Two-Staged Informed Dynamic Scheduling for Sequential Belief Propagation Decoding of LDPC Codes

Saejoon Kim, *Member, IEEE*, Karam Ko, Jun Heo, *Member, IEEE*, and Ji-Hwan Kim

***Abstract*—Recent studies have shown that sequential belief propagation decoding of low-density parity-check codes can increase the decoding convergence speed while simultaneously improving the asymptotic performance compared to the conventional flooding scheme. Two of the practical sequential decoding schemes known are the ones by Casado *et al.* [1] in which *informed dynamic scheduling* is used for scheduling the sequential updates of the messages. In this letter, we propose a *two-staged* informed dynamic scheduling that unifies and outperforms the two schemes of [1].**

***Index Terms*—LDPC codes, belief propagation decoding, sequential scheduling.**

I. Introduction

LOW-DENSITY parity-check (LDPC) codes have been a main research topic in coding theory recently due to their capacity-approaching performance over various channels using low-complexity iterative decoding. In the conventional iterative decoding scheme which is also called the flooding scheduling scheme, all the variable nodes and subsequently all the check nodes send messages to their neighbors. Recent studies, on the other hand, have shown that it may be more efficient in terms of number of iterations required to achieve a certain performance if the order of sending messages is serial, *i.e.*, serial updates of variable and check node's messages. For example in [6], it was proven that serial updates of messages can require half as many number of iterations to converge compared to the flooding scheme for long codes. Numerous simulations in the literature including those in [1] [4] and references therein have also indicated that sequential decoding converges roughly twice as fast compared to the conventional flooding technique. Sequential updates of the messages then presents us with the problem of efficient scheduling of the messages that provides the best possible convergence speed.

To this end, Casado *et al.* [1] have recently presented two informed dynamic scheduling (IDS) schemes in which the notion of *residual* is utilized to schedule the message updates. It was demonstrated that, compared to any known belief propagation decoding of roughly the same complexity as the flooding scheme, one of the two schemes called *residual belief propagation* (RBP) yields the best code performance for a small number of iterations while the other of the two called *node-wise* residual belief propagation (NWRBP) yields the best code performance as the number of iterations increases.

In this letter, we propose a *two-staged* IDS that combines RBP and NWRBP through exploiting only the desirable features of the respective schemes. Specifically, our proposed scheme will yield code performance that is at least as good as the better of RBP and NWRBP schemes at each decoding iteration. We will in fact show, empirically, that our proposed scheduling scheme outperforms the two schemes starting at a moderate number of iterations until decoding convergence.

Manuscript received November 17, 2008. The associate editor coordinating the review of this letter and approving it for publication was V. Stankovic.

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

J. Heo is with the School of Electrical Engineering, Korea University, Seoul, Korea.

The work of S. Kim was supported by the Special Research Grant of Sogang University 200811028.01. The work of J. Heo was supported by the IT R&D program of MKE/IITA [2008-F-006-01, Wireless Local Area Communication Systems on Tera Hertz Band]. The work of J.-H. Kim was supported by the Sogang University Research Grant of 200701117.01.

Digital Object Identifier 10.1109/LCOMM.2009.081938

II. Informed Dynamic Scheduling

Consider the bipartite graph of an LDPC code with the set of variable nodes V and the set of check nodes C. In a belief propagation decoding of LDPC codes, the log-likelihood ratio (LLR) message from a variable node $x \in V$ to a check node $c \in C$, m_{xc}, is defined as

$$m_{xc} = m_x + \sum_{c' \in \mathcal{N}(x)\backslash c} m_{c'x} \tag{1}$$

and the message from $c \in C$ to $x \in V$, m_{cx}, is defined as

$$m_{cx} = \log \frac{1 + \prod_{x' \in \mathcal{N}(c)\backslash x} \tanh(\frac{m_{x'c}}{2})}{1 - \prod_{x' \in \mathcal{N}(c)\backslash x} \tanh(\frac{m_{x'c}}{2})} \tag{2}$$

where m_x is the initial LLR of the variable node x from the channel information, $\mathcal{N}(\cdot)$ is the set of neighbors of $\cdot$, and $m_{c'x}$'s in Equation (1) are all initially equal to 0. In each decoding iteration of the conventional flooding scheme, all messages m_{xc} and m_{cx}, $x \in V,\ c \in C$, are updated according to Equations (1) and (2), respectively.

Sequential belief propagation decoding generally refers to updating messages in a serial order and can be classified broadly into two classes: nonadaptive scheduling and adaptive scheduling. Nonadaptive scheduling techniques examples of which are described in [6] and references therein update messages in a predetermined fixed order whereas the order of message updates is changed dynamically according to the messages generated during the decoding process in adaptive scheduling techniques. Motivated by the recent results of [1], we consider in this letter an *informed dynamic scheduling* which is an adaptive scheduling technique. In the next subsection, we review two examples of IDS from [1] that perform very well.

A. Informed Dynamic Scheduling using Residual

RBP is an algorithm that schedules the message updates according to the value of the *residual* of message m which, originally introduced in [2], is defined as

$$r(m) \triangleq \|m - m'\|$$

where m' is the message before an update. The idea of RBP is that only the message with the largest residual is selected for each update in the hope of increasing the decoding convergence speed. Initially, the residuals of all messages in the bipartite graph are set to 0. Then all the variable-to-check nodes 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. In both RBP and NWRBP, after sending or updating a message, the associated residual for that message is reset to 0. Note that after sending this check-to-variable node message, only the residuals of variable-to-check node messages $m_{x'c}$ where $c \in \mathcal{N}(x')\backslash c'$ change and hence only these $m_{x'c}$'s are updated. Then m_{cx}'s are calculated where $c \in \mathcal{N}(x')\backslash c'$ and $x \in \mathcal{N}(c)\backslash x'$, and among all the residuals of check-to-variable node messages, the message with the largest residual is updated. RBP consists of repeated application of this process of sequential scheduling. As explained in [1], RBP is a greedy method which performs well for a small number of decoding iterations however worse than the flooding scheme for increasing number of iterations.

To resolve this problem of early decoding convergence in RBP, a less greedy version of it called NWRBP was introduced in which the only difference from RBP is that instead of the check-to-variable node message with the largest residual being updated, all check-to-variable messages are updated for the check node that contains the message with the largest residual. In other words, if the message $m_{c'x'}$ has the largest residual from some $x' \in V$ and $c' \in C$, then messages $m_{c'x}$, $x \in \mathcal{N}(c')$, are all updated simultaneously in NWRBP. In both RBP and NWRBP, decoding stops if either all the parity-check equations are satisfied after an iteration or a predetermined number of iterations has reached. It was demonstrated that NWRBP has better convergence speed and asymptotic performance than the flooding scheme while it performs slightly worse than RBP for a small number of iterations.

B. Two-Staged Informed Dynamic Scheduling

In the simulation results shown in [1], it was observed that RBP and NWRBP yield a competitive decoding performance using belief propagation decoding for a small and mid-to-large number of iterations, respectively. Motivated by this observation, our proposed scheduling scheme will work in two stages: RBP in the first stage and NWRBP in the second stage. The main contribution of this letter is then the formulation of the condition to switch from the first stage to the second stage.

To this end, consider the set of unsatisfied check nodes which is defined as the check nodes whose neighbors (after hard-decision decoding) do not sum to 0 modulo 2. Then define the set of variable nodes S that are neighbors to the set of unsatisfied check nodes as *suspicious* nodes [7] and denote by S_l the number of suspicious nodes after decoding iteration l. Clearly, the set of suspicious nodes will include variable nodes in error with high probability. Simulation results of RBP, as will be observed in the next section, indicate that the average cardinality of S, when not equal to 0, drops significantly from S_0 to S_1 and then starts to increase until convergence while the word-error-rate (WER) decreases until convergence. We can deduce then that progressively less frequent occurrence of nonempty but larger-sized S for increasing number of iterations is reflective of lower WER. Simulation results also indicate that the cardinality of S converges slightly faster than WER which can be exploited to formulate the condition for the switch from RBP to NWRBP. Therefore in our proposed scheduling algorithm, the switch from the first stage to the second stage takes place immediately after decoding iteration l for $l \geq 4$ if the following condition is satisfied twice, in any order:

$$\frac{S_{l-1}}{S_{l-2}} \geq \frac{S_l}{S_{l-1}}. \qquad (3)$$

It seemed that the requirement of the inequality (3) being satisfied *twice* was appropriate to detect the start of convergence of the cardinality of S at the earliest possible iteration number without mistakingly interpreting an incidental decrease in the relative change of S as the start of its convergence. Since decoding stops after l iterations in RBP (and also in NWRBP) when $S_l = 0$, decoding stops if any of the S_l's in inequality (3) becomes 0. For example, the switch never takes place and decoding stops if $S_l = 0$ for $l \leq 4$. Besides when S_l becomes 0 before the switch condition is satisfied, the only way for the switch not to occur is if the inequality (3) is satisfied at most once for all values of l which is an unlikely event. For this matter, one can modify the switch condition so that RBP is switched to NWRBP if the described condition is satisfied or a predetermined number of iterations is reached whichever comes earlier, however, simulation results indicated that this was not necessary. When the switch does take place, to facilitate faster convergence, NWRBP is not started with its associated initial condition of residuals of all messages being equal to 0, but it starts with the residuals inherited from RBP. Note that the only additional computational cost of our algorithm compared to RBP or NWRBP, whose computational complexity is essentially equal to that of the flooding scheme [5], is the calculation of S_l for each $l \geq 1$ which takes only a linear amount of time.

A decoding iteration in RBP and NWRBP is defined as the number of check-to-variable node message updates that equals the number of edges in the bipartite graph and as the number of check-node message updates that equals the number of check nodes, respectively [1]. Consequently, in RBP and NWRBP, a *fraction* of a decoding iteration can be defined. Thus while it is possible to let the inequality (3) to take on non-integer values for the iteration number l, only integer values of l were considered in this letter for computational cost savings.

III. Results

In this section, we show simulation results of belief propagation decoding of BPSK-modulated LDPC codes used over the additive white Gaussian noise (AWGN) channel. In all simulations, we used a rate-$\frac{1}{2}$, length 500 code that was generated from the progressive-edge growth algorithm [3] using an optimal degree distribution.

Figure 1 shows the mean of S_l, $l = 0, 1, \cdots, 10$, for $S_l > 0$ of the flooding scheme, RBP and NWRBP. Not shown in the

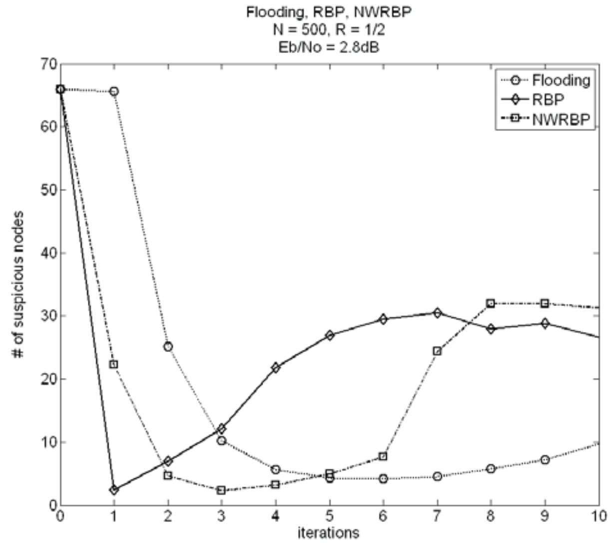

Fig. 1. Average number of suspicious nodes.

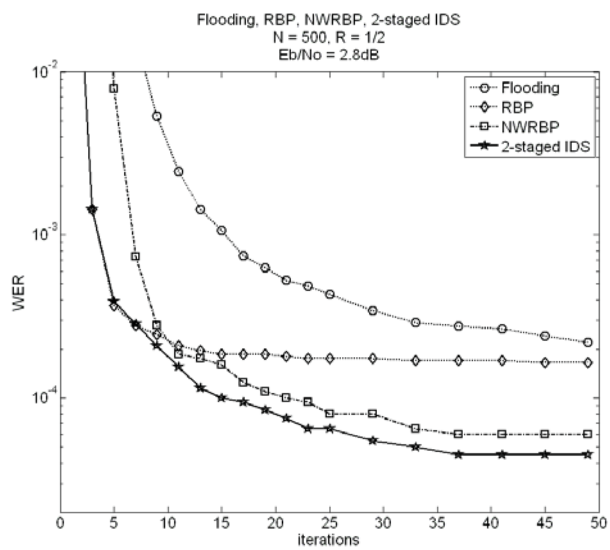

Fig. 2. WER performance of scheduling algorithms.

figure are the nonconvergent behaviors of S_l, $l > 10$, of the flooding scheme and NWRBP while S_l for RBP has nearly converged after iteration 5.

Figure 2 shows the WER performance of the three decoding schemes mentioned in Fig. 1 along with our proposed two-staged IDS scheme. Note that RBP and NWRBP perform noticeably better than the flooding scheme while RBP converges very quickly approximately at iteration 10 well after its associated S_l converges. Also note that NWRBP starts to outperform RBP at iteration 10 and converges much later at approximately iteration 35. Two-staged IDS scheme performs almost the same as RBP for a small number of iterations of up to around 7 and outperforms both RBP and NWRBP thereafter. The switch from the first stage to the second stage occured after 5.1 iterations on average with variance of 1.1. The fact that the second stage of our two-staged IDS scheme starts with fewer nodes in error than NWRBP establishes the asymptotically superior performance of the former over the latter. It should be emphasized that the timing of the switch

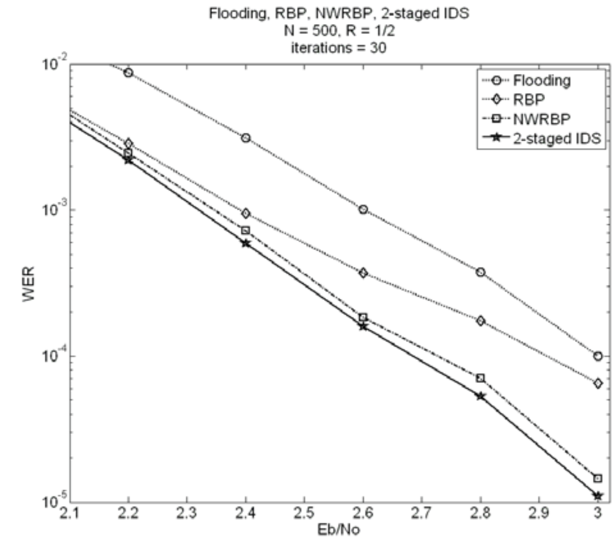

Fig. 3. WER performance after 30 iterations.

from the first to the second stage is crucial since RBP can outperform the two-staged IDS scheme in the early iterations if the switch is made too early. If the switch is made too late then the LLR's of the new variable nodes in error that were generated from RBP will be too large to be corrected by NWRBP.

Figure 3 shows the WER performance of the four decoding schemes after 30 iterations. In agreement with the results suggested in Fig. 2, the relative performances of the four schemes remain the same over a range of signal-to-noise ratios.

IV. Discussion

In this letter, a two-staged IDS for sequential belief propagation decoding of LDPC codes used over the AWGN channel was presented. It was shown that the presented scheduling algorithm combines and improves on the recently introduced RBP and NWRBP schemes at the expense of little amount of additional complexity.

References

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

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

[3] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386-398, Jan. 2005.

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

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

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

[7] N. Varnica, M. P. C. Fossorier, and A. Kavcic, "Augmented belief propagation decoding of low-density parity-check codes," *IEEE Trans. Commun.*, vol. 55, no. 7, pp. 1308-1317, July 2007.