

Improving the Belief-Propagation Convergence of Irregular LDPC Codes Using Column-Weight Based Scheduling

Aslam, Chaudhry Adnan; Guan, Yong Liang; Cai, Kui

2015

Aslam, C. A., Guan, Y. L., & Cai, K. (2015). Improving the Belief-Propagation Convergence of Irregular LDPC Codes Using Column-Weight Based Scheduling. *IEEE Communications Letters*, 19(8), 1283-1286.

<https://hdl.handle.net/10356/81360>

<https://doi.org/10.1109/LCOMM.2015.2444381>

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/LCOMM.2015.2444381>].

Downloaded on 02 Feb 2023 08:43:24 SGT

Improving the Belief-Propagation Convergence of Irregular LDPC Codes using Column-Weight based Scheduling

Chaudhry Adnan Aslam, *Student Member, IEEE*, Yong Liang Guan, *Member, IEEE*,
and Kui Cai, *Senior Member, IEEE*,

Abstract—In this letter, a novel scheduling scheme for decoding irregular low-density parity-check (LDPC) code, based on the column weight of variable nodes in the code graph, is introduced. In this scheme, the irregular LDPC code is decoded using the shuffled belief-propagation (BP) algorithm by selecting the variable nodes in descending order of their column weight. Via numerical simulation, it is shown that the proposed high-to-low column-weight based decoding schedule can noticeably increase the convergence speed at medium to high signal-to-noise ratio (SNR) over AWGN and Rayleigh fading channels without introducing additional complexity or error rate degradation. Furthermore, it is found that the improvement in decoding convergence is proportional to the maximum column-weight in the code graph.

Index Terms—LDPC codes, belief-propagation, shuffled decoding, column-weight, decoding schedule, AWGN, Rayleigh fading.

I. INTRODUCTION

LDPC codes [1], [2] are known for their capacity approaching performance when decoded using belief-propagation (BP) algorithm operating on the Tanner graph (TG) of the code. The BP decoder is an iterative algorithm in which soft messages are generated and passed between the variable and check nodes ($m_{v \rightarrow c}$ and $m_{c \rightarrow v}$) along the edges of the Tanner graph. To execute the BP algorithm, there exist different decoding schedules which define the order of message propagation over the Tanner graph. These schedules can be broadly categorized into two schemes, namely *flooding* and *serial* schedule. In flooding schedule, all the variable node and check node messages are simultaneously computed and propagated along the edges of the Tanner graph. In contrast, in serial schedule, the variable and check nodes are sequentially updated one at a time. In general, for any iterative decoder, the convergence speed or the number of message updates required for convergence are very important from the practical implementation perspective. In this regard, both flooding and serial schedules can be simplified to reduce the computational complexity of BP decoder [3]. Regarding the serial schedule, it can be implemented from the perspective of check node sequencing, or variable node sequencing. In check-node-based serial schedule, also known as layered decoding [4], each

check node and its connected variable nodes are updated one-by-one; whereas in the variable-node-based serial schedule, also known as shuffled decoding [5], each variable node and its neighbors are updated one-by-one.

In this letter, we will focus on the convergence of shuffled BP decoding schedule. As the variable nodes are updated serially by the shuffled BP decoder, the order or sequence of variable node selection becomes very important. An appropriate order of variable node updating can improve the code's error performance as well as the decoding convergence speed. With this motivation, an informed dynamic schedule (IDS) has been proposed in [6] where the variable node with the highest residual magnitude is updated first assuming that the selected variable node has not converged. The IDS scheme speeds up the convergence rate and improves the error performance but also increases the scheduling complexity. There are some other extensions of IDS technique which have further improved on the former scheme [7], [8]. However, they all have introduced additional complexity associated with the residual computations for the variable nodes. In addition to modifying the iterative decoder for faster convergence, other approaches are aimed at designing LDPC codes with inherent fast convergence properties [9], [10].

In this letter, we propose a novel shuffled BP decoding schedule, specifically for irregular LDPC codes, based on the column weight of variable nodes. Since an irregular LDPC code contains variable nodes with different column-weights, we propose to update them in a descending order such that high column-weight variable nodes are updated first, followed by low column-weight variable nodes. We refer to this scheduling as "high-to-low column-weight based scheduling". For the proposed scheme, the decoder sorts the variable nodes by decreasing column weight order only once offline, thereby precluding the need for run-time computations for the selection of variable nodes. Thus, the proposed scheme does not introduce any additional run-time computational complexity. Investigations further reveal that the convergence rate of BP decoder can be noticeably improved using the proposed scheduler.

II. COLUMN-WEIGHT BASED SHUFFLED BP SCHEDULING

The pseudo-code of the proposed column weight based shuffled BP scheduling is given in Algorithm 1. The vector S is used to store the variable nodes in high-to-low column-weight order. Then the variable nodes are serially updated

Chaudhry Adnan Aslam and Yong Liang Guan are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore (e-mail: ad0001ry@e.ntu.edu.sg; EYL.Guan@ntu.edu.sg).

Kui Cai is with the Department of Science, Singapore University of Technology and Design (email: cai_kui@sutd.edu.sg).

following the sequence of vector S . To investigate what brings about the improvement in convergence rate of the shuffled BP decoder, we first note that any sequence of variable node processing leads to the same number of edges to be updated over the Tanner graph per decoding iteration. Thus, in one decoding iteration, each edge is updated only once regardless of which sequence of variable node processing is executed. However, when we update the variable nodes in descending order of their column weight, we effectively allow the decoder to generate a higher number of *aggregated messages* per decoding iteration. Here, aggregated messages refer to the total number of updated messages (soft information, $m_{v \rightarrow c}$) summed over all the variable nodes. This quantity measures the amount of updated information available for processing the next-in-queue variable node. Therefore, a higher number of aggregated messages provides more timely information updates for subsequent variable nodes and helps accelerating the convergence-rate of shuffled BP decoder.

We now present an example to demonstrate the difference in the number of aggregated messages when a particular sequence of variable node processing is followed. In Fig. 1, we show a simple irregular LDPC Tanner graph with five variable nodes, where three nodes have column-weight 2 and two nodes have column-weight 3. The labeling on variable nodes from v_1 to v_5 represents the sequence of node processing. In this figure, both high-to-low and low-to-high column-weight sequencing are shown. We observe that, in both scenarios, the total number of updated edges per iteration are equal to 12. However, in the high-to-low order, individual variable nodes v_2 , v_3 , v_4 and v_5 take advantage of 3, 6, 8 and 10 information updates respectively at the instance of their node processing. On the other side, in the low-to-high order, the same variable nodes v_2 , v_3 , v_4 and v_5 receive 2, 4, 6 and 9 information updates respectively. Overall, the number of aggregated messages for the high-to-low (M_{H-L}) and low-to-high (M_{L-H}) ordering are 39 and 33 respectively. This difference in the number of aggregated messages influence the convergence speed of shuffled BP decoder.

To compute the number of aggregated messages for a general irregular LDPC code, let us denote the fraction of degree- i variable nodes with $\tilde{\lambda}_i$, the maximum variable node degree with $d_{v,max}$ and LDPC code-word length with N . Then, the number of aggregated messages for high-to-low (M_{H-L}) and low-to-high (M_{L-H}) column-weight ordering can be given by (1) and (2) as

$$M_{H-L} = \sum_{i=1}^{d_{v,max}} \left[\sum_{k=1}^{N\tilde{\lambda}_i} \left(\sum_{j=i+1}^{d_{v,max}} j(N\tilde{\lambda}_j) + ik \right) \right] \quad (1)$$

$$M_{L-H} = \sum_{i=1}^{d_{v,max}} \left[\sum_{k=1}^{N\tilde{\lambda}_i} \left(\sum_{j=1}^{i-1} j(N\tilde{\lambda}_j) + ik \right) \right] \quad (2)$$

For the case of random column-weight ordering, we can approximate the number of aggregated messages (M_R) as

$$M_R \approx \bar{d}_v \frac{N}{2} (N + 1) \quad (3)$$

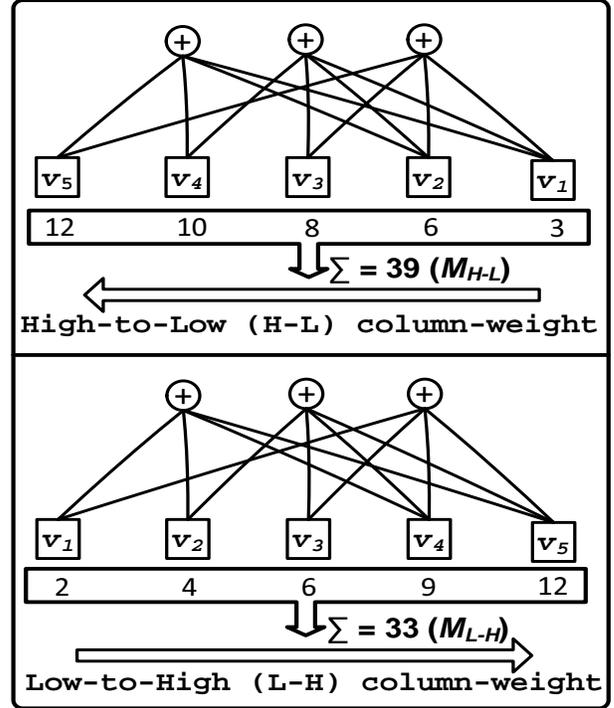


Fig. 1: Illustration of message propagation: upper-part is the proposed high-to-low column-weight based scheduling; lower-part is low-to-high column-weight based scheduling.

where \bar{d}_v is the average variable node degree, given by

$$\bar{d}_v = \sum_{i=1}^{d_{v,max}} i\tilde{\lambda}_i$$

To plot the number of aggregated messages, we make use of the capacity approaching rate-1/2 irregular LDPC codes with optimized degree distributions as given in [11]. For $N = 2000$ and LDPC codes with $d_{v,max} = 5$, $d_{v,max} = 10$, $d_{v,max} = 20$, $d_{v,max} = 30$ and $d_{v,max} = 50$ obtained from [11], we plot M_{H-L} , M_R and M_{L-H} quantities in Fig. 2. In this figure, we notice that the value of M_{H-L} is larger than both M_{L-H} and M_R as expected. Furthermore, we also observe that the value of M_{H-L} is proportional to the maximum variable node degree $d_{v,max}$. It will next be shown that the convergence speed is correlated with $d_{v,max}$.

Algorithm 1 : Column-Weight based Shuffled BP Schedule.

- 1: Load vector $S = \{v_1, v_2, \dots, v_N\}$, \Rightarrow variable nodes sorted by high-to-low column-weight order
 - 2: **for** all $n = 1 : N$ **do**
 - 3: **for** every $c_m \in$ neighborhood of v_n **do**
 - 4: Generate and propagate $m_{c_m \rightarrow v_n}$
 - 5: **end for**
 - 6: **for** every $c_m \in$ neighborhood of v_n **do**
 - 7: Generate and propagate $m_{v_n \rightarrow c_m}$
 - 8: **end for**
 - 9: **end for**
-

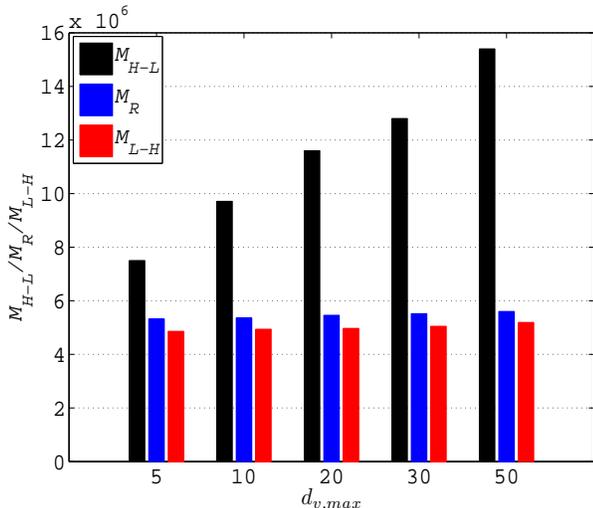


Fig. 2: Number of aggregated messages for high-to-low (M_{H-L}), random (M_R) and low-to-high (M_{L-H}) column-weight based scheduling schemes: Using $N = 2000$ and LDPC codes [11] with $d_{v,max} = 5$, $d_{v,max} = 10$, $d_{v,max} = 20$, $d_{v,max} = 30$ and $d_{v,max} = 50$ respectively.

III. DECODING CONVERGENCE AND ERROR RATE PERFORMANCE

In our simulations, we use the progressive-edge-growth (PEG) [12] algorithm to construct the parity-check matrices for the irregular LDPC codes given in [11]. This algorithm is well-known to construct finite-length LDPC codes with very good error performance. However, it is pertinent to mention that our proposed scheme is not just applicable to PEG algorithm but can also be incorporated with other code construction schemes, e.g. ACE [13] algorithm. To show the improvement in convergence-rate using Monte Carlo simulations, we first plot the average mutual-information (\mathcal{MI}) computed at the output of variable node processor versus the number of decoding iterations, allowing high-to-low (\mathcal{MI}_{H-L}) and low-to-high (\mathcal{MI}_{L-H}) column-weight ordering for the $d_{v,max} = 50$ code, as shown in Fig. 3. We observe that, for moderate channel SNR values, the mutual-information tends to converge faster for the proposed high-to-low column-weight based scheduling.

Using the same irregular LDPC codes, we simulate and plot the normalized average number of decoding iterations to visualize the increase in the convergence speed. In addition to rate-1/2 LDPC codes optimized for AWGN channel [11], we also plot the normalized decoding iterations for rate-1/2 LDPC codes optimized for Rayleigh fading channel [14]. For that, we denote I_{H-L} , I_{L-H} and I_R as the average number of LDPC iterations performed per decoded code-word for the high-to-low, low-to-high and random column-weight based scheduling schemes respectively. Then, the normalized average number of iterations can be given by the ratio I_{H-L}/I_{L-H} and I_{H-L}/I_R , as shown in Fig. 4 and Fig. 5 respectively. From these two figures, we notice that the proposed high-to-low column-weight based scheduling scheme speeds up

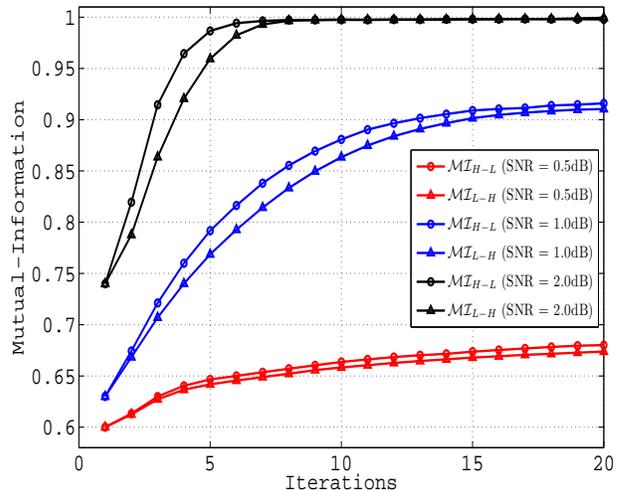


Fig. 3: Illustration of mutual-information curves for the high-to-low (\mathcal{MI}_{H-L}) and low-to-high (\mathcal{MI}_{L-H}) column-weight based scheduling schemes for $d_{v,max} = 50$ LDPC code [11] over AWGN channel.

the convergence-rate. To be more precise, for $d_{v,max} = 50$ code over AWGN channel, we get around 37% and 29% faster convergence speed compared to low-to-high and random column-weight based scheduling. Similarly for $d_{v,max} = 50$ code over Rayleigh fading channel, the proposed scheme increases the convergence speed by upto 35% and 22%. We also observe that the increase in the convergence speed is proportional to the maximum column-weight $d_{v,max}$, since the number of aggregated messages are directly related to this parameter as explained earlier.

To compare the error performance of different column-weight based scheduling schemes, we simulate the bit-error-rate (BER) and frame-error-rate (FER) curves and plot them in Fig. 6 for $d_{v,max} = 10$ and $d_{v,max} = 50$ codes over AWGN and fading channels using BPSK modulation. We observe that the error performance is not affected by any sequence of column-weight based processing. Hence, the convergence speed of shuffled BP decoder for an irregular LDPC code can be increased using high-to-low column-weight based scheduling, without incurring additional complexity or error rate degradation.

Finally, as a further merit of the proposed work, it is relevant to comment that the proposed high-to-low column-weight based scheduling is compatible with the existing low-complexity decoding schemes, such as [15], [16] wherein partial set of variable nodes are updated per decoding iteration, and can be used in complementary manner for achieving faster convergence speed.

IV. CONCLUSION

Noting the importance of decoding schedule on the complexity and convergence rate of iterative LDPC decoder, we present a novel shuffled BP decoding schedule based on the column weight of variable nodes. Specifically, we propose to

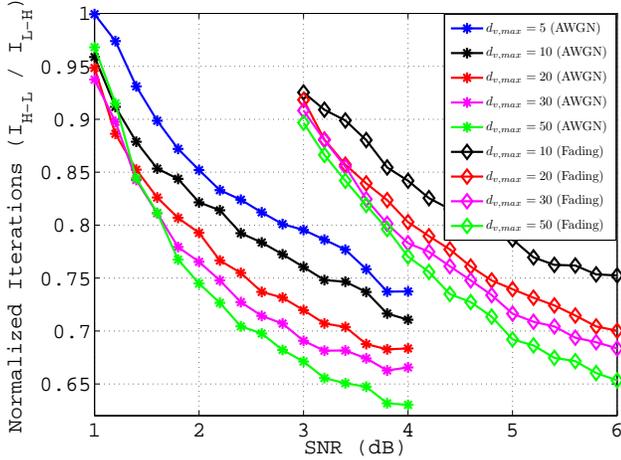


Fig. 4: Average number of decoding iterations for the proposed high-to-low column-weight ordering normalized by low-to-high column-weight ordering over AWGN and fading channels.

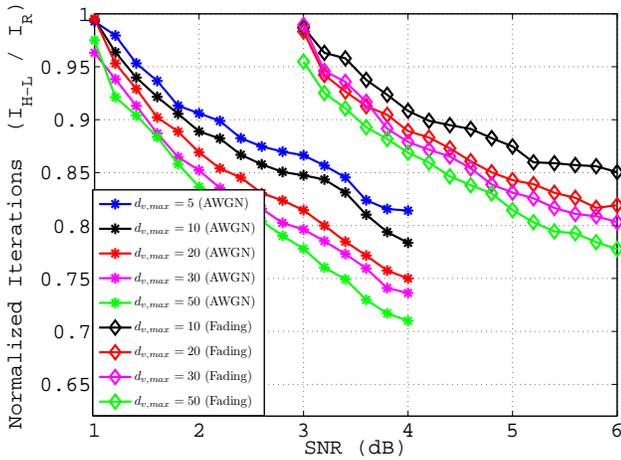


Fig. 5: Average number of decoding iterations for the proposed high-to-low column-weight ordering normalized by random column-weight ordering over AWGN and fading channels.

update the high column-weight variable nodes first, followed by the low column-weight variable nodes. This schedule requires sorting of variable nodes in descending order of their column-weights only once in the offline mode, eliminating the need for run-time computations. Using computer simulations, we observe a noticeable improvement in the convergence behavior of BP decoder by plotting the mutual-information curves and the normalized average number of decoding iterations for different rate-1/2 irregular LDPC codes operating in AWGN and Rayleigh fading channels. Furthermore, we show that the convergence speed improves with higher maximum variable node degree $d_{v,max}$ of the LDPC code, and the speedier convergence does not degrade the error performance.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," 1963.

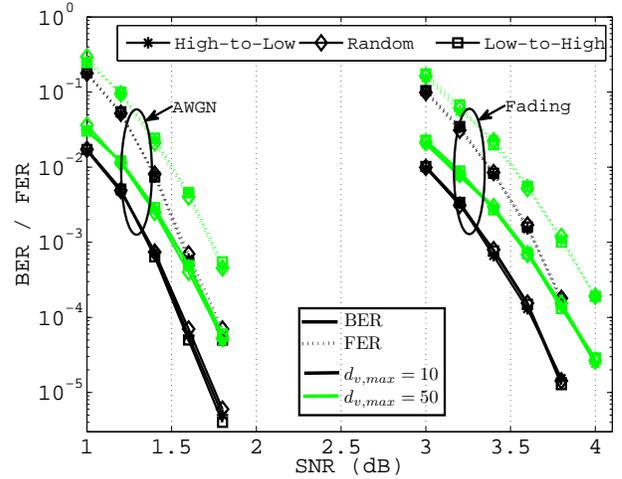


Fig. 6: BER and FER performance of high-to-low, random and low-to-high column-weight based scheduling for $d_{v,max} = 10$ and $d_{v,max} = 50$ codes over AWGN and fading channels.

- [2] D. MacKay and R. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457–458, Mar 1997.
- [3] G. Han, Y. L. Guan, and X. Huang, "Check node reliability-based scheduling for BP decoding of non-binary LDPC codes," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 877–885, March 2013.
- [4] E. Sharon, S. Litsyn, and J. Goldberger, "An efficient message-passing schedule for LDPC decoding," in *Proc. 23rd IEEE Conf. Electrical and Electronics Engineers in Israel*, Sept 2004, pp. 223–226.
- [5] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," in *Proc. 36th Asilomar Conf. Signals, Systems and Computers*, vol. 1, Nov 2002, pp. 8–15.
- [6] A. Vila Casado, M. Griot, and R. Wesel, "LDPC decoders with informed dynamic scheduling," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3470–3479, December 2010.
- [7] Y. Gong, X. Liu, W. Yecai, and G. Han, "Effective informed dynamic scheduling for belief propagation decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 59, no. 10, pp. 2683–2691, October 2011.
- [8] H.-C. Lee, Y.-L. Ueng, S.-M. Yeh, and W.-Y. Weng, "Two informed dynamic scheduling strategies for iterative LDPC decoders," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 886–896, March 2013.
- [9] A. Grant, I. Land, and G. Wang, "Iteration-constrained design of IRA codes," in *Proc. IEEE Int. Zurich Seminar Commun.*, Mar 2012.
- [10] I. Mulholland, E. Paolini, and M. Flanagan, "Design of LDPC code ensembles with fast convergence properties," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw.*, May 2015.
- [11] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 619–637, Feb 2001.
- [12] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan 2005.
- [13] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Construction of irregular LDPC codes with low error floors," in *Proc. IEEE Conf. Communications*, vol. 5, May 2003, pp. 3125–3129 vol.5.
- [14] J. Hou, P. Siegel, and L. Milstein, "Performance analysis and code optimization of low density parity-check codes on rayleigh fading channels," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 924–934, May 2001.
- [15] E. Zimmermann and G. Fettweis, "Reduced complexity LDPC decoding using forced convergence," in *Proc. 7th Int. Symp. Wireless Personal Multimedia Communications*, 2004, pp. 12–15.
- [16] D. Levin, E. Sharon, and S. Litsyn, "Lazy scheduling for LDPC decoding," *IEEE Commun. Lett.*, vol. 11, no. 1, pp. 70–72, Jan 2007.