Improving Belief Propagation on Graphs With Cycles

Mohammad R. Yazdani, *Student Member, IEEE*, Saied Hemati, *Student Member, IEEE*, and
Amir H. Banihashemi, *Associate Member, IEEE*

***Abstract*—In this letter, we propose two modifications to belief propagation (BP) decoding algorithm. The modifications are based on reducing the reliability of messages throughout the iteration process, and are particularly effective for short low-density parity-check codes, where the existence of cycles makes the original BP algorithm perform suboptimal. The proposed algorithms, referred to as "normalized BP" and "offset BP," reduce the absolute value of the outgoing log-likelihood ratio messages at variable nodes by using a multiplicative factor and an additive factor, respectively. Simulation results show that both algorithms perform more or less the same, and both outperform BP in error performance.**

***Index Terms*—Belief propagation, iterative decoding, low-density parity-check (LDPC) codes, message-passing decoding algorithms, normalized belief propagation, offset belief propagation.**

I. Introduction

GALLAGER proposed low-density parity-check (LDPC) codes in his thesis, along with several message-passing decoding algorithms [1], among which belief propagation (BP) algorithm is known to have the best performance. The remarkable performance of LDPC codes with message-passing decoding has positioned them as strong candidates for error-correction in many digital communication systems. As a linear block code, an LDPC code can be represented by a Tanner graph (TG) [2]. A TG is a bipartite graph in which one set of nodes, the variable nodes, corresponds to code symbols and the other set of nodes, the check nodes, corresponds to the set of parity-check constraints which define the code. An edge exists between a variable node v and a check node c if and only if v appears in the parity-check equation corresponding to c.

Given a TG for an LDPC code, iterative implementation of BP, which proceeds as if no cycles were present in the graph, has been shown to deliver impressive results. In fact, it is well-known that if the TG is cycle-free then the BP converges to a posteriori probabilities for variable nodes [2]. In many applications, however, LDPC codes have short to intermediate lengths (a few hundred to a few thousand bits) and the assumption of cycle-free graph is not valid. Consequently, in such cases, there is no guarantee that BP is optimal. Although, the nonoptimality of BP on graphs with cycles is well-known, only a few modified versions of BP that can outperform the standard BP algorithm have been introduced [3]–[5]. In [3], a multistage iterative decoding algorithm that combines BP with ordered statistic decoding has been used to bridge the error performance gap between BP and maximum likelihood decoding. The computational complexity however is considerably higher than that of BP. In [4], the authors devise a "probabilistic schedule" for message passing between variable nodes and check nodes in the TG, while keeping the operations performed in variable and check nodes the same as those of standard BP. Although the method of [4] has more or less the same total number of computations as standard BP, the implementation of schedule to control the flow of messages is the extra complexity associated with this method. In [5], the authors present "generalized belief propagation (GBP)" algorithms that work based on the communication among certain regions of nodes in the graph. Although GBP algorithms can outperform standard BP for a proper choice of regions, no systematic approach is yet known for properly choosing the regions, particularly for graphs with a large number of nodes. For more references on work related to GBP algorithms, see [5].

In this letter, we introduce two very simple modifications to BP algorithm, which provide nonnegligible improvement in the performance of BP, particularly at short and intermediate block lengths. In Section II, we explain how the reliabilities in BP are over-estimated for graphs with cycles. In Section III, the proposed modifications are introduced. Sections IV and V contain the simulation results, and conclusions, respectively.

II. Overestimation of Reliabilities in Belief Propagation

We consider the transmission of code words over a binary input additive white Gaussian noise (AWGN) channel, where the input symbols are ± 1. The channel output is processed by an iterative BP decoder implemented in log-likelihood ratio (LLR) domain which operates based on the so-called "flooding" or "parallel" schedule [2], [4]. Moreover, we assume that only extrinsic information is processed in both variable and check nodes, i.e., each outgoing message along an edge e is only a function of input messages coming to the node along edges other than e. This guarantees that for a cycle-free TG, incoming and outgoing messages through each edge are independent and at the end, the algorithm produces correct marginal *a posteriori* probabilities. However, for a graph with cycles, throughout the iteration process, dependency is created among the incoming messages to a node and/or among the incoming messages to a variable node on one side, and the initial message of the node, on the other side. In this case, BP algorithm will not be optimal anymore. This implies that, in a graph with cycles, the outgoing messages for BP, which would have been optimal for

Manuscript received June 2, 2003. The associate editor coordinating the review of this letter and approving it for publication was Dr. M. Fossorier. This work has been supported in part by National Capital Institute of Telecommunications (NCIT), Ottawa, ON, Canada.

The authors are with Broadband Communications and Wireless Systems (BCWS) Centre, Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: ryazdani@sce.carleton.ca; hemati@sce.carleton.ca; ahashemi@sce.carleton.ca).

Digital Object Identifier 10.1109/LCOMM.2003.822499

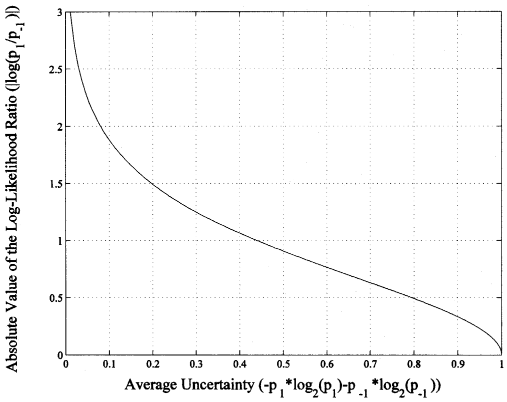

Fig. 1. The absolute value of the log-likelihood ratio is a monotonically decreasing function of the average uncertainty.

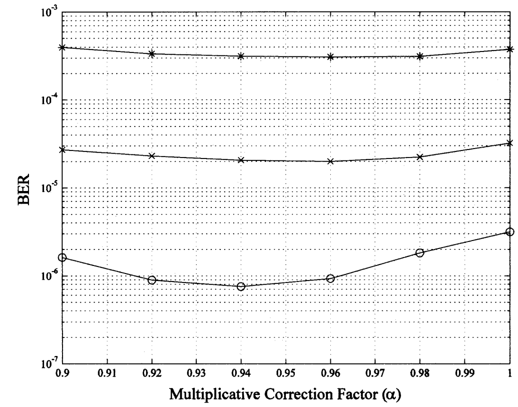

Fig. 2. Effect of α on BER at E_b/N_0's 1.7(*), 2(x), and 2.3 (o) for (1268, 456) LDPC code decoded by normalized BP.

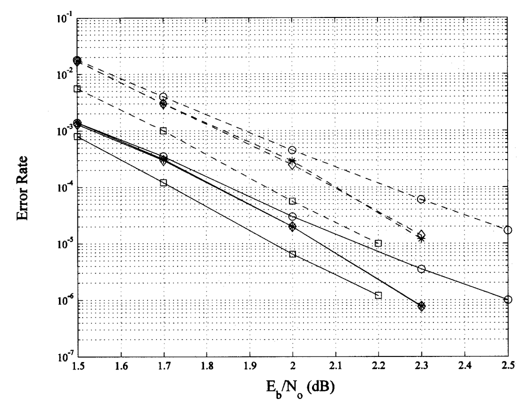

Fig. 3. BER(—) and WER(- -) for (1268, 456) LDPC code decoded by BP(o), algorithm of [3] (□), probabilistic scheduling with BP(◇), and normalized BP with $\alpha = 0.94(*)$.

a cycle-free graph, now have mistakenly a higher reliability, or equivalently a lower uncertainty, on average. (Fig. 1 shows that the magnitude of LLRs, as the reliability of messages, is a monotonically decreasing function of the average uncertainty.) In this work, we consider two remedies for the overestimation of reliabilities, as explained in the following section.

III. Normalized and Offset Belief Propagation

In *normalized BP*,[1] the outgoing message $m_{v\to c}$ from a variable node v to a check node c is replaced by

$$m'_{v\to c} = \alpha . m_{v\to c} \tag{1}$$

where α is a positive number less than or equal to one, called the *multiplicative correction factor*.[2] In offset BP however, the correction factor is additive, and is applied by replacing the magnitude of the outputs of variable nodes in BP by

$$|m'_{v\to c}| = \begin{cases} |m_{v\to c}| - \beta & |m_{v\to c}| > \beta \\ |m_{v\to c}| & |m_{v\to c}| \leq \beta \end{cases} \tag{2}$$

where β is a nonnegative number, called *additive correction factor* (signs remain unchanged). The optimal values for correction factors α and β are functions of E_b/N_0, where E_b is the average energy per information bit and N_0 is the one-sided power spectral density of AWGN. In general, optimal correction factors also depend on the iteration number, and the node pair (v, c). In this paper, for the sake of simplicity, we assume that α and β remain constant during the iteration process and are the same for all pairs of variable and check nodes.

IV. Simulation Results

To investigate the performance and complexity of modified BP algorithms, simulation results for an optimized irregular LDPC code with parameters $(n, k) = (1268, 456)$ [4] are presented. For all simulation results, the maximum number of iterations is chosen to be 500. Also, for each E_b/N_0, enough code words are simulated to generate at least 100 code word errors. Fig. 2 shows bit error rate (BER) curves of normalized BP versus α for different values of E_b/N_0. It can be seen that the optimal amount of correction required at each E_b/N_0 increases (optimal α decreases) with E_b/N_0. This can be justified by Fig. 1, as increasing E_b/N_0 corresponds to less average uncertainty, which in turn means moving to the left on Fig. 1, where the curve is steeper. This is a region where a slight error in message uncertainty due to using BP algorithm, instead of a presumably optimal algorithm, can considerably increase the magnitude of LLRs. From Fig. 2, one can also observe that at lower values of E_b/N_0, BER is less sensitive to α, and the optimal value of α is not very sensitive to E_b/N_0. Simulation results for offset BP show the same trend for the optimal value of β. In fact, for the given code, the optimal value of β is approximately 0.2 for a wide range of E_b/N_0 values. Moreover, we observe that, when optimized, normalized and offset BP perform more or less the same, with normalized BP performing slightly better at high $\mathrm{E_b/N_0}$'s. It is also worth mentioning that the average number of iterations required for convergence at each E_b/N_0 is more or less the same for both algorithms and is just slightly smaller than that of standard BP.

In Fig. 3, we have BER and word error rate (WER) curves of normalized BP with $\alpha = 0.94$ along with the curves for standard

[1]The same nomenclature is used in [6] to describe similar modifications at the output of check nodes in the so-called min-sum algorithm. The offset operation presented here is however slightly different than the one in [6] and provides better results for BP.

[2]It was brought to our attention by the reviewers that the observation that the performance of BP can be enhanced by scaling down the LLRs in the iteration process was also made by M. Tanner as reported in [7] and [8].

BP. The decoders are set to work in parallel and perform on the same set of received vectors. In the same figure, we have also shown the BER and WER curves of the algorithm in [3] from [9] ($\mathrm{BP} + \mathrm{order} - 1$ with maximum of 50 iterations) as well as the curves we obtained for BP with probabilistic scheduling. We observe that BP with probabilistic scheduling performs more or less the same as our proposed algorithm. It however has a higher complexity. The algorithm of [3] outperforms our algorithm with the gap closing down at higher values of E_b/N_0.[3] The complexity of our algorithm however is much less. It should be noted that in many cases the algorithms of [3] outperform normalized and offset BP by a large margin. The improvement obtained by the latter over standard BP is normally up to at most 0.3 dB, while the former can result in improvements of more than 1 dB [3].

V. Conclusions

For short and moderate block lengths, due to the existence of short cycles in the TG of LDPC codes, the messages passed through the edges of the graph in BP algorithm are statistically dependent. This implies that the "real" reliability of these messages is smaller compared to what is derived by BP under the assumption of cycle-free graph. In this letter, we have introduced two modified versions of BP algorithm, normalized and offset BP, in which the reliability overestimation of the messages is compensated for by using a multiplicative and an additive correction factor, respectively. Simulation results show that both algorithms, when optimized, perform more or less the same, and they both provide nonnegligible improvement in error performance over standard BP.

As a practical note, it is worth mentioning that normalized BP with a fixed correction factor can be easily implemented (with no additional transistors) by transistor scaling in current conveyers used in analog implementation of BP [10].

[3]Note that the real performance difference between the two algorithms seems to be even less than what is reported here, as the curves for standard BP obtained from [9] appear to outperform our BP curves in Fig. 3.

Acknowledgment

The first two authors would like to thank F. Zarkeshvari for providing them with the simulation program of BP algorithm.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[2] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.

[3] M. Fossorier, "Iterative reliability-based decoding of low-density parity check code," *IEEE J. Select. Areas Communications*, vol. 19, pp. 908–917, May 2001.

[4] Y. Mao and A. H. Banihashemi, "Decoding low-density parity-check codes with probabilistic schedule," *IEEE Commun. Lett.*, vol. 5, pp. 414–416, Oct. 2001.

[5] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," IEEE Trans. Inform. Theory, submitted for publication.

[6] J. Chen and M. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, pp. 208–210, May 2002.

[7] D. MacKay and M. Davey, "Evaluation of Gallager codes for short block length and high rate applications," in *IMA Workshop on Codes, Systems and Graphical Models*, 1999, pp. ???–???.

[8] R. Lucas, M. Fossorier, Y. Kou, and S. Lin, "Iterative decoding of one-step majority logic decodable codes based on belief propagation," *IEEE Trans. Commun.*, vol. 48, pp. 931–937, June 2000.

[9] M. Fossorier, private communication, May 2003.

[10] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkoy, "Probability propagation and decoding in analog VLSI," *IEEE Trans. Inform. Theory*, vol. 47, pp. 837–843, Feb. 2001.