Reliability Ratio Weighted Bit Flipping– Sum Product Algorithm for Regular LDPC Codes

Chakir Aqil, [1,*] *Ismail Akharraz*[2] *and Abdelaziz Ahaitouf*[1]

[1]Engineering Sciences Laboratory, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[2]Laboratory of Mathematics and informatics Engineering, Ibn Zohr University, Agadir, Morocco

Abstract. In this work, we introduce a novel decoding algorithm named "Reliability Ratio Weighted Bit Flipping–Sum Product" (RRWBFSP) is proposed for regular LDPC codes. "Sum Product" [4] and "Reliability Ratio Weighted Bit Flipping" [6] are two separate methods that are combined in the new algorithm. The simulations show novel algorithm to exceed Sum-Product decoding algorithms by 0.34 dB. In addition, when compared to the Sum-Product, the RRWBFSP approach has about the same computational complexity. Thus, LDPC codes, of which the "Double-Orthogonal Convolutional Recursive" (RCDO) subfamily is envisioned for use in electronic hard disks and mobile terminals, can be easily iteratively decoded. This would have the impact of prolonging the life of the batteries and consequently reducing the ecological footprint of the discarded batteries.

1 Introduction

In the 1960s, Gallager was the first to discover Low-Density Parity-Check (LDPC) codes [1].LDPC codes are frequently employed in communication systems like as "10 gigabit Ethernet"[2] and "digital video broadcasting" [3] due to their powerful error correcting capacity.
LDPC codes are iteratively decoded by three types of algorithms: hard decision, soft decision and finally hybrid [1, 4].
On the Additive White Gaussian Noise (AWGN) channel, the soft-decision algorithms as sum-product (SP) and Min-Sum (MS) perform well[4], but these methods require a significant number of arithmetic operations to be repeated during iterative decoding. However, the hard-decision Bit Flipping (BF) algorithm [1] and its improved variants, require small computational complexity but incur great performance loss compared to SP or MS. Various hybrid decoding algorithms based on the type of hard decision, as "Weighted Bit Flipping (WBF)"[5] and "Reliability Ratio Weighted Bit Flipping (RRWBF)"[6], have been suggested to reduce the performance gap among soft decision and hard decision types. To improve further the decoding performance, various novel WBF algorithms have recently been proposed. In [7, 8], for example, WBF and BP (SP) are combined. These novel WBF algorithms have significantly better performance than the basic WBF, RRWBF and BF algorithms, but at the cost of increased complexity. The essence of such algorithms is to exploit the performances presented by the algorithms used in the combinations, which leads to the design of a more efficient algorithm by comparing with the basic algorithms [9, 10]. Several algorithms can also be found in the literature that improves performance or simplifies complexity [11, 12].
Aiming at increasing the performance of LDPC codes to getting closer and closer to the Shannon limit, we proposed a new algorithm called Reliability Ratio Weighted Bit Flipping Sum Product, which combines two different algorithms. The main idea of this algorithm is correcting many errors in iteration, that is, the main algorithm receives a frame from the channel and tries to correct the errors, and then the frame passes to the auxiliary algorithm, which corrects, on its part, the errors that have not been corrected by the main algorithm.
The following is a breakdown of the structure of this paper: Section II briefly presents the standard decoding algorithms. Novel decoder "RRWBFSP" is treated in Section III. Complexity of decoding is highlighted in IV. Afterward, In Section V, we compare the error performance of various codes and iterations using the suggested method. Section VI concludes the paper.

2 Standard decoding algorithms of LDPC codes

2.1. Notation

A LDPC code is defined by matrix H, also known as the parity check matrix, which consists largely of 0's with a small amount of 1's. H is a matrice with N columns and M rows. An LDPC code is defined as follows: (N, K, d_v, d_c) The code word length, the information bits, the column weight, and the row weight, in that order. The coding rate is calculated using K/N.

[*] Aqil Chakir: chakir.aqil@usmba.ac.ma

LDPC codes are specified by a graph called a "Tanner graph," which contained two types of interconnected nodes, variable (VN) and control (CN) nodes. Therefore, LDPC codes have N variable nodes and M control nodes. There exists an edge, on the Tanner graph, connected a variable node i and a control node j if only if h_{ij} = 1 in H. An LDPC code (N, K, d_c, d_r) is called regular if d_c and d_r are constants. On the other hand, it is called irregular.

A binary codeword (x_1, x_2,..., x_N) modulated with Binary Phase Shift Keying (BPSK) is assumed in the following. The message is sent via an AWGN channel, while the received message is given by (y_1, y_2, ..., y_N).

Let at the binary hard decision sequence that corresponds to it:

$$z = \begin{bmatrix} z_1 & z_2 & \dots & z_i & \dots & z_n \end{bmatrix} \quad \text{with} \quad \begin{cases} z_j = 0 & \text{if} \quad y_j < 0 \\ z_j = 1 & \text{if} \quad y_j > 0 \end{cases} \tag{1}$$

2.2 "Weighted Bit Flipping Decoding" [5]

The decoding algorithm WBF is based in its process to find the least reliable variable node y_{min} linked to each verification node.

1- Initialization: we consider the parity control matrix H, and we set I=0. For m = 1, 2, …, M, compute

$$y_{min} = min_{\{i \in N(m)\}} |y_i| \tag{2}$$

2- Calculate the syndrom $S = z \,.\, H^T$ (mod 2). If S=0 or when I=Imax, decoding is stopped and output z, otherwise ,I=I+1.

3- For n Є N , calculate the flipping function

$$E_n = \sum_{m \in M(n)} (2S_m - 1) y_{min} \tag{3}$$

4- Find n* Є N with $n^* = arg\ max_{i \in N} E_i$. Flip the variable n*; and go to step 1.

2.3 "Reliability Ratio Weighted Bit Flipping Decoding"[6]

Among the BF based algorithms, the RRWBF decoding performs better. The RRWBF algorithm described in [6] according to the following steps:

1. Initialization: we consider the matrix H, and we set I=0. Calculate Called the Reliability Ratio:

$$R_{mn} = \beta \frac{|y_n|}{|y_m^{max}|} \tag{4}$$

For n and m ranging from 1 to N and from 1 to M respectively

The notation $|y_m^{max}|$ is the largest magnitude of all the VNs involved in the m CN:

$$|y_m^{max}| = max_{\{i \in N(m)\}} |y_i| \tag{5}$$

The parameter β used to ensure that:

$$\sum_{n \in N(m)} R_{mn} = 1 \tag{6}$$

2. Calculate the syndrome S. If S=0 or when I=I_{max}, decoding is stopped and output z, otherwise, I=I+1.

3. For n Є N , calculate

$$E_n = \sum_{m \in M(n)} (2S_m - 1) / R_{mn} \tag{7}$$

4. Find n* Є N with n*=arg $max_{i \in N}$ Ei. Flip the variable n*; and go to step 1.

2.4 Sum – Product Decoding

The set of VNs involved in the control equation i is defined as V(i) = {j : H_{ij} =1}.

The set of CNs involved in updating the VN j is designated by C(j)={i : H_{ij} =1}.

Moreover, V (i) \ j and C (j) \ i mean all the VNs of V (i) excluding node j, and all CNs of C (j) excluding node i, respectively.

And λ_j is the message coming from the logarithmic likelihood ratio of the message received y_i determined by:

$$\lambda_j = \ln\left(\frac{P(x_j = 0 | y_j)}{P(x_j = 1 | y_j)} \right) \tag{8}$$

α_{ij} and β_{ij} are the messages coming from CN i to VN j, and from VN j to CN i, respectively.

The steps of the SPA decoding algorithm are:

1) Initialization: The value λ_i is used to initialize β_{ij} for each i and j. The messages α and β are calculated and sent between VN and CN across the branch of the graph during each iteration, as in the following steps:

2) Check node update: using the β_{ij} values of all VN linked to the CN C_i to calculate the messages α_{ij}, except the β message from V_j:

$$\alpha_{ij} = \prod_{j' \in V(i)\backslash j} sign(\beta_{ij'}) \times \varphi\left(\sum_{j' \in V(i)\backslash j} \varphi(|\beta_{ij'}|) \right) \tag{9}$$

$$\varphi(x) = -\log\left(\tanh\frac{|x|}{2}\right) \tag{10}$$

3) Variable node update: Calculate β_{ij} using λ_j and α from all other CN's to VN j , except CN i:

$$\beta_{ij} = \lambda_j + \sum_{i' \in C(j)\backslash i} \alpha_{i'j} \tag{11}$$

4) Calculate the Syndrome and the termination condition: for each column j the λ_j and α message from nearby CN's are added for update the w_j as flowing:

$$w_j = \lambda_j + \sum_{i' \in C(j)} \alpha_{i'j} \tag{12}$$

An estimated code $\hat{x}_i = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N\}$ is generated from the updated vector. Then the $\hat{x}_i$ is calculated by:

$$\hat{x}_i = \begin{cases} 1, & \text{if } w_i \le 0 \\ 0, & \text{if } w_i > 0 \end{cases} \tag{13}$$

If the condition $H \ . \ \hat{x}^T = 0$ is satisfied, then the codeword is correct and the decoding ends. Otherwise, the procedure continues and the steps repeat until a valid codeword is found or $I=I_{max}$, this is when the decoding process is complete.

3 "Reliability Ratio Weighted Bit Flipping Sum Product Algorithm" (RRWBFSP)

The RRWBFSP algorithm is suggested to improve the decoding performance by using two algorithms which work one after the other; this makes it possible to correct a large number of errors in the same iteration, because the second is used to correct errors which are not corrected by the first algorithm.
The RRWBFSP algorithm is characterized by the following steps:

1. Initialization:
   The value λ_j is used to initialize β_{ij} for each i and j.
2. Check node update:

$$\alpha_{ij} = \prod_{j' \in V(i)\backslash j} sign(\beta_{ij'}) \times \varphi\left(\sum_{j' \in V(i)\backslash j} \varphi(|\beta_{ij'}|) \right) \tag{14}$$

3. variable node update:

$$\beta_{ij} = \lambda_j + \sum_{i' \in C(j)\backslash i} \alpha_{i'j} \tag{15}$$

4. Calculate the Syndrome and the termination condition: :

$$w_j = \lambda_j + \sum_{i' \in C(j)} \alpha_{i'j} \tag{16}$$

An estimated code $\hat{x}_i = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N\}$ is generated from the updated vector. Then the $\hat{x}_i$ is calculated by:

$$\hat{x}_i = \begin{cases} 1, & \text{if } w_i \le 0 \\ 0, & \text{if } w_i > 0 \end{cases} \tag{17}$$

Calculate the syndrome $S=\hat{x} \cdot H^T$. If $S=0$ stop decoding and output $\hat{x}$, or passing in following steps.

5. For $n \in N$, Calculate
   $$E_n = \sum_{m \in M(n)} (2S_m - 1) / R_{mn} \tag{18}$$
6. Find $n^* \in N$ with $n^*=\arg\max_{i \in N} E_i$.
7. Flip n^* and Calculate the syndrome $S=\hat{x} \cdot H^T$, If $S=0$ stop decoding and output $\hat{x}$, or go to step 1.

2 Decoding complexity

The number of operations, such as multiplication, division, and addition, are used to determine the complexity of decoding LDPC codes. The complexity of the RRWBF, SP, and RRWBFSP decoding algorithms is compared here. The number of operations necessary for updating the control and variable nodes, as well as the decision at a single iteration, has been calculated for each algorithm, and the results of the parameters used in the simulation (N, K, $d_v=3$, $d_c=6$) have been grouped in table1.
Table 1 shows that the SP decoding method is more difficult to implement than the RRWBF algorithm, so the proposed RRWBFSP algorithm needs to add a small number of operations compared to SP, which allows us to consider that this algorithm is almost at the same level of complexity as SP .

Table 1: the complexity of several algorithms using an AWGN channel

algorithm / Operation	RRWBF	SP	RRWBFSP
Addition	$(d_c d_v - 1)N + (d_c - 1)M$	$2d_c(d_c - 2)M + d_v^2 N$	$(d_v^2 + d_c d_v - 1)N + (2d_c^2 - 3d_c - 1)M$
multiplication	$N d_c$	$d_c(d_c + 3)M$	$d_c((d_c + 3)M + N)$

Division	N	$2d_c(d_c - 1)M$	$2d_c(d_c - 1)M + N$

5 Error Performance Simulation Results

We utilized Matlab to run simulations to test the performance of the RRWBFSP method.
The performance of two regular LDPC codes (600,300) and (960,480) for the proposed method is shown in this section. The simulations are carried out using an AWGN channel and modulated using BPSK.

A comparison of the RRWBFSP and SP algorithms in performance of error correction with different iterations and codes is presented in curves 1, 2, 3 and 4.
The decoding performance of the RRWBFSP algorithm is evaluated in this section. Figures 1 to 4 present the BER performance of the LDPC code (600, 300) and (960,480) based on the RRWBFSP and SP decoding algorithms with different maxi-mum iteration numbers.

For Figure 1 and 2 we used LDPC codes generated by the parity matrix characterized by (d_v = 3, d_c = 6), N = 600 bits and R = 1/2. For Figure 3 and 4 we used LDPC codes generated by the parity matrix characterized by (d_v = 3, d_c = 6), N = 960 bits and R = 1/2. The number of iterations is limited at 7 and 8 iterations respectively, for each value of (E_b / N_0)dB.

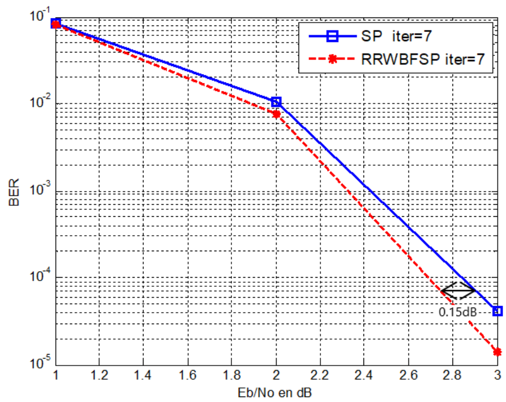

Fig.1: The simulation for N=600 bits and I_{max}=7.

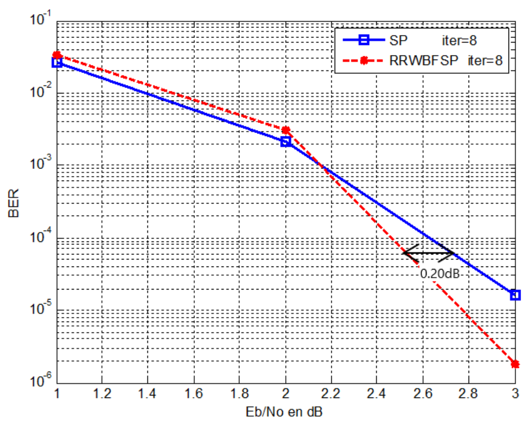

Fig.2: The simulation for N=600 bits and I_{max}=8

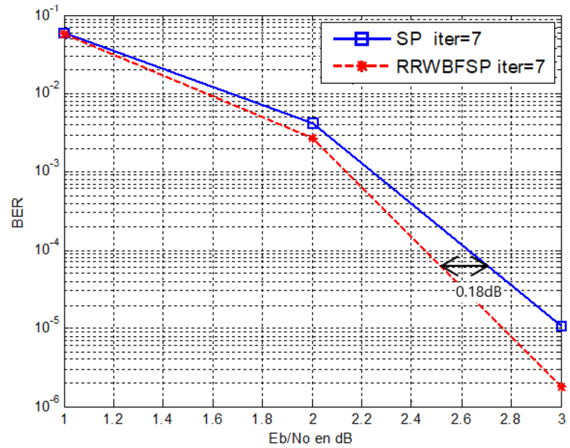

Fig.3: The simulation for N=960 bits and I_{max}=7.

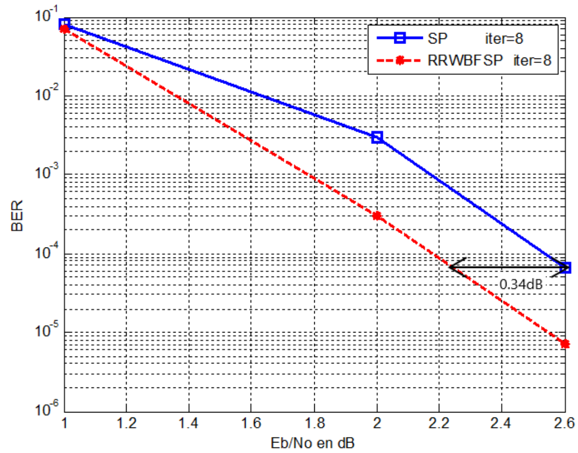

Fig.4: The simulation for N=960 bits and I_{max}=8.

For low SNR values, as observed in all figures, the noise is particularly high, so in the bit flip step, the suggested method and standard flip the bits that are correct and erroneous. As a result, when compared to the standard sum product, the suggested algorithm performs similarly at SNR values of 1 dB to 2 dB.
For high SNR levels of 2 dB or more, the results found by the simulations as shown in all the figures, indicate that the RRWBFSP algorithm offers better performance compared to the SP. We can also observe that the performances are still proportionate with number of iterations and the length of the code word in this range of SNR.
The following table summarizes the results obtained from the four figures:

Table 2: SNR Gain the RRWBFSP compared with SP

at a bit error rate (BER) = 7.10^{-5}			
$I_{max} = 7$		$I_{max} = 8$	
N	**The gain**	**N**	**The gain**
600	0.15 dB	600	0.2 dB
960	0.18 dB	960	0.34 dB

The analysis of the BER graphs and the SNR gain table shows a strong influence the number of iterations and the length of the code word on decoding performance. When increasing the number of iterations and/or the length of the code word this allows the proposed RRWBFSP algorithm to correct several errors, leading to faster convergence than the standard.
On the other hand, when we compare our algorithm with algorithms developed in the same way, for example "Hybrid Iterative Decoding Based on Improved Variable Multi Weighted Bit-Flipping Algorithm (IVMWBF) and BP Algorithms" [9] , we found the following results: Our algorithm, which uses parameters (960, 480) and a maximum iteration number of 8, achieves a binary error rate of 7.10^{-4}, which is better than "Hybrid Iterative Decoding Based on Improved Variable Multi Weighted Bit-Flipping Algorith (IVMWBF) and BP Algorithms" [9], which use (4161, 3431) and have maximum iteration numbers of 10 and 100, respectively [9], by at least 1dB.

6 Conclusion

We introduced a novel algorithm for decoding regular LDPC codes in this work, which we termed "Reliability Ratio Weighted Bit Rollover Sum Product". The results obtained by comparing it with the algorithm existing in the literature called SP, show that we are able to decrease the binary error rate, with a gain of more than 0.3 dB, while maintaining almost the same level of decoding complexity. Then, these codes are applicable in many fields of daily life especially in hard drives and mobile terminals to improve the lifespan of batteries and therefore the contribution of effective way to protect the environment.

References

1. R.G. Gallager, *Low-density parity check codes*, IRE Transaction Info.Theory, vol. IT-**8**, pp. 21–28, Jan. (1962)
2. *IEEE P802.3an, 10GBASE-T task force*, http://www.ieee802.org/3/an
3. *T.T.S.I. digital video broadcasting (DVB) second generation framing structure for broadband satellite applications*, http://www.dvb. org
4. D. J. C. MacKay, *Good error-correcting codes based on very sparse matrices*, IEEE Transactions on Information Theory, vol.**45**, pp. 399–431, Mar.(1999)
5. Y. Kou, S. Lin, and M. *Fossorier, Low density parity check codes based on finite geometries: A rediscovery and more*, IEEE Trans. Inform. Theory, vol. **47**, pp. 2711–2736, Nov. (2001)
6. F. Guo and L. Hanzo, *Reliability ratio based Weighted Bit-flipping decoding for LDPC codes*, IEEE Communications Letters, vol. **40**, pp. 1356-1358, Oct. (2004)
7. E.H. Lu, T.C. Chen and P.Y. Lu, *Theoretic approach to BP-based WBF decoding algorithm of LDPC codes*, Proc. 2013 IEEE Int. Symp. On Wireless and Pervasive Computing (ISWPC), (2013)
8. Q. Zhu and L. Wu, *Weighted-bit-flipping-based sequential scheduling decoding algorithms for LDPC codes*, Math. Prob. Eng., 2013, Article ID 371206. (2013)
9. E. O. Torshizi, H. Sharifi, M. Seyrafi, *A new hybrid decoding algorithm for LDPC codes based on the improved variable multi weighted bit-flipping and BP algorithms,* 21st Iranian Conference on Electrical Engineering,pp.1-6. (2013)
10. H. D. Tiwari and K.Y. Lee, *Hybrid weighted bit-flipping low density parity check decoding*, Digital Signal Processing Journal (Elsevier), **28**(1), 82–92. (2014)
11. T.C.Y. Chang, P.H. Wang and Y.T. Su, *Multi-Stage Bit-Flipping Decoding Algorithms for LDPC Codes*, IEEE Communications Letters, vol. **23**, no. 9, sept. (2019)
12. C. Aqil, I. Akharraz and Az. Ahaitouf, *A New Reliability Ratio Weighted Bit Flipping Algorithm for Decoding LDPC Codes*, Wireless Communications and Mobile Computing ,Volume **2021**, Article ID 6698602, https://doi.org/10.1155/2021/6698602