IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024TAP0003

Publicized:2024/09/04

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

LETTER

Reliability-List-based Check-Belief Propagation Decoding of LDPC Codes

Zhe LIU[†,††a)], **Wu GUAN**[†††b)], **Ziqin YAN**[††††c)], ***and*** **Liping LIANG**[†d)], ***Nonmembers***

SUMMARY

Reliability-based belief propagation (RBP) decoding algorithms are used to decode low-density parity-check (LDPC) codes. However, due to the reliability of comparing and sorting in traditional algorithms, conventional RBP decoders significantly lose in resource consumption. This letter presents an enhanced reliability list-based check-belief propagation (RL-CBP) algorithm. The RL-CBP algorithm reduces computational complexity by scheduling a concise list of check-beliefs. Moreover, the list is applied for comparisons and selections of check-beliefs. The selected check-belief transforms the decoding message between edges; all check-beliefs are iteratively enlarged according to the reliabilities, and high-performance decoding will be achieved. The simulation results and analyses show that the proposed method achieves a reliability-list gain compared with the check-belief propagation (CBP) algorithm but consumes much fewer calculations than the traditional RBP algorithm.

key words: *Belief propagation (BP), check-belief, low-density parity-check (LDPC) codes, reliability-list, scheduling*

1. Introduction

Since LDPC codes were rediscovered in the 1990s, various low-complexity iterative decoding algorithms have been proposed to balance the performance and complexity of LDPC decoding[1]. The flooding belief propagation (FBP) algorithm is the fundamental decoding algorithm for LDPC code[2], and most decoding algorithms for LDPC code are proposed based on the BP decoding algorithm. It updates all the variable nodes simultaneously using the previously generated check-to-variable (C2V) and then updates all the check nodes simultaneously using the previously generated variable-to-check (V2C). To reduce the complexity, various simplified FBP algorithms are presented, such as the min-sum[3], normalized min-sum algorithms[4]. A shuffled version of the belief propagation (SBP) algorithm is proposed to provide a good trade-off between error performance and complexity for decoding LDPC codes[5]. To speed up the LDPC decoding process, layered belief-propagation (LBP) has been proposed to converge faster than the traditional flooding schedule while allowing parallel decoding of LDPC codes[6]. The LBP method calculates the check-belief propagation in rows or columns to get the check-belief propagation of every node[7], resulting in improved data through rate in decoding. The above non-dynamic message updating strategies only update and propagate messages in predetermined orders. However, although the non dynamic message updating strategies have improved performance compared with the FBP algorithm, applying the latest updating message to the following message update process is still impossible. A kind of BP decoding algorithm based on a reliability scheduling method is presented called residual BP (RBP) to lessen the number of iterations ulteriorly [8]. The decoding algorithms need to speed up the convergence and reduce the cost of the average calculation in each message update process[9]. The FBP and LBP methods require many registers to store V2C and C2V messages, increasing decoding complexity and power consumption.

The check-belief propagation (CBP) decoding method has been presented to reduce decoding complexity[10]. It transfers check-belief message between two check-nodes via only one variable node; compared with other LDPC decoding methods, the CBP renews check-belief propagation through two nodes, which eliminates accumulations and multiplications in the process of LDPC decoding. However, the CBP algorithm cannot take advantage of the reliability of the check nodes.

In order to improve the propagation rate of check-beliefs, the RBP algorithms need to compare the value of all residuals to choose the edges corresponding to the maximum residual for priority decoding. However, the comparisons of residuals lead to increasing resource consumption. To improve convergence, various modified residual-based algorithms including node-wise RBP (NW-RBP)[11], silent-variable-node-free RBP (SVNF-RBP)[12], residual-decaying-based RBP (RD-RBP)[13], and conditional innovation based RBP (CI-RBP)[14] were presented. Aiming to tackle the problem of large amount of residual comparisons, the reliability-list-based check-belief propagation is proposed to decrease the number of comparisons for check-beliefs, which can reduce the decoding complexity with little performance loss.

2. Preliminaries

A binary (N, K) LDPC code with rate $R = K/N$ is defined by a code graph $G = (V, C, E)$, where V, C, and E represent the set of variable-nodes, check-nodes, and edges, respectively. The set V contains N variable nodes, and there are $M = N-K$ check nodes in C. Let $N(v)$ and $N(c)$ denote the adjacent check-nodes of variable-node v and adjacent variable-nodes

[†]The authors are with the School of Integrated Circuits, Beijing University of Posts and Telecommunications, Beijing, 100876, China.
[††††]This work was supported by the National Natural Science Foundation of China (Grant No. 62271069).
a) E-mail: liu_zhe@bupt.edu.cn
b) E-mail: guanwu@bupt.edu.cn(Corresponding author)
c) E-mail: yzqqqaqq@bupt.edu.cn
d) E-mail: liangliping@bupt.edu.cn

of check-node c, respectively. The symbols $N(v)\backslash c$ and $N(c)\backslash v$ denote the set $N(v)$ except for check-node c and the set $N(c)$ except for variable-node v, respectively.

2.1 CBP Decoding

In previous decoding methods such as BP and LBP, the V2C and C2V messages are produced in the cumulative calculations of contiguous nodes, enlarging the decoding complexity.

In order to solve this problem, CBP decoding exchanges each message of LDPC codes between two check nodes. The check-belief is defined as below:

$$\Omega_{c_i} = \log\left(\frac{p_r(Sc_i=0|Y)}{p_r(Sc_i=1|Y)}\right) \tag{1}$$

where Sc_i is the parity check corresponding to check-node c_i. Let Y be the signal received.

The check-belief represents the probability that the parity-check of the check node is satisfied. The check-belief is a positive value if the parity-check is satisfied.

The process of the CBP algorithm can be summarized as follows.

For each check-node c_i, the updating of check-belief is recursively. The latest updated adjacent check-node of variable-node v_a is presented as c_j.

Calculate the updating message of check-belief from check-node c_j to variable-node v_a (B2V) following (2)

$$R^{new}_{c_j\to v_a} = \psi^-(\Omega_{c_j}, Q_{v_a\to c^*}) \tag{2}$$

where

$$\psi^-(x,y) = \phi(|\phi(|x|) - \phi(|y|)|) \cdot sgn(x) \cdot sgn(y) \tag{3}$$

$$\phi(x) = -\log(\tanh\frac{x}{2}) \tag{4}$$

Corresponding messages for variable-node v_a (V2C) updates following (5) and the update of posterior information generates as (6).

$$Q^{new}_{v_a\to c^*} = Q_{v_a\to c^*} + R^{new}_{c_j\to v_a} - R_{c_i\to v_a} \tag{5}$$

$$\Lambda^{new}_{v_a} = Q_{v_a\to c^*} + R^{new}_{c_j\to v_a} \tag{6}$$

Update the V2C message to check-belief (C2B) following (7).

$$\Omega^{(n)}_{c_i} = \psi^+(\Omega^{(n-1)}_{c_i}, Q^{new}_{v_a\to c^*}) \tag{7}$$

where $n = 0, 1, ..., |N(c_i)| - 1$, $|N(c_i)|$ is the number of elements in set $N(c_i)$, and

$$\psi^+(x,y) = \phi(\phi(|x|) - \phi(|y|)) \cdot sgn(x) \cdot sgn(y) \tag{8}$$

$$\Omega^{-1}_{c_i} = \infty, \Omega^{new}_{c_i} = \Omega^{N(c_i)-1}_{c_i} \tag{9}$$

In order to make all the parity-checks satisfied, every check-belief is enlarged by iteration in a serial recursive order. The CBP algorithm propagates check-belief with no cumulative calculations, which causes a low decoding performance loss.

2.2 RBP Decoding

The basic idea of RBP decoding is to adopt a dynamic scheduling strategy, using residuals as a measurement and prioritizing the update for the message of the highest residuals. In the RBP algorithms, the residual is defined as the degree of deviation before and after a message update process. The message transition of RBP is exchanged in descending order of extrinsic information value between edges[15]. The RBP algorithm performs a better convergence than FBP and LBP due to the application of the informed dynamic scheduling strategy[16].

The C2V residual is generated by the magnitude difference between the current C2V message $R_{c_i\to v_a}$ and the precomputed message $R^{pre}_{c_i\to v_a}$.

$$r_{c_i\to v_a} = |R^{pre}_{c_i\to v_a} - R_{c_i\to v_a}| \tag{10}$$

The RBP schedules the edge with the maximum C2V residual to be updated, and each updated C2V message is propagated to its adjacent nodes. In this way, the message of check-node is promoted from all the adjacent nodes renewal, which results in a significantly increasing convergence speed. However, the V2C and C2V processes collect messages from all the neighboring nodes, which increases the calculation complexity of decoding.

3. Reliability-List-Based CBP Decoding

3.1 Reliability-List-Based Check-Belief Scheduling

In CBP decoding, it updates the check-belief for all check nodes. However, the check-belief of some nodes will become stable after a certain number of updates, so it is a waste of resources and may introduce unreliable information. To improve the performance of CBP, residuals are used to select the check-node with higher extrinsic information. Meanwhile, a small reliability list is adopted to reduce the number of check-nodes involved in the residual comparisons. This filters out some check-beliefs that have become stable and accelerated the transmission of newly updated reliable messages to a certain extent. In general, we summarize the decoding process of RL-CBP into two steps:

1) Scheduling decoding the maximum reliability check-node in the reliability list
2) Update the reliability list

The RL-CBP decoding process is shown in Fig.1. In this process, firstly, the C2V message $L^{new}_{c_i\to v_a}$ is updated from the a posterior check-belief Ω_{c_i}. Secondly, the corresponding posterior information $\Lambda^{new}_{v_a}$ of variable-node v_a is updated. Thirdly, the variable node v_a sends a new V2C message $L^{new}_{v_a\to c_j}$ to the check node c_j. Then, the check node c_j updates its check-belief $\Omega^{new}_{c_j}$ in a recursive way. Finally, renew the reliability list if the absolute value of the new reliability $R^{new}_{v_a}$ is higher than the minimum absolute value of check-belief in the list, update check-beliefs and reliabilities for all check nodes connected to the variable-node v_a except c_i, and update reliability and check-belief of check-node until all variable nodes connected to the check-node c_i

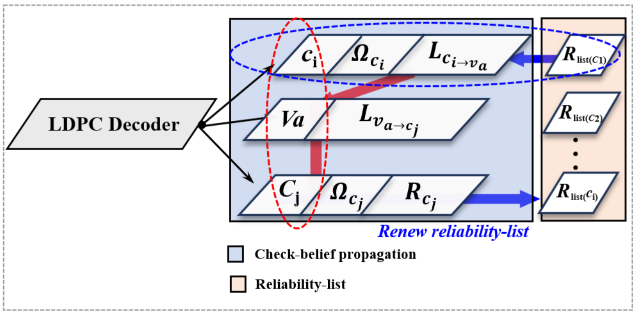

Fig. 1 Reliability-list-based check-belief scheduling

are traversed.

The reliability value of the check-node c_j is calculated by:

$$R_{c_j}^{new} = |\Omega_{c_j}^{new} - \Omega_{c_j}| \tag{11}$$

Based on the above process, the proposed RL-CBP method is summarized by using the pseudocode shown in Algorithm 1. The stopping criteron for RL-CBP is either all the check-beliefs satisfying positive values or the reaching of a predefined maximum number of iterations. The initialization of algorithm 1 is based on the binary phase shift keying (BPSK) modulated additive white Gaussian noise (AWGN) channel.

Algorithm 1 Reliability-List Based Check-Belief Scheduling

1: Initialize all Ω_{c_i}=+∞, R_{c_i}=+∞ all $L_{v_a \to c^*} = 2y_{v_a}/\sigma^2$ where σ represents the noise variance, all $L_{c_i \to v_a}$=0, $L_{c_j \to v_a}$=0
2: Initialize reliability-list $[L_{total}] = 0 \ldots L_{total} - 1$
3: **for** every check node **do**
4: Set the variable nodes v_a connected to c_i as the 0th neighboring node
5: **for** every $v_a \in N(c_i)$ **do**
6: Generate and propagate the external message $L_{v_a \to c_j}^{new}$ by Eq.(2)-(4)
7: Update $\Lambda_{v_a}^{new}$ by Eq.(6)
8: Output the hard decision $\hat{x}v_a$ generated from $\Lambda_{v_a}^{new}$
9: **for** every $c_j \in N(v_a) \backslash c_i$ **do**
10: Update $\Omega_{c_j}^{new}$ by Eq.(7)-(9)
11: Update $R_{c_j}^{new}$ by Eq.(11)
12: Preserve the check node of which reliability values ranks top $[L_{total}]$
13: **end for**
14: **end for**
15: **end for**
16: **for** every $c_i \in L_{total}$ **do**
17: do states 6-11
18: Select the minimum absolute value of $R_{c_i}^{new}$ in $[L_{total}]$
19: **if** the vlaue of $R_{c_j}^{new}$ is higher than $R_{c_i}^{new}$ **then**
20: Renew the list
21: **end if**
22: **end for**
23: **while** not iteration stopping rule **do**
24: Return back to state16
25: **end while**

3.2 Choice of for List Parmeters

We will find an appropriate size for the reliability list by balancing error rate and convergence speed. Too few check nodes in the list may cause a loss of decoding messages and prevent successful decoding. An excessive number of check nodes stored in the list will increase the complexity of calculation and cause difficulties for hardware implementation.

The error rate performance and the convergence speed of the RL-CBP algorithm for different sizes of the reliability list under different irregular LDPC codes are shown in Fig.2 and Fig.3. The regular (3,6) LDPC codes and irregular LDPC codes under degree distributions $(\lambda(x) = 0.45x + 0.3708x^2 + 0.0307x^3 + 0.1485x^{11}$, $\rho(x) = 0.5467x^4 + 0.4533x^5)$ are applied in simulation[17]. The progressive edge growth (PEG) algorithm is used to construct the LDPC codes. In addition, the code rate of LDPC codes is set as 1/2, the simulation employs LDPC codes with code lengths of 2048 and 8192, and the maximum number of iterations is 50.

We can see that the error rate performs best when the L_{total} equals 64. Meanwhile, we can see that the convergence has little increase when L_{total} is bigger than 64. Considering the cost of resource consumption, the L_{total} is selected as 64 for RL-CBP in this letter.

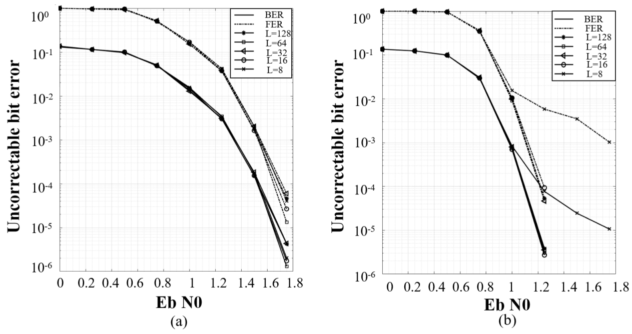

Fig. 2 Performance comparisons of the proposed RL-CBP methods with the reliability list for different lengths. (a) Length-2048 irregular codes. (b) Length-8192 irregular codes.

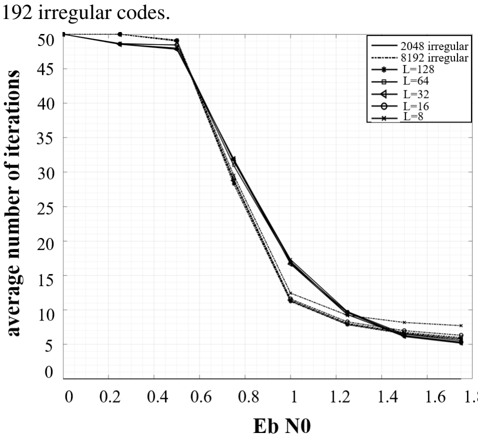

Fig. 3 Iteration performance comparisons of the RL-CBP methods for Length-2048 irregular codes and Length-8192 irregular codes.

4. Simulation Results

In this section, the performances of the traditional algorithms, the CBP algorithm, and the proposed RL-CBP algorithm are analyzed, the setting of simulation parameters is same as section 3.2.

4.1 Error Correction Performance

Fig.4 and Fig.5 illustrate the error correction performance

Table 1 Total Complexity

Schedules	sums	products	Comparison	Dispatching
FBP	$2E$	$2E$	0	$E \cdot max(d_v{}^i, d_c{}^j)$
LBP	E	E	0	$E \cdot max(d_c{}^j)/2$
RBP	$\sum_i E\lambda_i(d_v{}^i - 1)^2/4$	$\sum_{i,j} E\lambda_i\rho_j(d_v{}^i - 1)(d_c{}^j - 1)^2/4 + \sum_j E\rho_j(d_c{}^j - 1)/4$	$E(E-1)/4$	0
CBP	E	E	0	0
RL-CBP	E	E	$ML(d_v - 1)/4$	0

of the different decoding algorithms for both irregular and regular LDPC codes in the AWGN channel. When the 2048 irregular code is used, and the BER approaches 10^{-5}, RL-CBP can obtain a coding gain of approximately 0.06 dB over CBP 0.11 dB over LBP, 0.16 dB over FBP, and is very close to RBP.

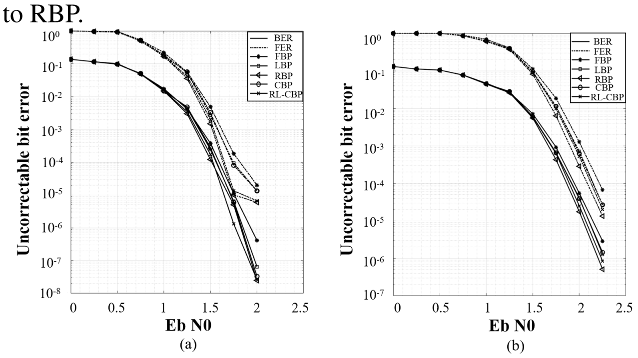

Fig. 4 Performance comparisons of the BP, LBP, RBP, CBP, and the proposed RL-CBP methods (a) Length-2048 irregular codes. (b) Length-2048 regular codes.

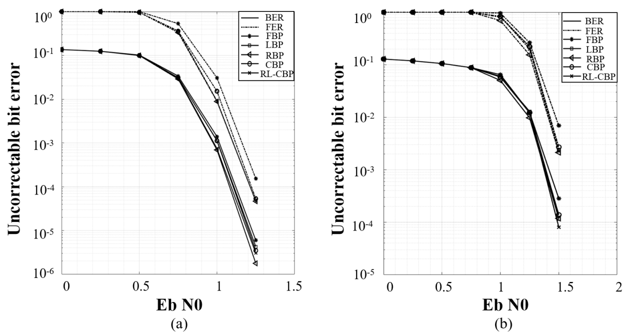

Fig. 5 Performance comparisons of the BP, LBP, RBP, CBP, and the proposed RL-CBP methods (a) Length-8192 irregular codes. (b) Length-8192 regular codes.

The average number of iterations is simulated to measure the convergence speed for LDPC decoding. The simulation results for the regular and irregular codes are shown in Fig.6. We find that the proposed RL-CBP has twice the convergence speed compared to CBP, a similar convergence speed as RBP, and it reaches a lower BER than FBP and LBP under the same number of iterations. The proposed RL-CBP method can combine the advantages of both CBP and RBP approaches. The transform of the decoding message is processed between edges with no cumulative calculation. Furthermore, a limited list of reliabilities is proposed for the reliability comparing, reducing the comparisons for each reliability value in decoding.

4.2 Calculation Complexity

In this subsection, we analyze the decoding complexity for the proposed RL-CBP algorithms according to the number of message updates in each iteration and the calculations required for each message update. Let d_v^i and d_c^j denote the average degrees of variable and check nodes, respectively. The calculation complexities of different BP decoding algorithms are shown in Table 1. The data for FBP, LBP, RBP and CBP in the table are from [10].

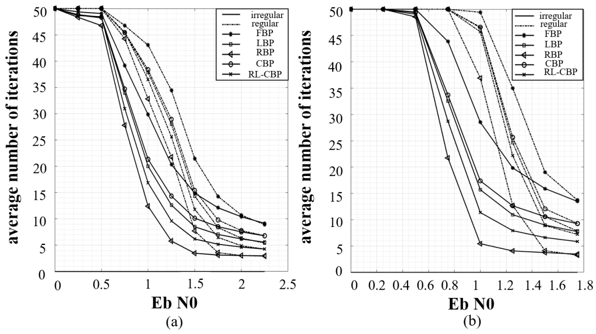

Fig. 6 Convergence performance comparisons of the FBP, LBP, RBP, CBP, and the proposed RL-CBP methods (a) Length-2048 codes. (b)Length-8192 codes.

a) Updates in Each Iteration: For each check-belief renewed process in the proposed strategy, there are d_c^j B2V updates, d_c^j V2C updates, d_c^j C2B updates, and $M \cdot L \cdot (d_v - 1)$ updates of comparison, where L is the short for L_{total}.
b) Calculation in Each Update: In the proposed RL-CBP method, one product exists in the update process of B2V, two sums (including substrates) in the V2C update process, one product in the C2B update, and one comparison for the update of the reliability list.
c) Total complexity: We set the convergence speed of decoding as 1/4 for RBP and RL-CBP to obtain the total complexity. From Table 1, there are much less comparisons in RL-CBP than in RBP. Hence, the complexity of the proposed decoding strategy is much smaller than that of RBP.

5. Conclusion

In this letter, we propose a decoding method of sequence scheduling based on the reliability list in sequence order. It propagates check-belief between edges selected from the small list and reduces the computational complexity. This reduces comparisons in scheduling, and significantly improves the efficiency of the decoding process. Simulation and analysis results show that the proposed algorithm has little performance loss compared with the previous reliability-based algorithms and consumes much fewer comparisons than the traditional RBP algorithms.

References

[1] Wiberg, N., Loeliger, H. A., and Kotter, R. (1995). ”Codes and iterative decoding on general graphs.” European Transactions on telecommunications, 6(5), 513-525.

[2] MacKay, D. J., and Neal, R. M. (1997). ”Near Shannon limit performance of low density parity check codes.” Electronics letters, 33(6), 457-458.

[3] Zhao, J., Zarkeshvari, F., and Banihashemi, A. H. (2005). ”On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes.” IEEE transactions on communications, 53(4), 549-554.

[4] Chen, J., Dholakia, A., Eleftheriou, E., Fossorier, M. P., and Hu, X. Y. (2005). ”Reduced-complexity decoding of LDPC codes.” IEEE transactions on communications, 53(8), 1288-1299.

[5] Juntan Zhang and M. Fossorier, ”Shuffled belief propagation decoding,” Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002., Pacific Grove, CA, USA, 2002, pp. 8-15 vol.1, doi: 10.1109/ACSSC.2002.1197141.

[6] Hocevar, D. E. ”A reduced complexity decoder architecture via layered decoding of LDPC codes.” IEEE Workshop onSignal Processing Systems, 2004. SIPS 2004. (pp. 107-112). IEEE.

[7] Li, H., Ding, H., Zheng, L., Lei, Z., H.,and Wang, T. (2016, October). ”An efficient scheduling scheme for layered belief propagation decoding of regular LDPC codes.” 2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) (pp. 397-400). IEEE.

[8] Casado, A. I. V., Griot, M., and Wesel, R. D. (2007, June). ”Informed dynamic scheduling for belief-propagation decoding of LDPC codes.” In 2007 IEEE International Conference on Communications (pp. 932-937). IEEE.

[9] Yan, Dan, et al. ”Low-loss belief propagation decoder with Tanner graph in quantum error-correction codes.” Chinese Physics B 31.1(2022):010304 (7pp).

[10] W. Guan and L. Liang, ”Check-Belief Propagation Decoding of LDPC Codes,” in IEEE Transactions on Communications, vol. 71, no. 12, pp. 6849-6858, Dec. 2023, doi: 10.1109/TCOMM.2023.3308155.

[11] A. I. V. Casado, M. Griot, and R. D. Wesel, “LDPC decoders with informed dynamic scheduling,” IEEE Trans. Communications, vol. 58, no. 12, pp. 3470–3479, Dec. 2010

[12] H.-C. Lee, Y.-L. Ueng, S.-M. Yeh, and W.-Y. Weng, “Two informed dynamic scheduling strategies for iterative LDPC decoders,” IEEE Trans.Communications., vol. 61, no. 3, pp. 886–896, Mar. 2013.

[13] H. Zhang and S. Chen, “Residual-decaying-based informed dynamic scheduling for belief-propagation decoding of LDPC codes,” IEEE Access, vol. 7, pp. 23656–23666, 2019

[14] T. C.-Y. Chang, P. -H. Wang, J. -J. Weng, I. -H. Lee and Y. T. Su, ”Belief-Propagation Decoding of LDPC Codes With Variable Node–Centric Dynamic Schedules,” in IEEE Transactions on Communications, vol. 69, no. 8, pp. 5014-5027, Aug. 2021, doi: 10.1109/TCOMM.2021.3078776.

[15] Qing, Zhu . ”Low-Complexity Check-Node-Based Serial Scheduling Belief Propagation for LDPC Codes.” Journal of Signal Processing (2013).

[16] Yuan, R. , T.Xie , and Z. Wang . ”A Reliability Profile Based Low-Complexity Dynamic Schedule LDPC Decoding.” IEEE Access 10-(2022):10.

[17] H. -Y. Kwak, J. -S. No and H. Park, ”Design of Irregular SC-LDPC Codes With Non-Uniform Degree Distributions by Linear Programming,” in IEEE Transactions on Communications, vol. 67, no. 4, pp. 2632-2646, April 2019, doi: 10.1109/TCOMM.2018.2889850.