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

Chakir Aqil, Abdelaziz Ahaitouf, Ismailc Akharraz
Faculty of Taza
Morocco
{chakir.aqil@usmba.ac.ma} {abdelaziz.ahaitouf@usmba.ac.ma}
{ismail.akharraz@usmba.ac.ma}

ABSTRACT: *In this paper, a new algorithm called Reliability Ratio Weighted Bit Flipping–Sum Product (RRWBFSP) decoding is proposed for low-density parity-check codes. The new algorithm combines two different algorithms, Sum Product [4] and Reliability Ratio Weighted Bit Flipping* [6].

The results of the simulation show that the new algorithm achieves a 0.34 dB performance gain over of the standard Sum-Product decoding algorithms. Furthermore, RRWBFSP algorithm has almost the same computational complexity compared to the standard Sum-Product algorithm.

Keywords: Low-density-parity-check Codes, Sum-product, Reliability Ratio, Weighted Bit Flipping– Sum Product

Received: 6 July 2021, Revised 3 September 2021, Accepted 13 September 2021

DOI: 10.6025/dspaial/2022/1/1/28-36

1. Introduction

Low-density parity-check (LDPC) codes were firstly discovered by Gallager in 1960s [1]. Due to powerful error correction capability, LDPC codes are widely used in communication standards such as 10 Gigabit Ethernet (10GBASET) [2] and digital video broadcasting (DVB-S2) [3].

LDPC codes can be iteratively decoded by hard-decision, soft-decision and hybrid iterative decoding algorithms [1, 4].

The soft-decision sum-product algorithm (SPA) (also called Belief Propagation (BP)) and the approximate Min-Sum(MS) algorithm offer the best performance on the additive white Gaussian noise(AWGN) channel [4], but these algorithms require a large number of arithmetic operations repeated during the iteratively 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 SPA or MS. to bridge the performance gap between soft-decision and hard-decision algorithms, various hybrid decoding algorithms were proposed based on the hard-decision decoding algorithm, such as weighted BF (WBF) [5], Reliability Ratio Weighted Bit Flipping (RRWBF) [6]. Recently, several new WBF algorithms have been propos to achieve better decoding performance. For example, WBF is combined with BP (SPA) in [7, 8]. The performance of these WBF algorithms is greatly improved compared to simple WBF, RRWBF and BF algorithms variants, at the expense of a significant increase in 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.

This paper is organized as follows: Section 2 reviews standard iterative decoding of LDPC codes. In Section 3, Reliability Ratio Weighted Bit Flipping–Sum Product decoder is explained. The decoding complexity is shed light on in section4. The error performance comparisons of different codes and different iterations with the proposed methods are highlighted in Section 5. Finally, the conclusion is presented in section 6.

2. Standard Iterative Decoding of LDPC Codes

2.1. Notation

A LDPC code is identified with the parity check matrix H that contains mostly zeros and only a small number of ones. The parity check matrix **H** has N columns and M rows. We describe a LDPC code by (N, K, dv, dc) LDPC code where N, K denote the length of the code word and the information bits, respectively, and dv, dc denotes the column and row weight, respectively. The code rate is given by K/N.

The structure of LDPC codes is well represented by Tanner graph. This graph consists of two kinds of nodes, namely the variable nodes and the check nodes. Consequently, the LDPC code has N variable nodes and M check nodes. An edge exists between a variable node and a check node if and only if there is a "1" in the corresponding entry in the parity check matrix. *A* (N, K) ($j = dv$, $k = dc$) LDPC code has a Tanner graph in which all the variable nodes have degree j and all the check nodes have degree k.

A LDPC code is called regular if its variable node degree j and check node degree k are constants. Otherwise, it is irregular.

In the following, we assume a binary code word (x_1, x_2, ..., x_N) is transmitted using a binary phase-shift keying (BPSK) modulation. Then the sequence is transmitted over an additive white Gaussian noise (AWGN) channel and the received symbol is (y_1, y_2, ..., y_N). Let the corresponding binary hard decision sequence b

$$\mathrm{z} = [\mathrm{z}_1 \quad \mathrm{z}_2 \quad \ldots \quad \mathrm{z}_\mathrm{i} \quad \ldots \quad \mathrm{z}_\mathrm{n}] \text{ with } \begin{cases} \mathrm{z}_\mathrm{j} = 0 & \text{if} \quad \mathrm{y}_\mathrm{j} < 0 \\ \mathrm{z}_\mathrm{j} = 1 & \text{if} \quad \mathrm{y}_\mathrm{j} > 0 \end{cases} \tag{1}$$

2. Weighted Bit Flipping Decoding (WBF)

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

1. Initialization: we consider the parity control *matrix H*, and we set $I = 0$.

For $m = 1, 2, \ldots, M$, compute

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

2. Calculate the syndrom $S = z \cdot H^T \, (mod\, 2)$. If $S = 0$ or the number of iterations reaches a maximum number I_{max} stop decoding and output z, otherwise, $I = I + 1$.

3. For $n^* \in N$, calculate the flipping function

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

Find $n^* \in N$ with $n^* = arg\ max_{i \in N} E_i$. Flip n^* and go to step 1.

2.3. Reliability Ratio Weighted Bit Flipping Decoding (RRWBF)

RRWBF performs best among the BF-based algorithms. The RRWBF algorithm described in [6] according to the following steps:

1. **Initialization:** we consider the parity control *matrix H*, and we set $I = 0$.

2. For $n = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$

Calculate Called the Reliability Ratio (RR): $R_{mn} = \beta \frac{|y_n|}{|y_m^{max}|}$ (4)

The notation $|y_m^{max}|$ is used to denote the highest soft magnitude of all the message nodes participating in the m check:

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

The variable β is the normalization factor introduced to ensure that we have

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

3. Calculate the syndrome $S = z \cdot H^T (mod\ 2)$. If $S = 0$ or the number of iterations reaches a maximum number I_{max} stop decoding and output z, otherwise, $I = I + 1$.

4. For $n \in N$, calculate

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

Find $n^* \in N$ with $n^* = arg\ max_{iN} Ei$.

5. Flip n^* and go to step 1.

4. Sum – Product Decoding

We define $V(i) = \{j : H_{ij} = 1\}$ as the set of variable nodes which participate in the check equation i. $C(j) = \{i: H_{ij} = 1\}$ denotes the set of check nodes which participate in the variable node j update. Also $V(i) \backslash j$ denotes all variable nodes in $V(i)$ except node j. $C(j) \backslash i$ denotes all check nodes in $C(j)$ except node i. Moreover, we define the following variables which are used throughout this paper.

λ_j is defined as the information derived from the log-likelihood ratio of received symbol y_i,

$$\lambda_i = \ln\left(\frac{P(x_i = 0 | y_i)}{P(x_i = 1 | y_i)}\right) \quad (8)$$

α_{ij} is the message from check node i to variable node j.

β_{ij} is the message from variable node j to check node i.

SPA decoding can be summarized in these four steps:

1) Initialization: For each i and j, initialize β_{ij} to the value of the log-likelihood ratio of the received symbol y_j, which is λ_j. During each iteration, α and β messages are computed and exchanged between variable nodes and check nodes through the graph edges according to the following steps numbered 2–4.

2) Row processing or check node update: Compute α_{ij} messages using β messages from all other variable nodes connected to check node C_i, excluding the β information from V_j:

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

Where the non-linear function

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

3) Column processing or variable node update: Compute β_{ij} messages using channel information (λ_j) and incoming a messages from all other check nodes connected to variable node Vj, excluding check node Ci.

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

4) Syndrome check and early termination: When column processing is finished, every bit in column j is updated by adding the channel information (λ_j) and a messages from neighboring check nodes.

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

From the updated vector, an estimated code $\hat{x}_i = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N\}$ is calculated by:

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

If $H \cdot \hat{x}\ T = 0$, then $\hat{x}$ is a valid code word and therefore the iterative process has converged and decoding stops. Otherwise, the decoding repeats from step 2 until a valid code word is obtained or the number of iterations reaches a maximum number, I_{max}, which terminates the decoding process.

3. Reliability Ratio Weighted Bit Flipping– Sum Product Algorithm

Our algorithm is proposed to improve decoding performance by using two algorithms that work one after the other, this allows to correct a large number of errors in the same iteration, because the second algorithm is used to correct errors that are not corrected by the first algorithm.

RRWBFSP decoding can be summarized in these steps:

Initialization: For each i and j, initialize β_{ij} to the value of the log-likelihood ratio of the received symbol y_j, which is λ_j.

2) Row processing or 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) Column processing or variable node update:

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

4) Syndrome check and early termination:

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

From the updated vector, an estimated code vector $\hat{\boldsymbol{x}}_i = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N\}$ is calculated by:

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

Calculate the syndrome S = $\hat{x} \cdot H^T$ (mod 2). If $S = 0$ stop decoding and output $\hat{x}$, or passing in following steps.

For $n \in N$, Calculate $E_n = \sum_{m \in M(n)} (2S_m - 1) / R_{mn}$ (18)

Find $n^* \in N$ with n^* = .

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.

4. Decoding Complexity

The complexity of decoding LDPC codes is evaluated based on the number of operations, such as multiplication, division and

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	Nd_c	$d_c(d_c + 3)M$	$d_c((d_c + 3)M + N)$
Division	N	$2d_c(d_c - 1)M$	$2d_c(d_c - 1)M + N$

Table 1. Complexity for different algorithms with AWGN channel

addition operations. Here we compare the complexity of the RRWBF, SP to RRWBFSP decoding algorithms. For each algorithm the number of operations required for the update of the control nodes and the variable nodes and also the decision at a single iteration has been calculated, the results found of the parameters used in the simulation $(N, K)(dv = 3, dc = 6)$ are grouped in the table 1.

Table 1 shows that the SP decoding algorithm is more complex 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.

5. Error Performance Simulation Results

To evaluate the performance of the algorithm RRWBFSP, we used Matlab to perform simulations.

In this section, the error performance of two regular LDPC codes (600,300) and (960,480) for the proposed algorithm are presented. The simulations are performed over an additive white Gaussian noise (AWGN) channel with binary phase-shift keying (BPSK) modulation.

A comparison of the performances of the proposed algorithm RRWBFSP and the standard decoding algorithm SP are shown in curves 1 and 2, 3 and 4.

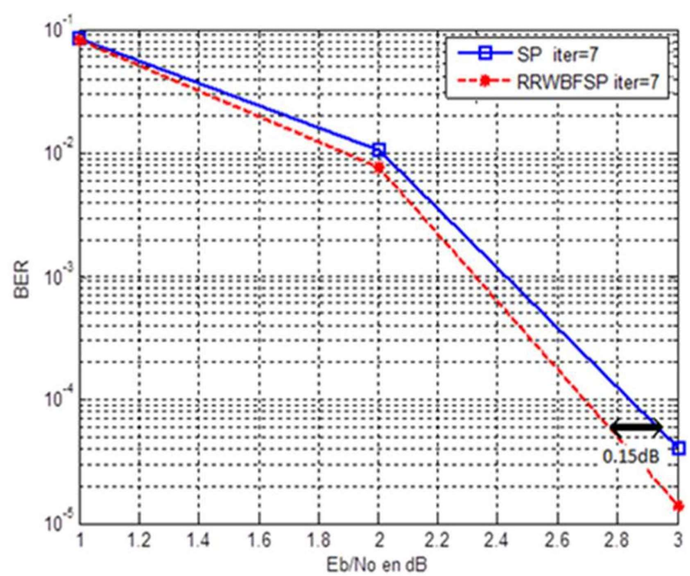

Figure 1. Simulation result for regular LDPC code (3,6) with $N = 600$ bits and $I_{max} = 7$

For Figure 2 we used LDPC codes generated by the parity matrix characterized by ($dv = 3$, dc = 6), $N = 600$ bits and $R = 1/2$. The maximum number of iterations is fixed at 8 iterations for each value of $(Eb / N0)_{dB}$.

For Figure 3 we used LDPC codes generated by the parity matrix characterized by ($dv = 3$, $dc = 6$), $N = 960$ bits and $R = 1/2$. The maximum number of iterations is fixed at 7 iterations for each value of $(\mathrm{Eb} / N0)_{dB}$.

For Figure 4 we used LDPC codes generated by the parity matrix characterized by (dv = 3, dc = 6), N = 960 bits and R = 1/2. The maximum number of iterations is fixed at 8 iterations for each value of $(Eb / N0)_{dB}$.

For low SNR values, as observed in all figures, the noise is very strong, so in the bit flip step, the proposed algorithm and standard flip the bits that are correct and erroneous. Thus, the performance of the proposed algorithm is similar to SNR values of 1 dB to 2 dB compared to the standard product sum algorithm.

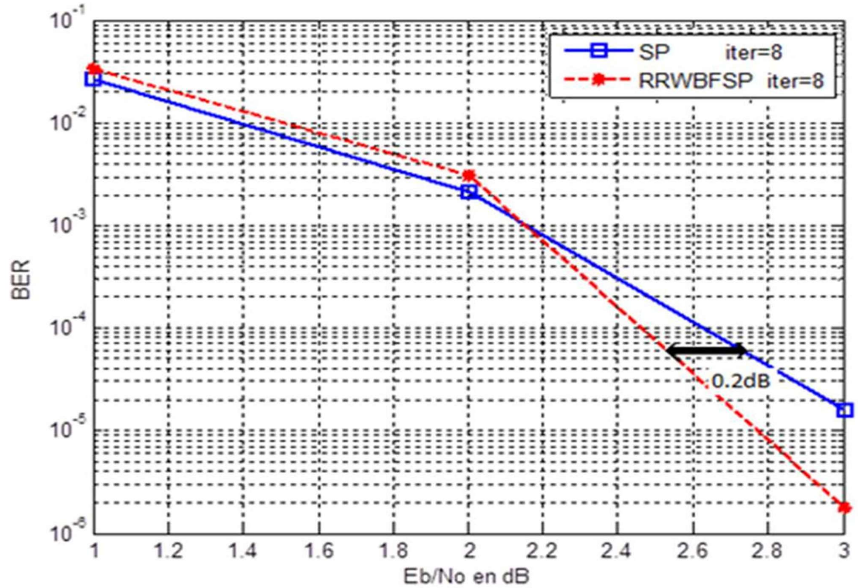

Figure 2. Simulation result for regular LDPC code (3,6) with $N = 600$ bits and $I_{max} = 8$

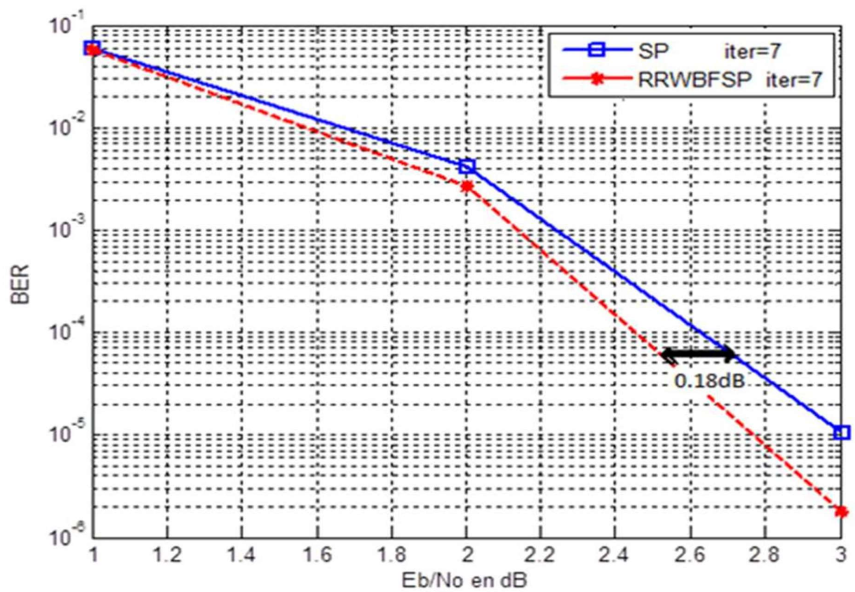

Figure 3. Simulation result for regular LDPC code (3, 6) with $N = 960$ bits and $I_{max} = 7$.

For high SNR values, that is, 2 dB higher, 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 also see that there are still in this range of SNR the performances are related to the number of iterations and the length of the code word. The following table summarizes the results obtained from the four figures:

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

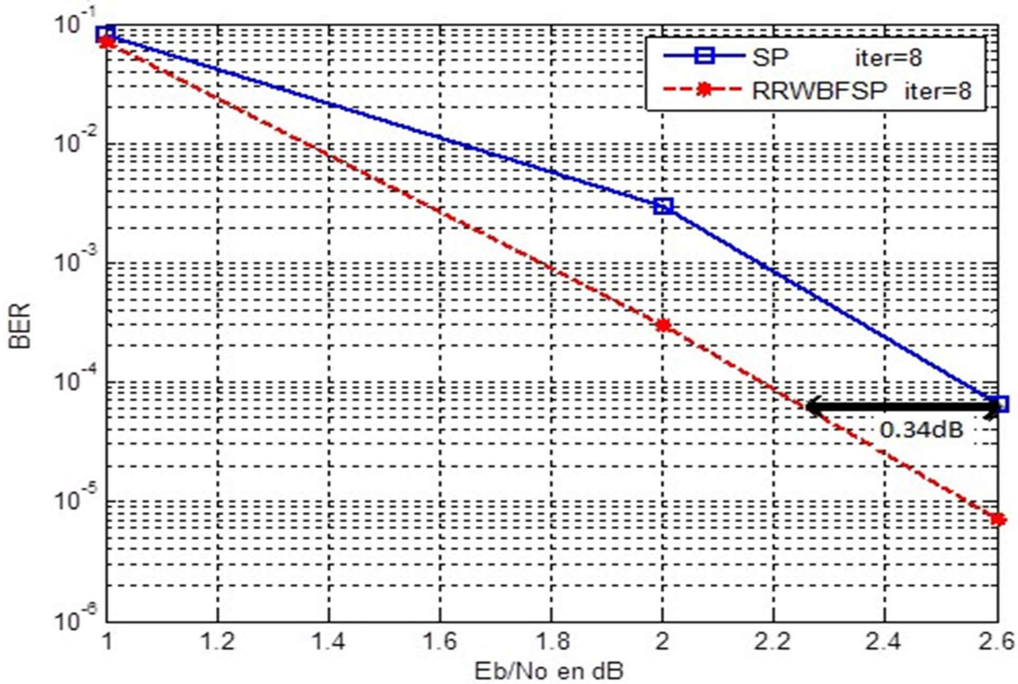

Figure 4. Simulation result for regular LDPC code (3, 6) with $N = 960$ bits and $I_{max} = 8$

At a Bit Error Rate (BER) = 7.10^{-5}	
The Number of Iterations Is Set To 7	
Code word length N	The gain
600	0.15 dB
960	0.18 dB
The number of iterations is set to 7	
Code word length N	The gain
600	0.2 dB
960	0.34 dB

Table 2. SNR Gain the RRWBFSP compared with SP

Decoding Based on Improved Variable Multi Weighted Bit-Flipping Algorithm (IVMWBF) and BP Algorithms , we found the following results: Our algorithm using parameters (960, 480) with a maximum iteration number is 8, for binary error rate 7.10-4, exceeds Hybrid Iterative Decoding Based on Improved Variable Multi Weighted Bit-Flipping Algorithm (IVMWBF) and BP Algorithms using (4161.3431) with the maximum number of iterations for the IVMWBF and BP algorithms has been set at 10 and 100 respectively [9], at least 1dB.

6. Conclusion

In this paper, we proposed a new algorithm for decoding regular LDPC codes, the algorithm named "Reliability Ratio Weighted Bit Flipping Sum Product". The results obtained by comparing it to the existing algorithm in the literature named SP, shows that we are able to reduce the bit error rate, with a gain more 0.3 dB, while maintaining almost the same level of complexity of decoding.

References

[1] Gallager, R.G. (1962). Low-density parity check codes, *IRE Transaction Info.Theory*, IT-8, 21–28, January, 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] MacKay, D. J. C. (1999). Good error-correcting codes based on very sparse matrices, *IEEE Transactions on Information Theory*, 45, 399–431, March.

[5] Kou, Y., Lin, S., Fossorier, M. (2001). Low density parity check codes based on finite geometries: A rediscovery and more, *IEEE Trans. Inform. Theory*, 47, 2711–2736, November.

[6] Guo et, F., Hanzo, L. (2004). Reliability ratio based Weighted Bit-flipping decoding for LDPC codes. (). *IEEE Communications Letters*, 40, 356- 1358, October.

[7] Lu, E.H., Chen, T. C., Lu, P.Y. (2013). Theoretic approach to BP-based WBF decoding algorithm of LDPC codes. *In*: Proceedings 2013 *IEEE Int. Symp. On Wireless and Pervasive Computing (ISWPC)*, 2013.

[8] Zhu, Q., Wu, L. (2013). Weighted-bit-flipping-based sequential scheduling decoding algorithms for LDPC codes, *Math. Prob. Eng*., Article ID 371206.

[9] Torshizi, E. O., Sharifi, H., Seyrafi, M. (2013). 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*, 1-6.

[10] Tiwari, H. D., Tiwari, H. D., Lee, K.-Y. (2014). Hybrid weighted bit-flipping low density parity check decoding. *Digital Signal Processing Journal (Elsevier)*, 28 (1) 82–92.

[11] Tofar, C.-Y., Chang, P. H., Wang, Yu., Su, T. (2019). Multi-Stage Bit-Flipping Decoding Algorithms for LDPC Codes, IEEE Communications Letters, 23 (9) September.

[12] Aqil, C., Akharraz, I., Ahaitouf, Az. (2021). A New Reliability Ratio Weighted Bit Flipping Algorithm for Decoding LDPC Codes, *Wireless Communications and Mobile Computing* , 2021, Article ID 6698602, https://doi.org/10.1155/2021/6698602