Trapping Set Error Correction through Adaptive Informed Dynamic Scheduling Decoding of LDPC Codes

Saejoon Kim, *Member, IEEE*

***Abstract*—Recent studies indicate that sequential scheduling of iterative decoding of low-density parity-check codes can provide improved code performance in comparison with the conventional flooding in terms of decoding success rate and convergence speed. In particular, a class of sequential scheduling schemes known as informed dynamic scheduling (IDS) has shown to outperform other sequential scheduling schemes from its capability to correct trapping set errors. In this letter, we present an *adaptive* IDS scheme which combines and outperforms previously proposed sequential scheduling schemes.**

Index Terms- LDPC codes, informed dynamic scheduling, trapping sets.

I. Introduction

Low-density parity-check (LDPC) codes are generally decoded by the iterative belief propagation (BP) algorithm which offers low complexity without much sacrifice in performance relative to the maximum-likelihood decoding. Many versions of the BP algorithm currently exist, and recent studies [1], [2], [3] and references therein have shown that sequential scheduling schemes of BP decoding can improve code performance compared to the conventional parallel scheduling scheme or flooding. Specifically, it was shown that sequential scheduling can yield better asymptotic word-error-rate (WER) performance than and require half as many number of iterations to converge as its parallel counterpart [9], [12]. Examples of sequential scheduling schemes include layered belief propagation (LBP) [6], residual belief propagation (RBP) [1], node-wise residual belief propagation (NWRBP) [1] and informed variable-to-check residual belief propagation (IVC-RBP) [5] among others. Sequential scheduling schemes can broadly be categorized according to the rule with which the messages are updated. In the deterministic scheduling scheme, which includes LBP, the order of message updates is predetermined, whereas in the informed dynamic scheduling (IDS) scheme, which includes RBP, NWRBP and IVC-RBP, the order of message updates dynamically changes depending on the state of the decoding process.

IDS schemes have shown to be effective at correcting some of trapping set errors which contribute to the error floor behavior over the additive white Gaussian noise (AWGN) channel. We note that trapping set errors are those that cannot be corrected by flooding or sequential deterministic scheduling schemes. While many of IDS schemes have shown to outperform other sequential scheduling schemes for this reason, it has also been observed that IDS schemes may incur some non-trapping set errors that other schemes are able to correct. For this matter, Casado *et al.* [2], [3] have proposed a mixed scheduling scheme in which LBP is first processed to take care of non-trapping set errors while NWRBP is later processed to solve trapping set errors that often occur at higher signal-to-noise ratios (SNRs).

In this letter, we present an improved mixed scheduling scheme based on the decoding structure proposed in [2] through adopting the techniques used in [7] and [8]. Specifically, we present an adaptive informed dynamic scheduling scheme in which the scheduling scheme changes *adaptively* according to the state of the decoding process in order to improve code performance. It will be demonstrated that our proposed scheme outperforms previously presented scheduling schemes at every decoding iteration for a wide range of SNRs.

II. Informed Dynamic Scheduling Decoding

In an IDS decoding, the notion of *residual* [4] is utilized to schedule the order of message updates in which the message with the largest residual is updated first. Here, the residual of a message is defined as the difference in the message between after and before its update. This results in only the most unreliable parts of the graph being updated at a time leading to faster decoding convergence and correction of trapping set errors [1], [3]. RBP and NWRBP are the main realizations of IDS, and they have shown to perform extremely well for a small and mid-to-large number of decoding iterations, respectively.

Let V and C represent the sets of variable and check nodes, respectively, and let $\mathcal{N}(x)$ denote the set of neighbors of x. The residuals of all messages are first initialized to 0. Then in RBP, all the variable-to-check node messages m_{xc}, $x \in V,\ c \in C$, are sent and only the check-to-variable node message $m_{c'x'}$ with the largest residual for some x' and c' is sent. After sending this check-to-variable node message, its corresponding residual is reset to 0 and only the residuals of variable-to-check node messages $m_{x'c}$ where $c \in \mathcal{N}(x')\backslash c'$ change which means that only these $m_{x'c}$'s are updated. Then all m_{cx}'s are calculated for $c \in \mathcal{N}(x')\backslash c'$ and $x \in \mathcal{N}(c)\backslash x'$, and the message corresponding to the largest residual among all the check-to-variable nodes is sent. This process is repeated in RBP decoding and the difference with NWRBP decoding lies only in the generation of check-to-variable node messages where the messages $m_{c'x}$, $x \in \mathcal{N}(c')$, are all sent simultaneously in NWRBP if the message $m_{c'x'}$ has the largest residual for some $x' \in V$ and $c' \in C$. In other words, for the check node associated with the largest check-to-variable node

Manuscript received February 6, 2012. The associate editor coordinating the review of this letter and approving it for publication was A. Burr.

S. Kim is with the Department of Computer Science and Engineering, Sogang University, Seoul, Korea (e-mail: saejoon@sogang.ac.kr).

The work of S. Kim was supported by the Sogang University Research Grants of 2010 and 2011.

Digital Object Identifier 10.1109/LCOMM.2012.050112.120262

residual, all of its check-to-variable node messages are updated simultaneouly in NWRBP. NWRBP is clearly a less greedy version of RBP, and it has the property of not converging as quickly as but having better, with respect to decoding iterations, asymptotic performance than RBP. In particular, it has been shown that RBP and NWRBP outperform any other known iterative decoding scheme in which the order of message updates is predetermined such as flooding or LBP for a small and mid-to-large number of iterations, respectively.

IDS decoding has its weakness, however, especially for short length codes, which is its inability to correct some of non-trapping set errors that a deterministic scheduling decoding such as flooding or LBP may correct. This weakness results from its greedy structure concentrating in updating only parts of the graph at a time. To mitigate this problem, a mixed scheduling scheme in which the decoding process begins with a deterministic scheduling scheme to correct non-trapping set errors and later switches to a dynamic scheduling one was proposed in [2]. This mixed scheduling scheme bases its efficacy on the fact that the most problematic trapping sets have a small number of unsatisfied check nodes [11] which often occurs later in the decoding iterations. LBP and NWRBP were used as the deterministic and dynamic scheduling schemes in [2], respectively, and a predetermined number of unsatisfied check nodes was used as the threshold measure for the switch of the two scheduling schemes.

III. Proposed Mixed Scheduling Decoding

The mixed scheduling scheme of [2] seems to be an ideal decoding structure in the sense that both non-trapping and trapping set errors are being corrected. Our goal is to improve this scheme by noting that it does not perform very well for a small number of decoding iterations and the switch condition from LBP to NWRBP does not adapt well to different decoding environments. The reason for the former note results from the characteristics of LBP and that for the latter results from the fact that the number of unsatisfied check nodes can vary nontrivially for different environments such as iteration numbers and SNRs.

To this end, an attempt to resolve the latter problem has been proposed in [8] with success using a weighted log-likelihood ratio (LLR) value for the switch condition. Specifically, it turned out that a local maximum of the sum of $|\text{LLR}(i)|/\text{deg}(i)$ over all variable nodes (i's) is a much easier quantity to compute and provide better code performance than the "minimum" value of unsatisfied check nodes in the duration of decoding for the switch condition where $\text{deg}(i)$ denotes the degree of variable node i. Since smaller value of unsatisfied check nodes for the switch threshold generally yields better code performance than otherwise, it seems that the described weighted sum of LLR values as the switch criterion cannot be improved drastically. In [8], the second local maximum of the weighted LLR sum was used to obtain performance that was superior to any other known iterative decoding algorithm.

So now the former problem remains to be solved. To this end, since RBP provides the best known decoding performance in the early iterations by propagating messages only to the less reliable nodes first, a natural approach is to process RBP first and then switch over to the mixed scheduling of LBP and NWRBP. For this matter, consider the decoding scheme proposed in [7] where RBP and NWRBP are processed in a sequence providing performance that is superior or equal, at every iteration, to either of the two schemes processed alone. An implication of this result is that errors, including possibly trapping set errors, that RBP and NWRBP are capable of correcting do not coincide. Moreover, if the switch from RBP to NWRBP is made appropriately, processing RBP first does not deteriorate the code's asymptotic performance. We will exploit this implication and extend it to the mixed scheduling of LBP and NWRBP schemes of [8]. Thus, in our proposed decoding algorithm, three scheduling schemes, RBP, LBP and NWRBP, will be processed in a sequence where the switch from LBP and NWRBP has already been described. The switch from RBP to LBP should be made when the advantage of RBP in the early iterations has tapered off which is just before the convergence of its WER. To this end, in [7] it was observed that S_l, the number of neighbors of the set of unsatisfied check nodes after decoding iteration l, converges slightly faster than the WER of RBP decoding, and the condition for its convergence detection was provided. We will use this condition, described below, as the switch criterion from RBP to LBP in our mixed scheduling scheme. The overall scheme is then, by construction, expected to provide performance that is better than or equal to other known scheduling schemes for all decoding iterations. Since the scheduling schemes change adaptively through exploiting the advantageous properties of the respective schemes, we will call our scheme *adaptive informed dynamic scheduling*. We note that additional computational cost incurred in our scheme relative to RBP or NWRBP, whose complexity is essentially equal to that of the flooding scheme [10], is linear in the length of the code as mentioned in [7] and [8]. A pseudo-code of our proposed algorithm is as follows:

Algorithm 1 Adaptive Informed Dynamic Scheduling

1: Begin with RBP and switch to LBP if $\frac{S_{l-1}}{S_{l-2}} \geq \frac{S_l}{S_{l-1}}$ is satisfied twice for $l \geq 4$, or a predetermined number of iterations is reached whichever comes earlier.

2: From LBP, switch to NWRBP if $\sum_{i=1}^{n} \frac{|\text{LLR}(i)|}{\text{deg}(i)}$ hits the second local maximum, or a predetermined number of iterations is reached whichever comes earlier. (n is the number of variable nodes.)

IV. Results

In this section, we present simulation results of flooding, RBP, NWRBP, two-staged mixed scheduling of [2] (2S-LBP+NWRBP), and our proposed scheduling scheme (3S-RLN) for a rate-$\frac{1}{2}$, length 500 code generated from the progressive-edge growth algorithm with an optimized degree distribution used over the AWGN channel. Fig. 1 shows the WER curves for the five schemes with respect to SNRs after 70 decoding iterations. For the 2S-LBP+NWRBP scheme, the

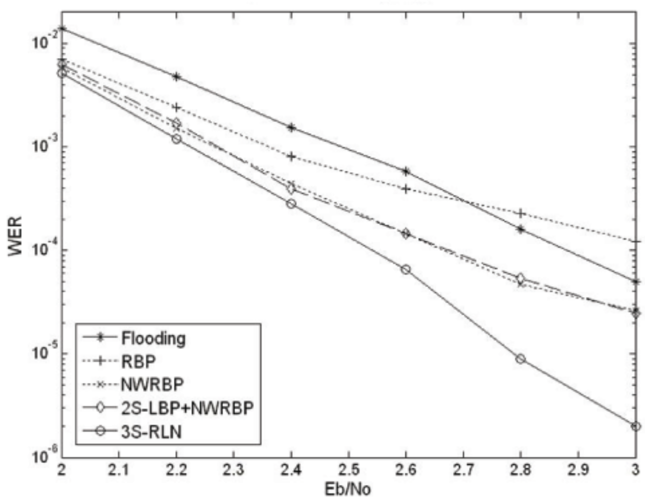

Fig. 1. WER performance of flooding, RBP, NWRBP, LBP+NWRBP and RLN algorithms for $N = 500$, $R = \frac{1}{2}$ code after 70 iterations.

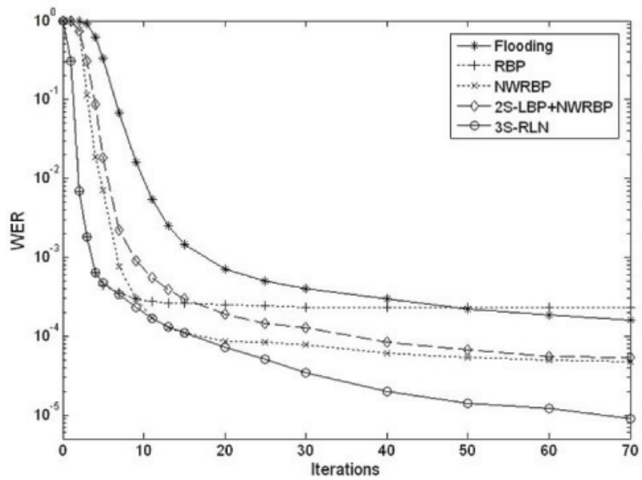

Fig. 2. WER performance of flooding, RBP, NWRBP, LBP+NWRBP and RLN algorithms for $N = 500$, $R = \frac{1}{2}$ code at 2.8dB.

switch from LBP to NWRBP was activated when the number of unsatisfied check nodes was below 11 which was optimized through exhaustive search from 5 to 20, or when iteration number hit 30 whichever came earlier. For our proposed scheme, the predetermined maximum number of iterations before switch was set to 10 and 30 for the first and second stages, respectively. Performance improvement of our adaptive scheme over all other schemes shown becomes increasingly pronounced as SNR grows. This behavior was expected from our construction whose goal includes correcting more trapping set errors, which limit codes' performance at higher SNRs, than other scheduling schemes. We mention that a recently proposed IVC-RBP scheme of [5] also performs very well, however, it seems that the performance differential between our proposed scheme and NWRBP is slightly larger than that of IVC-RBP and NWRBP.

Fig. 2 shows the performance curves with respect to decoding iterations at 2.8dB. The figure shows that our proposed scheme performs better than or equal to all other scheduling schemes for all decoding iterations. In particular, note that our proposed scheme performs increasingly better as iteration progresses which results from correcting more trapping set errors without incurring many non-trapping set errors relative to other scheduling schemes. For example, we mention that our proposed scheme outperforms the mixed scheduling scheme of [8] for all decoding iterations precisely for this reason. We also mention that the undetected WER of our proposed scheme was sufficiently lower than the (detected) WER shown in the two figures for the range of SNRs and iterations considered.

V. Conclusion

In this letter, an adaptive informed dynamic scheduling consisting of three sequential scheduling schemes that can correct both non-trapping and trapping set errors was presented. It was shown that the presented scheme provides performance that is superior to that of other known iterative decoding algorithms at every decoding iteration observed.

References

[1] A. Casado, M. Griot, and R. D. Wesel, "Informed dynamic scheduling for belief-propagation decoding of LDPC codes," *2007 IEEE International Conference on Communications.*

[2] A. Casado, M. Griot, and R. D. Wesel, "Improving LDPC decoders via informed dynamic scheduling," *2007 IEEE Information Theory Workshop.*

[3] A. Casado, M. Griot, and R. D. Wesel, "LDPC decoders with informed dynamic scheduling," *IEEE Trans. Commun.*, vol. 58, no 12, pp. 3470–3479, Dec. 2010.

[4] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: informed scheduling for asynchronous message passing," *Proc. 2006 Conference on Uncertainty in Artificial Intelligence.*

[5] Y. Gong, X. Liu, W. Ye, and G. Han, "Effective informed dynamic scheduling for belief propagation decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 59, no. 10, pp. 2683–2691, Oct. 2011.

[6] D. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. 2004 Signal Processing Systems*, pp. 107–112.

[7] S. Kim, K. Ko, J. Heo, and J.-H. Kim, "Two-staged informed dynamic scheduling for sequential belief propagation decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 13, no. 3, pp. 193–195, Mar. 2009.

[8] S. Kim, S. Lee, J. Heo, and C. S. Rim, "Adaptive mixed scheduling for correcting errors in trapping sets of LDPC codes," *IEEE Commun. Lett.*, vol. 14, no. 7, pp. 664–666, July 2010.

[9] H. Kfir and I. Kanter, "Parallel versus sequential updating for belief propagation decoding," *Physica A*, vol. 330, pp. 259–270, Sep. 2003.

[10] P. Radosavljevic, A. de Baynast, and J. R. Cavallaro, "Optimized message passing schedules for LDPC decoding," *Proc. 2005 Asilomar Conf. on Signals, Systems and Computers.*

[11] T. Richardson, "Error floors of LDPC codes," *Proc. 2003 Allerton Conf. on Communication, Control, and Computing.*

[12] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," *IEEE Trans. Inf. Theory,* vol. 53, no. 11, pp. 4076–4091, Nov. 2007.