}^{K}\!\boxplus L_k = \left(\sum_{k=1}^{K}\!\boxplus L_k\right) \boxminus L_j \tag{22}$$

and using this for the decomposition of (17) yields

$$E_{pn}^{(i)} = \sum_{j\in\mathrm{supp}(\underline{b}_p)\setminus n}\!\boxplus Z_{pj}^{(i)} = \left(\sum_{j\in\mathrm{supp}(\underline{b}_p)}\!\boxplus Z_{pj}^{(i)}\right) \boxminus Z_{pj}^{(i)} \quad . \tag{23}$$

C. Computational Complexity Comparison

The difference in complexity between the different compared algorithms consists mainly in the computation of (23). Table I lists the required operations for a row of weight J. The last line gives the total operations, weighted according to the weights of the ETSI basic operators [17]. Most operations have a weight of $w = 1$ (assuming 16 bit accuracy). The division has a weight of $w_{\text{div}} = 18$. The weights of "exp" and "log", $w_{\exp}$ resp. $w_{\log}$, depend on the complexity of their implementation, e.g., series expansion, since "exp" and "log" are not basic operators themselves. For simplicity we set $w_{\exp}, w_{\log} \geq 1$, although usually $w_{\exp}, w_{\log} \gg 1$ will be the case.

The algorithms compared in Table I are the *min-sum* ($\overset{\approx}{\boxplus}, \overset{\approx}{\boxminus}$) algorithm, the *lookup-sum* ($\tilde{\boxplus}, \tilde{\boxminus}$) algorithm [10] using efficient lookup-tables $\tilde{f}_{\pm}$, the *log-likelihood ratio sum-product algorithm* (LLR-SPA) with correction [9], which uses a forward-backward algorithm without decomposition, and the exact *sum-product* ($\boxplus, \boxminus$) algorithm with $\tanh(x/2) = (e^x - 1)/(e^x + 1)$ and $2\,\mathrm{atanh}(x) = \log((1+x)/(1-x))$.

TABLE I

REQUIRED OPERATIONS FOR A ROW OF WEIGHT J

	$\overset{\approx}{\boxplus}, \overset{\approx}{\boxminus}$	$\tilde{\boxplus}, \tilde{\boxminus}$	LLR-SPA	$\boxplus, \boxminus$
$\mathrm{sgn}(x)\,\mathrm{sgn}(y)$	$J-1$	$J-1$	$3J-2$	-
$\mathrm{sgn}(x)y$	$J+2$	$2J-1$	$3J-2$	-
$\min(\lvert x\rvert, \lvert y\rvert)$	$J-1$	$J-1$	$3J-2$	-
$\tilde{f}_{\pm}(\lvert x\rvert)$	-	$4J-2$	$6J-4$	-
$+$ and $-$	-	$8J-4$	$12J-8$	$4J$
$*$	-	-	-	$J-1$
$/$	-	-	-	$3J$
$\exp(x), \log(x)$	-	-	-	$2J$
weighted Op.	$3J$	$16J-9$	$27J-18$	$\geq 61J-1$

D. Simulation Results

Fig. 1 depicts the simulation results for the (273,191) DSC (difference set cyclic) code [8],[18] with a row weight of $J = 17$. AWGN serves as channel distortion and the maximum number of iterations is 50. The lookup-tables $\tilde{f}_{\pm}$ of the *lookup-sum* algorithm have T entries, equally distributed for the input $|x|$ between 0 and $x_{\max}$. For $|x| > x_{\max}$ we set $\tilde{f}_{\pm}(|x|) = 0$. These values of the lookup-tables can be optimzed for a specific code. But for simplicity we used identical sized input bins for each table, with the correction factor of the center of a bin as output value.

The bit-error rate (BER) and frame-error rate (FER) of *sum-product* and *lookup-sum* with $T = 16$ coincide. *Lookup-sum* with $T = 1$ shows a negligible degradation below $0.1\,\mathrm{dB}$, while the penalty of *min-sum* of above $1\,\mathrm{dB}$. The center plot depicts the iterations executed in average. For the bottom plot these average iterations are combined with the weighted operations (wOp) in Table I ($w_{\exp} = w_{\log} = 1$). For the relevant E_b/N_0 (BER $10^{-3} \ldots 10^{-5}$) the complexity of *lookup-sum* approaches the one of *min-sum*, with *lookup-sum* significantly outperforming *min-sum* in terms of BER.

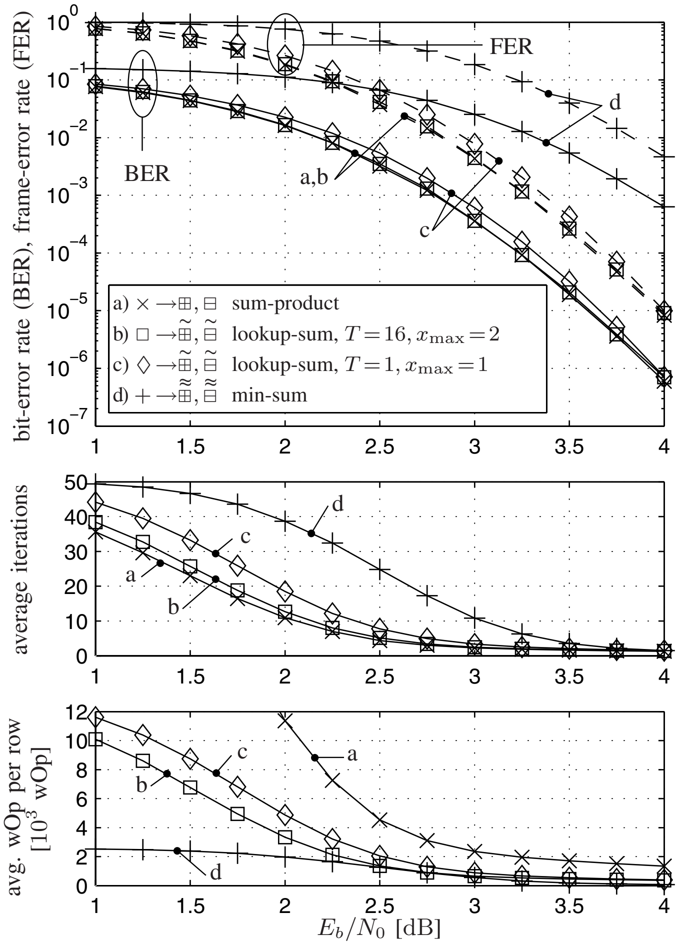

Fig. 1. Simulation results for the (273,191) DSC code

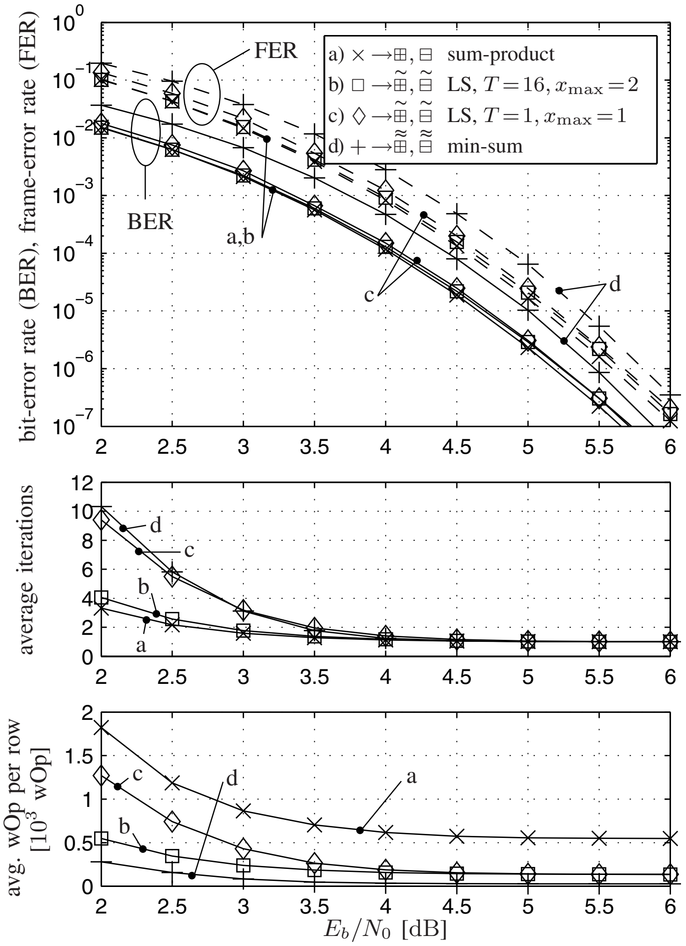

Fig. 2. Simulation results for the (73,45) DSC code

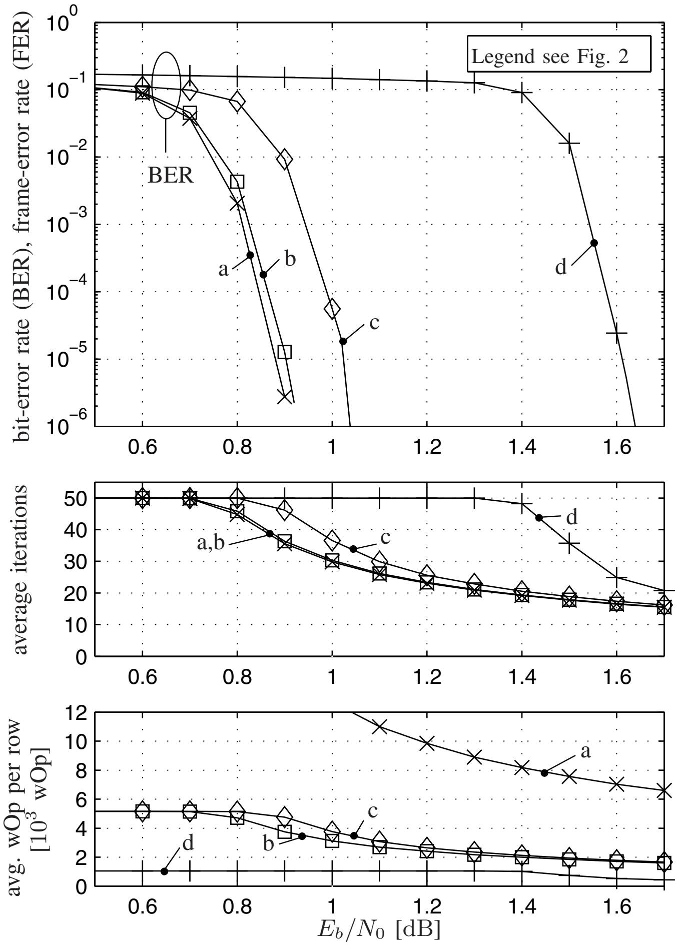

Fig. 3. Simulation results for the $r = 1/2$ LDPC code of DVB-S2 [11]

In Fig. 2 similar simulation results are depicted for the (73,45) DSC code with $J = 9$. Again the BER and FER curves *sum-product* and *lookup-sum* (LS) with $T=16$ mostly coincide and a barely noticeable detriment of *lookup-sum* with $T=1$ can be observed. The penalty of the *min-sum* algorithm decreases to $\approx 0.7\,\mathrm{dB}$ due to the smaller row weight. Only the information of 7 out of $J-1=8$ other entries is omitted by the *min*-operation without correction, compared to 15 out of $J-1=16$ entries for the (273,191) DSC code in Fig. 1.

For the simulation results presented in Fig. 3 we used a LDPC code defined in the DVB-S2 standard [11]. We choose the (64800,32400)-LDPC code with code rate $r=1/2$. This is an irregular LDPC code [19] with column weights of 2, 3, and 8 for the parity-check matrix [12]. The row weight is $J=7$, except for a single row with weight 6. As visible in Fig. 3 this long and irregular LDPC code approaches the channel capacity $(E_b/N_0)_{\min} \approx 0.2\,\mathrm{dB}$ for a $r=1/2$ code and BPSK transmission by $\approx 0.7\,\mathrm{dB}$. The performance loss of *lookup-sum* with a single correction factor, i.e., $T=1$, is about $0.1\,\mathrm{dB}$ and *lookup-sum* with $T=16$ still almost coincides with the *sum-product* algorithm. The penalty of the *min-sum* algorithm is again $\approx 0.7\,\mathrm{dB}$.

IV. Conclusion

We introduced the novel arithmetic operation *box-minus* $\boxminus$ for L-values as complementary operation to the well-known *box-plus* operation $\boxplus$. Additionally, two approximations are presented for both operations. Using the operators and their approximations the low-complexity belief propagation algorithm *lookup-sum* is described, and its capabilities in terms of a significantly reduced computational complexity with almost no loss in performance are analyzed by simulations. It was shown that there is no significant difference in performance compared to standard belief propagation decoding even if the lookup-table size is very small. Already for a correction factor, i.e., a lookup-table with a single entry, the performance of the complex, non-approximated belief propagation algorithm can be approached by less than $0.1\,\mathrm{dB}$ in E_b/N_0.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding," *IEEE International Conference on Communications (ICC)*, Geneva, Switzerland, May 1993.

[2] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," *IEEE Transactions on Communications*, pp. 1261–1271, Oct. 1996.

[3] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Convolutional Codes," *IEEE Transactions on Information Theory*, pp. 429–445, Mar. 1996.

[4] D. J. C. MacKay and R. M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes," *Electronics Letters*, pp. 1645–1646, Aug. 1996.

[5] D. J. C. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Transactions on Information Theory*, pp. 399–431, Mar. 1999.

[6] R. G. Gallager, "Low-Density Parity-Check Codes," *IRE Transaction on Information Theory*, pp. 21–28, Jan. 1962.

[7] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and Sub-Optimal Maximum a Posteriori Algorithms Suitable for Turbo Decoding," *European Transactions Telecommunications*, pp. 119–125, Mar. 1997.

[8] R. Lucas, M. P. Fossorier, Y. Kou, and S. Lin, "Iterative Decoding of One-Step Majority Logic Decodable Codes Based on Belief Propagation," *IEEE Transactions on Communications*, pp. 931–937, June 2000.

[9] E. Eleftheriou, T. Mittelholzer, and A. Dholakia, "Reduced-complexity decoding algortihm for low-density parity-check codes," *Electronics Letters*, pp. 102–104, Jan. 2001.

[10] T. Clevorn and P. Vary, "Low-Complexity Belief Propagation by Approximations with Lookup-Tables," *5th Internantional ITG Conference on Source and Channel Coding (SCC)*, Erlangen, Germany, Jan. 2004.

[11] ETSI, "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications," June 2004, Draft EN 302 307 V1.1.1.

[12] M. Eroz, F.-W. Sun, and L.-N. Lee, "DVB-S2 low density parity check codes with near Shannon limit performance," *International Journal of Satellite Communications and Networking*, pp. 269–279, May 2004.

[13] IEEE, "Standard for Local and metropolitan area networks; Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems; Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands," Sept. 2004, IEEE P802.16e/D5 (Draft).

[14] P. Vary and T. Fingscheidt, "From Soft Decision Channel Decoding to Soft Decision Speech Decoding," *2. ITG-Fachtagung "Codierung für Quelle, Kanal und Übertragung"*, Aachen, Germany, Mar. 1998.

[15] J. Hagenauer and T. Stockhammer, "Channel Coding and Transmission Aspects for Wireless Multimedia," *Proceedings of the IEEE*, pp. 1764–1777, Oct. 1999.

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

[17] ETSI SMG11, "AMR permanent document (AMR-9): Complexity and delay assessment," ETSI Tdoc SMG11 136/98.

[18] E. J. Weldon, "Difference-set cyclic codes," *The Bell Systems Techical Journal*, pp. 1045–1055, Sept. 1966.

[19] T. J. Richardson, M. A. Shokrallahi, and R. L. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, pp. 619–637, Feb. 2001.