



The Best, The Requested, and The Default Non-Binary LDPC Decoding Algorithm

Joseph Jabour, Cédric Marchand, E. Boutillon

► To cite this version:

Joseph Jabour, Cédric Marchand, E. Boutillon. The Best, The Requested, and The Default Non-Binary LDPC Decoding Algorithm. IEEE Wireless Communications and Networking Conference, IEEE, Mar 2023, Glasgow, United Kingdom. hal-04048548

HAL Id: hal-04048548

<https://hal.science/hal-04048548v1>

Submitted on 28 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Best, The Requested, and The Default Non-Binary LDPC Decoding Algorithm

JOSEPH JABOUR^{1,2}, CÉDRIC MARCHAND¹, EMMANUEL BOUTILLON¹

¹Laboratoire Lab-STICC, UMR CNRS 6285, Université de Bretagne Sud, Lorient, France

²Lebanese International University, Beirut, Lebanon.

{joseph.jabour, cedric.marchand, emmanuel.boutillon} @univ-ubs.fr

Abstract—This paper presents a different vision of the Non-Binary LDPC decoding algorithm. So far, state-of-the-art decoders are mainly focused on the reduction of the check node processing complexity, with a potential side effect on the size of the messages that are exchanged in the iterative decoder. However, it is possible to take a different approach and consider the reduction of the message size as the main objective. The paper presents “the Best, the Requested and the Default” algorithm that shrinks dramatically the communication load between the variable nodes and check nodes.

Index Terms—NB-LDPC, Extended Min-Sum, Trellis-EMS, (Best, Requested, and Default).

I. INTRODUCTION

Non-Binary Low-Density Parity Check (NB-LDPC) codes [1] are an extension of the Binary LDPC codes that were proposed by R. Gallager in 1963 [2], defined by a set of parity check equations on $\text{GF}(q = 2^p)$ with $p > 1$. NB-LDPC codes outperform LDPC codes when short frames are used since they have higher coding gain and lower error floor [1]. However, NB-LDPC is significantly more complex in decoding than LDPC which limits its hardware implementation for high field order. To reduce the complexity of the NB-LDPC, different approaches and schemes were proposed and implemented.

The Extended Min-Sum (EMS) introduced in [3] truncates the message vector between the Variable Nodes (VNs) and the Check Nodes (CNs) from q down to n_m where $n_m \ll q$ to reduce the computational load in the CN unit and hence reduces the hardware complexity. In the Forward-Backward (FWBW) implementation of EMS [4], the truncation is extended to all messages, which allows reducing the memory allocations from q to n_m . The FWBW architecture has a low level of parallelism due to the serial architecture and data dependencies of the elementary check nodes. Another EMS-based approach is the Syndrome-Based (SYN) decoder [5], SYN decoder helps in achieving a high level of parallelism and higher throughput in the CN unit. In [6], a hybrid approach is proposed that combines both the FWBW and the SYN architectures. The hybrid architecture achieved the lowest-reported number of exchanged messages using the EMS algorithm (6 candidates as input to CN and 20 candidates as output).

Trellis Extended Min-Sum (T-EMS) decoders [7], [8] are based on the trellis representation of the input messages which allows high parallelism between the CN and VN. Recently, contributions were implemented to reduce the memory require-

ments and the size of the messages exchanged. In [9], *TS-TEC-TEMS* is a threshold-based truncation decoding scheme where two thresholds are used to truncate the incoming messages of the CN M_{V2C} and the outgoing messages from the CN to the VN M_{C2V} . In [10], the implemented decoder truncates the M_{V2C} vector to $q/2$ before proceeding with the TEMS decoding scheme. In [11], the authors propose to replace d_c point-to-point messages of size q (where d_c is number of connected VNs for a given CN) by a $5 \times q$ broadcasting message from one source to d_c destinations.

In this paper, we postulate that reducing the message size of the message decoder is not a problem as long as the correct (transmitted) symbol belongs to the message and thus, participates in the convergence of the code. From this idea, a novel decoding algorithm called “The Best, the Request, and the Default” (BRD) is proposed. The BRD algorithm defines the CN to VN M_{C2V} message as the union of three sets: $M_{C2V} = M_{C2V}^B \cup M_{C2V}^R \cup M_{C2V}^D$. The proposed scheme allows reducing the size of the exchanged messages without major performance loss. Decreasing the message size allows increasing the throughput in serial architectures and reduces the routing congestion in highly parallel architectures. The paper is divided into the following sections, Section II briefly introduces the NB-LDPC structure and decoding process. Section III introduces the proposed BRD Algorithm, Section IV presents the statistical analysis, simulation results, and the complexity analysis. Lastly, the concluded points are summarized in Section V.

II. BACKGROUND

NB-LDPC code designed on $\text{GF}(q)$ has a block length of N symbols and an information length of K symbols on $\text{GF}(q)$ with a coding rate $r = K/N$. The connections between the CNs and the VNs are specified by the parity check matrix \mathbf{H} which consists of $N - K$ rows and N columns representing the total number of CNs and VNs respectively, where its element in row i and column j is denoted as $h_{i,j}$.

In Fig. 1(a), the connection between a VN and a CN is illustrated. Let $\mathbf{x} = (x_1, x_2, \dots, x_N)$ be a codeword verifying $\mathbf{H}\mathbf{x} = 0$ transmitted through a noisy channel. For each transmitted GF symbol x_j , the receiver computes I_j , the Log-Likelihood Ratio (LLR) intrinsic vector defined as $I_j(a) = \log(P(x_j = \hat{a})/P(x_j = a))$, $a \in \text{GF}(q)$ with $\hat{a} = \arg \max\{P(x_j = a)\}$. Note that by construction,

$I_j(\hat{a}) = 0$ and $I_j(a) \geq 0, a \in \text{GF}(q)$ [12]. Let $M_{V_j2C_i}$ be the message sent from VN_j to CN_i , the vector is permuted by $h_{i,j}$ before being processed by the CN, and denoted as $M_{V_j2C_i}^P$ with $M_{V_j2C_i}^P(a) = M_{V_j2C_i}(h_{i,j} \cdot a), a \in \text{GF}(q)$, and $h_{i,j} = \mathbf{H}(i, j)$. The CN generates the message $M_{C_i2V_j}^P$ that is inversely permuted and inputted to the VN as $M_{C_i2V_j}$. Lastly, the notations $\mathcal{N}(i)$ and $\mathcal{M}(j)$ represent the indices of the VN/CN connected to the $i^{\text{th}}/j^{\text{th}}$ CN/VN respectively.

The horizontal layered decoding scheduling is described in Algo. 1. The degree d_v of the variable node is assumed to be equal to 2 (a degree that gives good NB-LDPC codes [1]). In the sequel, the check node index i of C_i is omitted, since using $d_v = 2$ suppresses the ambiguity on the check node index during the decoding process.

The decoding process starts with the initialization stage (line 1-4). Then, each iteration is composed of the serial processing of the CN of the matrix (line 5). Each CN processing implies edge permutations and inverse edge permutation (lines 8 and 10). The CN processing (line 9) is presented by the Φ function that processes the d_c incoming $M_{V_j2C}, j \in \mathcal{N}(i)$ messages to generate d_c $M_{C2V_j}, j \in \mathcal{N}(i)$ messages. The function Φ could be any of the different algorithms used for CN processing, such as the Min-Max [13], Extended Min-Sum [3] or Trellis-EMS [9]. After the CN processing, all VNs connected to the CN are updated (line 13-17). The third stage is the decision stage (line 19) which computes the tentative codeword and checks if the decoded message is a codeword (all parity constraints are satisfied).

III. THE BEST, THE REQUESTED AND DEFAULT ALGORITHM

The BRD algorithm is a generic decoding algorithm used with any check node processing algorithm such as [3], [8], [9], [13] and for any variable/check node degree of connectivity d_v and d_c respectively (see Fig. 1(b)). The algorithm allows the variable node to request specific symbols such that they are sent back by the check node. The obtained LLR of the requested elements are concatenated with the most reliable extrinsic elements obtained at the CN and sent all together to the VN. The M_{C2V} message is then considered as a union of three subsets, the first set is M_{C2V}^B , similar to the classical EMS algorithm, which includes the n_B best candidates generated by the CN. The second set M_{C2V}^R is the set of n_R requested candidates by the VN. Lastly, The third set M_{C2V}^D is the *default* set containing the remaining candidates.

This concatenation contributes to increasing the probability of the correct symbol x_j being propagating from/to the CN, hence, increasing the decoding performance of the decoder.

In the sequel, the index i and j of VN and CN (see Fig. 1(a)) are omitted for simplicity. An exponent A^+ indicates the LLR values, and A^\oplus indicates the GF values of the set A . Fig. 1(b) shows the size of the messages, while Fig. 1(c) gives a toy example on GF(8) to illustrate the algorithm.

Algorithm 1: Layered $d_v = 2$ NB-LDPC Decoding Algorithm

Input: $\mathbf{H}, iter_{max}, \{I_j\}_{j=1,2,\dots,N}$
1 Initialization: $iter = 0$
2 for $j = 1$ **to** N **do**
3 $\hat{x}_j = \min(I_j); M_{V_j2C} = I_j; M_{C2V_j}^{old} = 0;$
4 end
5 while $iter < iter_{max}$ & $\mathbf{H} \cdot \hat{\mathbf{x}} \neq 0$ **do**
6 **for** $i = 1$ **to** $N - K$ **do**
7 Check Node Processing:
8 $\forall j \in \mathcal{N}(i), M_{V_j2C}^P = h_{i,j} \times M_{V_j2C}$
9 $\{M_{C2V_j}^P\}_{j \in \mathcal{N}(i)} = \Phi(\{M_{V_j'2C}^P\}_{j' \in \mathcal{N}(i) \setminus j})$
10 $\forall j \in \mathcal{N}(i) M_{C2V_j} = h_{i,j}^{-1} \times M_{C2V_j}^P$
11 Variable Nodes Processing:
12 **forall** $j \in \mathcal{N}(i)$ **do**
13 $APP_j = M_{C2V_j}^{old} + M_{C2V_j} + I_j$
14 $M_{V_j2C} = APP_j - M_{C2V_j}^{old}$
15 $M_{V_j2C} = M_{V_j2C} - \min(M_{V_j2C})$
16 $M_{C2V_j}^{old} = M_{C2V_j}$
17 **end**
18 Decision Stage:
19 **for** $j = 1$ **to** N **do**
20 $\hat{x}_j = \arg \min_{a \in \text{GF}(q)} \{APP_j[a]\}$
21 **end**
22 $iter++$
23 **end**
24 end
Output: $\hat{\mathbf{x}}$

A. Compression and Decompression of M_{V2C} Message

The compression block Ω generates the message $\Omega(M_{V2C})$ from M_{V2C} by selecting the n_{vc} smallest LLRs and their associated GF value. Since the smallest LLR is always equal to 0, $\Omega(M_{V2C})$ is composed of $(n^+, n^\oplus) = (n_{vc} - 1, n_{vc})$ LLR and GF values, respectively. The compression block Ω also sends to the decompression block Γ^{-1} the message $M^{R,\oplus}$ composed of the n_R requested GF symbols, i.e. the GF values of the first n_R couples of $\Omega(M_{V2C})$. The edge multiplicative factor h is applied to each GF element of $\Omega(M_{V2C})$ to generate $\Omega(M_{V2C}^P)$.

The decompression module Ω^{-1} decompresses the message $\Omega(M_{V2C}^P)$ back to a message of size q by setting the n_{vc} GF values of $\Omega(M_{V2C})$ with their corresponding LLRs and by setting infinity to the remaining GF values. Moreover, the decompression block Ω^{-1} also sends to the compression block Γ the message $M^{P,R,\oplus}$ containing the requested symbols. It is composed of the n_R most reliable GF symbols in $\Omega(M_{V2C}^P)$ (see example of Fig. 1(c) with $n_{vc} = 3$ and $n_R = 2$). It could be noted that when the EMS algorithm is used for the check node processing, the Ω and Ω^{-1} modules are bypassed, since EMS, by construction, used truncated messages.

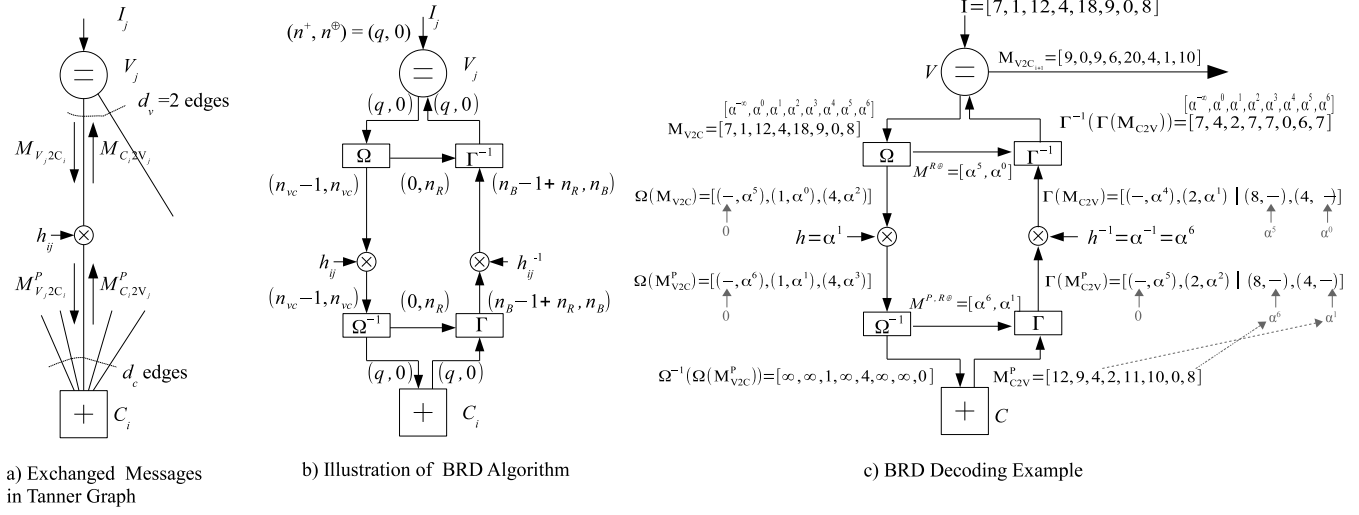


Fig. 1. The Best, the Requested and the Default Decoder

B. Compression and Decompression of M_{C2V} Message

Once all the variable to check messages are received, the CN processes the M_{V2C}^P messages using any processing algorithm such as EMS or T-EMS. The generated M_{C2V}^P message is then truncated by the Γ compression block in two steps. Firstly, the n_B most reliable candidates are selected to generate the subset $M_{C2V}^{P,B}$ of size $(n^+, n^\oplus) = (n_B - 1, n_B)$. Secondly, the LLR of the n_R requested symbols of $M^{P,R\oplus}$ are extracted from M_{C2V}^P and concatenated with $M_{C2V}^{P,B}$ to generate the $\Gamma(M_{C2V}^P)$ message of size $(n^+, n^\oplus) = (n_B + n_R - 1, n_B)$. The n_B GF symbols are inversely permuted by h^{-1} to generate $\Gamma(M_{C2V})$ and sent to Γ^{-1} .

The Γ^{-1} decompression block reconstructs the full-set q message $\Gamma^{-1}(\Gamma(M_{C2V}))$. This reconstruction is based on a 3 steps process. First, the LLRs of the *best* candidates are placed into their corresponding GF positions. Then, the LLR of the *requested* symbols are also set into their corresponding GF positions among with a saturation process that prevents high values from entering the variable node. Finally, the remaining positions are filled with the default LLR value S_D . In summary, for a given GF value $a \in \text{GF}(q)$,

$$\Gamma^{-1}(\Gamma(M_{C2V}))[a]^+ = \begin{cases} M_{C2V}^{B+}[a], & a \in M_{C2V}^{B\oplus} \\ \min(M_{C2V}^{R+}[a], S_R), & a \in M_{C2V}^{R\oplus} \\ S_D, & \text{Otherwise} \end{cases} \quad (1)$$

The saturation values of S_R and S_D impact significantly on the decoder performance. By empirical trial and error, S_R and S_D are determined as $S_R = S + O_R$ and $S_D = S + O_D$, with S given as a linear function of the maximum LLR values of set M_{C2V}^{B+} and M_{C2V}^{R+} , i.e.

$$S = \gamma_B \cdot \max\{M_{C2V}^{B+}\} + \gamma_R \cdot \max\{M_{C2V}^{R+}\} \quad (2)$$

where the values of $(\gamma_B, \gamma_R, O_R, O_D)$ depend on the code rate.

C. BRD Example on $\text{GF}(8)$

In the example of Fig. 1(c), the saturation parameters $(\gamma_B, \gamma_R, O_R, O_D)$ are set to $(2, 1/8, 1, 2)$ respectively. Moreover, the size of the messages are defined as $n_{vc} = 3$, $n_B = 2$ and $n_R = 2$. The intrinsic message I is given as $I = [7, 1, 12, 4, 18, 9, 0, 9]$. The position of each LLR values gives implicitly the associated GF value, as shown in Fig. 1(c). Assuming the processing of the first iteration, the first variable to check message is equal to $M_{V2C} = I$. The compression module Ω extracts the 3 smallest LLR of the message to generate the 3 couples $[(-, \alpha^5), (1, \alpha^0), (4, \alpha^2)]$. Since the LLR of the first couple is always equal to 0, it could be omitted. The message contains 2 LLR and 3 GF symbols. The $n_R = 2$ GF symbols of the most reliable candidates $M^{R\oplus} = (\alpha^5, \alpha^0)$ are also sent to the Γ^{-1} module. The GF value of message $\Omega(M_{V2C})$ are multiplied by $h = \alpha^1$, generating $\Omega(M_{V2C}^P) = [(-, \alpha^6), (1, \alpha^1), (4, \alpha^3)]$. The decompression module Ω^{-1} generates the input check node message $\Omega^{-1}(\Omega(M_{V2C}^P))$ by assigning the LLRs 0, 1 and 4 in positions α^6, α^1 and α^3 , respectively, and infinite value otherwise. It also sends the $n_R = 2$ GF values $M^{P,R\oplus} = (\alpha^6, \alpha^1)$ to the compression module Γ .

In the example, the check node generates the message $M_{C2V}^P = [12, 9, 4, 2, 4, 10, 0, 8]$. The Γ compression block selects the most reliable (Best) $n_B = 2$ candidates from M_{C2V}^P , and it concatenates the LLR of the requested candidates of $M^{P,R\oplus}$, thus $\Gamma(M_{C2V}^P) = M_{C2V}^{B\oplus} \cup M_{C2V}^{R\oplus} = [(-, \alpha^5), (2, \alpha^2)|(8, -), (2, -)]$. After the inverse permutation process, $\Gamma(M_{C2V})$ is received by the Γ^{-1} and expanded back to $q = 8$ as $\Gamma^{-1}(\Gamma(M_{C2V}))$. Since $\max\{M_{C2V}^{B+}\} = 2$ and $\max\{M_{C2V}^{R+}\} = 8$, with the chosen saturation parameters (2) gives $S = 5$, and thus, $S_R = 6$ and $S_D = 7$. The best candidates are kept unchanged, the requested LLR on α^0 is also kept unchanged, while the requested LLR on α^5 is saturated from 8 down to $S_R = 6$. The remaining candidates are all assigned the value $S_D = 7$. The next M_{V2C} message

computed by the VN processing (see Algo. 1, lines (13, 14, 15)) is also indicated in the right output of the VN.

IV. ANALYSIS AND RESULTS

In the EMS algorithm, it is known that the size n_m of the M_{C2V} messages highly affects the performance of the decoder. The statistical study presented hereafter shows that the main criterion that affects the performance is not the size of the message in itself, but the probability $P(x_j \in M_{C2V_j})$ that the transmitted symbol x_j belongs effectively to the exchanged message. It is shown that the *requested* candidates increase this probability significantly, thus leading to good decoding performance even with low message size.

A. Statistical Analysis

In this section, a Monte-Carlo estimation of $P(x_j \in M_{C2V_j})$ is presented as a function of the message size for several decoding algorithms. The Monte-Carlo simulations are performed for a rate 5/6 ($d_v = 2, d_c = 12$) GF(64) NB-LDPC code of size $N = 864$ symbols. The Signal-to-Noise Ratio (SNR) is set to 3 dB (beginning of the waterfall region) with a maximum of 30 decoding iterations. The probability $P(x_j \in M_{C2V_j})$ is estimated in the whole decoding process as a function of the M_{C2V} message length. For the EMS algorithm, the message length is given by n_m . The algorithm parameters are $n_{op} = n_m + 5$ and offset value equals to 0.3 (see [3] for more details). For the BRD algorithm, the message length is characterized by $n_{cv} = n_B + n_R$. The CN process is based on the TEC-TEMS algorithm [9], with specific parameters given in the following section. The TEC-TEMS is also simulated where only $n_m = 20$ candidates are analyzed from the full-set vector $q = 64$.

It is noticeable in Fig. 2 that the impact of including the requested symbols greatly enhances the probability that the transmitted symbol x_j belongs to the propagated message M_{C2V} , and hence becomes a possible candidate to be processed at both VN and CN. The probability $P(x_j \in M_{C2V_j})$ is around 85% in EMS with $n_m = 20$. The probability $P(x_j \in M_{C2V_j})$ in TEC-TEMS has a similar percentage (87%) to EMS. When truncating the size of considered candidates to $n_{vc} = 4, n_B = 7, n_R = 0$, the probability drops down to 59%. Including one requested symbol ($n_B = 6, n_R = 1$) enhances the overall probability from 59% to 89% with the same message size ($n_{cv} = 7$). The proposed decoder with $n_B = 4, n_R = 3$ achieves a probability of 96%.

In Fig. 3, the performance of the aforementioned analyzed schemes is tested down to a FER of 10^{-3} . It is noticeable that the worst performance is for the truncated TEC-TEMS with $n_{vc} = 4, n_{cv} = 7$. Moreover, it is also remarkable that increasing the number of requested symbols further does not enhance the decoding performance of the decoder.

B. Results and Discussion

The number of the exchanged messages reflects the number of memory resources required for the decoding process to be performed. Reducing the number of messages exchanged

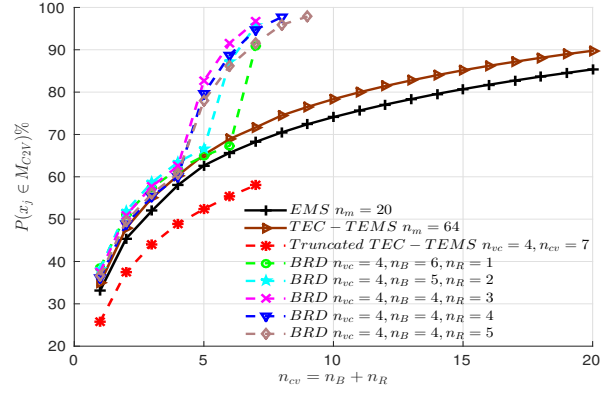


Fig. 2. $P(x_j \in M_{C2V_j})$ for $N = 864, K = 720$ on GF(64)

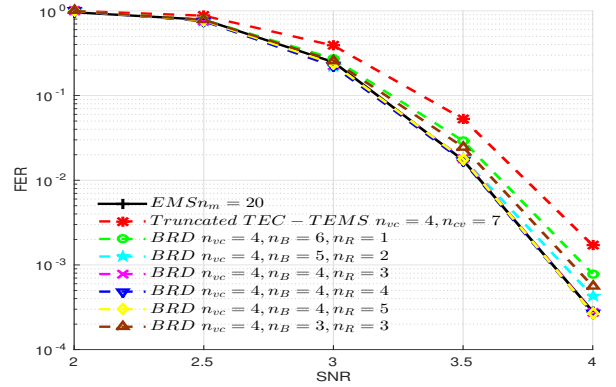


Fig. 3. Decoder Parameters Impact on FER, $N = 864, K = 720$

reduces the latency/complexity of different processes of the decoder such as the sorting processes, the permutation and inverse permutation processes, the configuration sets builder, and the memory resources allocation which benefits both hardware-implemented and software-implemented decoders. The overhead complexity of the compression and decompression block depends on the CN processing algorithm used. Table I shows the size of exchanged messages per edge (a connection between a VN and a CN) for different schemes based on GF(64). The message size is characterized by the number of GF values (coded on $\log_2(64) = 6$ bits) and the number of LLR values (also coded on 6 bits). The last column sums the number of GF and LLR elements exchanged per edge and per iteration. For a code on GF(64) with $r = 5/6$ and a CN degree of connectivity d_c of 12, the lowest number of the exchanged elements among the studied scheme is 48 (Table. I) achieved by the hybrid EMS decoder [6], whereas the total number of exchanged elements per edge in the BRD decoder is 17. Furthermore, the number of the memory resources reserved for M_{V2C} is greatly reduced from $d_c \times q$ down to $d_c \times 7$ and from $d_c \times q$ down to $d_c \times 10$ for M_{C2V} . For a code rate $r = 0.5$ with a degree of connectivity $d_c = 4$, the number of exchanged messages is set to $n_{vc} = 5, n_B = 6$, and $n_R = 3$. The factors γ_R and γ_B are obtained to be 0.127 and 2.24 respectively and optimized to 0.125 and 2 such that the multiplication is

Scheme	Algorithm	Inputs		Outputs		Total
		n_{vc}^{\oplus}	n_{vc}^{+}	n_{cv}^{\oplus}	n_{cv}^{+}	
EMS	FWBW[4]	20	19	20	19	78
	Hybrid [6]	5	4	20	19	48
TEMS	TEMS[7], [8]	0	64	0	64	128
	TEC-TEMS [9]	0	64	0	64	128
	Improved TEMS [10]	32	31	0	64	127
BRD	for $r \geq 5/6$	4	3	4	6	17
	for $r = 1/2$	5	4	6	8	23

TABLE I
NUMBER OF EXCHANGED MESSAGES PER EDGE ON GF(64)

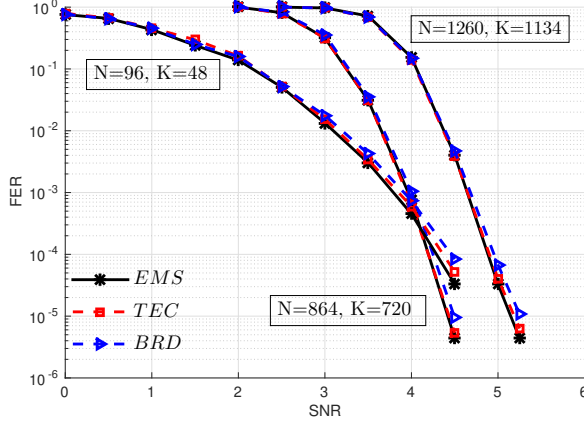


Fig. 4. Simulation Results on GF(64)

simplified to a shifting operation.

The simulation is done for NB-LDPC codes of size (in bits) (96, 48), (864, 720), and (1260, 1134) on GF(64) [14]. The simulated decoder is based on the layered scheduling and a maximum of 10 decoding iterations. As shown in Fig. 4, the proposed scheme has a performance similar to that of the EMS ($n_m = 20$, $n_{op} = 25$) and TEC-TEMS [9]. The performance is simulated down to a FER of 10^{-5} and no significant performance degradation is experienced. The parameters used for simulating the BRD decoder for $r \geq 5/6$ are $n_{vc} = 4$, $n_B = 4$, $n_R = 3$, $\gamma_B = 2$, $\gamma_R = 0.125$, $O_D = 0.4$, $O_R = 0.2$ and a compensation factor (TEC-TEMS) = 0.8. For the code of size $N = 96, K = 48$ with $d_c = 4$, the BRD decoder showed similar performance to EMS (see Fig. 4) with a lower number of exchanged messages $n_{vc} = 5$, $n_B = 6$ and $n_R = 3$ instead of $n_m = 20$ and $n_{op} = 25$. The parameters used for simulation are $\gamma_B = 2$, $\gamma_R = 0.125$, $O_D = 0.2$, $O_R = 0.1$ and a compensation factor (TEC-TEMS) = 0.8.

A binary LDPC code sends 2 messages per edge per iteration. For a $d_v = 3$ (since $d_v = 2$ binary LDPC matrix would have poor performance) LDPC code, each binary VN exchanges 6 messages per iteration. Thus, 6 binary VNs (equivalent to 1 VN in GF(64)) sends 36 messages per iteration. Considering a rate $r \geq 5/6$ and $d_v = 2$ GF(64)-LDPC code, only $17 \times d_v = 34$ messages are exchanged. Therefore, the BRD decoder requires less communication load than a binary decoder.

V. CONCLUSION

The BRD algorithm is a generic decoding, independent of the CN/VN processing or the connectivity degrees d_c and d_v .

The algorithm is based on requesting relevant GF symbols from the CN such that their LLR are concatenated with the best reliable candidates and sent back to the VN. The BRD algorithm allows decreasing the size of the exchanged messages, which reduces the complexity and latency of the decoder without significant performance degradation.

The BRD compression of exchange messages is fully compatible with the EMS message representation. Thus, augmenting the EMS algorithm with the capacity to deliver requested messages is a natural extension of this work. Preliminary results are very promising.

Finally, additional investigations could be done to allow other NB decoders such as NB-Polar or NB-Turbo decoders to benefit also from the BRD algorithm.

ACKNOWLEDGEMENT

This work has been funded in part by the French ANR under grant number ANR-19-CE25-0013-01 and the Lebanese International University.

REFERENCES

- [1] M. Davey and D. Mackay, "Low Density Parity Check Codes over GF(q)," *IEEE Communications Letters*, vol. 2, pp. 165–167, 1998.
- [2] R. Gallager, *Low Density Parity-Check Codes*. Cambridge MA: MIT Press, 1963.
- [3] D. Declercq and M. Fossorier, "Decoding Algorithms for Non-Binary LDPC Codes Over GF(q)," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, 2007.
- [4] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urdard, "Low-complexity Decoding for Non-Binary LDPC Codes in High Order Fields," *IEEE Transactions on Communications*, vol. 58, no. 5, pp. 1365–1375, 2010.
- [5] P. Schl  fer, N. Wehn, M. Alles, T. Lehnigk-Emden, and E. Boutillon, "Syndrome-based Check Node Processing of High Order NB-LDPC Decoders," in *2015 22nd International Conference on Telecommunications (ICT)*, 2015, pp. 156–162.
- [6] C. Marchand, E. Boutillon, H. Harb, L. Conde-Canencia, and A. Al Ghouwayel, "Hybrid Check Node Architectures for NB-LDPC Decoders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 869–880, 2019.
- [7] E. Li, K. Gunnam, and D. Declercq, "Trellis-Based Extended Min-Sum for Decoding Non-Binary LDPC Codes," in *2011 8th International Symposium on Wireless Communication Systems*, 2011, pp. 46–50.
- [8] E. Li, F. Garc  a-Herrero, D. Declercq, K. Gunnam, J. O. Lacruz, and J. Valls, "Low Latency T-EMS Decoder for Non-Binary LDPC codes," in *2013 Asilomar Conference on Signals, Systems and Computers*, 2013, pp. 831–835.
- [9] J. Tian, S. Song, J. Lin, and Z. Wang, "Efficient T-EMS Based Decoding Algorithms for High-Order LDPC Codes," *IEEE Access*, vol. 7, pp. 50 980–50 992, 2019.
- [10] S. Kim, "Improved Trellis-Based Decoder for Non-Binary LDPC Codes," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016, pp. 1–5.
- [11] J. O. Lacruz, F. Garc  a-Herrero, and J. Valls, "Reduction of Complexity for Non Binary LDPC Decoders With Compressed Messages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2676–2679, 2015.
- [12] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-Domain Decoding of LDPC codes over GF(q)," in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, vol. 2, 2004, 772–776 Vol.2.
- [13] V. Savin, "Min-Max decoding for non binary LDPC codes," in *2008 IEEE International Symposium on Information Theory*, 2008, pp. 960–964.
- [14] C. Marchand, H. Harb, T. Gendron, B. Orvoine, and E. Boutillon. (2018). "Free NB-LDPC Code Database of the Lab-STICC Laboratory," [Online]. Available: http://www-labsticc.univ-ubs.fr/nb_ldpc.