



A Modified Offset Min-Sum Decoding Algorithm for LDPC Codes

(National ASIC System Engineering Center, Southeast University, Nanjing, 210096)

Abstract: In this paper, a modified Offset Min-Sum decoding algorithm for Low-Density Parity Check Codes is presented. In this modified algorithm, the offset factor in the original Offset Min-Sum algorithm is adjusted iteratively with the assistance of the check-node computation outputs of the Normalized Min-Sum algorithm. As a result, the offset factor in our modified algorithm can be calculated adaptively and more efficiently according to the previous decoding procedure, which can minimize the decoding degradation compared with the Belief Propagation decoding algorithm. The simulation results show that, compared with the original Offset Min-Sum decoding algorithm, our modified algorithm can achieve noticeable performance improvement with minor extra hardware complexity. For example, when BER is 10^{-5} , our algorithm can achieve 0.1dB and 0.2dB decoding gain over Offset Min-Sum algorithm for regular and irregular LDPC codes respectively.

Key words: Belief Propagation algorithm; LDPC codes; Normalized and Offset Min-Sum algorithm

1 Introduction

Low-Density Parity Check (LDPC) codes were first proposed by Gallager in 1962^[1]. After the discovery of turbo code, D. J. C. Mackay, M. Neal and N. Wiberg^[2-3] reinvestigated LDPC codes and found it can achieve good Bit Error Rate (BER) performance approaching Shannon limit.

The main decoding algorithm for LDPC codes is called the Belief Propagation (BP) algorithm. But the BP algorithm involves complex check-node computation and is difficult for hardware implementation. Then Fossorier introduced the Min-Sum (MS) algorithm by simplifying the check-node computation in the BP algorithm^[4]. Though the MS algorithm can reduce the computational complexity, the decoding performance has been sacrificed too much. Then much work has been done to modify the MS algorithm to achieve

better performance^[5-6]. Chen then introduced the Normalized Min-Sum (NMS) and Offset Min-Sum (OMS) algorithms^[7]. In these two algorithms, the normalized factor and offset factor are applied to the check-node update equation to improve the decoding performance. But as mentioned in Res.^[8], the normalized factor is determined by the magnitude of the minimum value and the decoding performance suffers severe degradation when the output is close to zero. The offset factor is set before decoding and does not take the output value of each iteration output into account, so the decoding performance is under optimization. In Res.^[8], the authors introduce the modified OMS algorithm to solve the above problems and achieve some improvements.

In this paper, we introduce a modified Offset Min-Sum decoding algorithm (MOMS) to further improve the decoding performance. In our modified algorithm the offset factor is adjusted iteratively taking account of the previous check-node computation outputs. This adaptive scheme can solve the problems of the NMS and OMS algorithms and achieve better decoding performance. At the same time, our modified algorithm requires no extra increase in hardware complexity.

Authors' Introduction:

2 BP and BP-based Algorithms

LDPC codes are linear block codes described by a sparse $M \times N$ parity-check matrix H . For LDPC codes based on Tanner graph shown in Fig. 1 using the BP algorithm, the key decoding strategy is the application of Bayes' rule at each node and the exchange of the messages with neighboring nodes. For each iteration step, two types of messages are passed: probability or "beliefs" from variable nodes to check nodes, and probability or "beliefs" from check nodes to variable nodes. In this section, all the notations are similar to that in Res.[2].

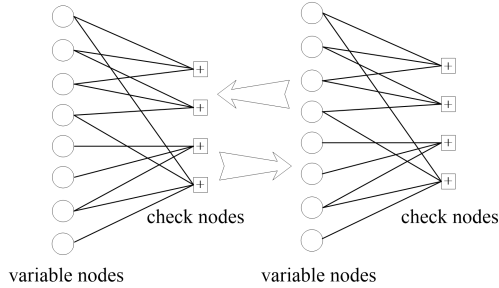


Fig.1 Message Passing in BP Algorithm

Define that $M(n)$ is the set of check nodes which are connected to variable node n , and $N(m)$ is the set of variable nodes which are connected to check node m . In the probability domain, the inputs to the BP decoding algorithm are the posteriori probabilities based on the channel statistics. And it is defined as below in Eq.(1):

$$\Lambda_c(c_n) = \log \frac{P(c_n = 1 | y_n)}{P(c_n = 0 | y_n)} = L_c y_n \quad (1)$$

Where $L_c = -4\sqrt{E_s}/N_0$ on Additive White Gaussian Noise (AWGN) channel, c_n is the code transmitted into the channel and y_n is the code received after transmission through the channel.

The check-node update equation is defined as below in Eq.(2):

$$L_{n \leftarrow m}^{(k)}(c_n) = 2 \arctan h \left(\prod_{u \in N(m) - (n)} \tanh \frac{\Lambda_{u \rightarrow m}^{(k)}(c_u)}{2} \right) \quad (2)$$

In the MS algorithm, Eq.(3) below is used to

replace the complex equation in Eq.(2) to simplify the computation:

$$L_{n \leftarrow m}^{(k)}(c_n) = \prod_{u \in N(m) - (n)} \text{sign}(\Lambda_{u \rightarrow m}^{(k)}(c_u)) \min_{u \in N(m) - (n)} \left(|\Lambda_{u \rightarrow m}^{(k)}(c_u)| \right) \quad (3)$$

Define Eq.(3) in the MS algorithm as L_2 and the corresponding Eq.(2) in the BP decoding algorithm as L_1 . It is easy to prove that the sign of L_2 is the same as that of L_1 , but the absolute value of L_1 is smaller than L_2 . In other words, $|L_1| < |L_2|$, and that can explain why the MS algorithm suffers a degradation of decoding performance. Then the decoding performance can be improved by reducing $|L_2|$ and modifying it equal to $|L_1|$. That strategy leads to the NMS and OMS algorithms. These two algorithms use Eq.(4) and Eq.(5) below to modify $|L_2|$ respectively.

$$L_N = \alpha L_2 \quad (4)$$

Where $0 < \alpha < 1$, and $1/\alpha$ is called the normalized factor.

$$L_0 = \text{sign}(|L_2|) \max(|L_2| - \beta, 0) \quad (5)$$

Where $\beta > 0$, and it is called the offset factor. Obviously, in order to achieve best decoding performance, $|L_2|$ should be equal to $|L_1|$. And that requires $\alpha = E[L_1]/E[L_2]$ and $\beta = E[L_1] - E[L_2]$.

3 Modified Offset MS Algorithm

But as mentioned in Res.[8], both the NMS and OMS algorithms have their own drawbacks. For the NMS algorithm, the normalized factor $1/\alpha$ can efficiently modify the check-node computation equation and improve the decoding performance. But the decoding process is entirely determined by the magnitude of the minimum output value. And when the output is nearly zero, the decoding performance will suffer severe degradation. And for the OMS algorithm, the offset factor β is constant during decoding procedure and does not change adaptively according to the output value. As a result the OMS algorithm can only achieve limited improvement against the MS algorithm.

To solve these problems, we propose a MOMS



algorithm which uses check-node computation output values from the NMS algorithm to modify the offset factor in each iterative decoding step. The idea of our MOMS algorithm is demonstrated as follows:

First, we introduce a modified NMS algorithm based on minimum mean square error rule. In our algorithm, check-node computation equation is shown below in Eq.(6):

$$L_{mn,MNMS}^{(k)} = \begin{cases} \alpha L_{mn,MS}^{(k)} & |L_{mn,MS}^{(k)}| \geq \frac{1}{P} \sum_{i=k-p}^{k-1} |L_{mn,MS}^{(i)}| \\ L_{mn,MS}^{(k)} & \text{else} \end{cases} \quad (6)$$

Where p is a positive integer and $L_{mn,MS}^{(k)}$ is the k th iterative check-node computation value of the MS algorithm.

Eq.(6) shows that check-node computation output in the MS algorithm is only normalized when its magnitude is big enough. And the threshold is determined by the arithmetic average of previous P check-node computation outputs. Then how to choose the parameter P is a tradeoff between computational complexity and decoding performance. Choosing a bigger P means using more previous check-node computation outputs to adjust the offset factor and that can improve the decoding performance. But meanwhile, this will increase the computational complexity of the algorithm as shown in Table 1. To maintain minimum computational complexity, P is set to be 2 in our experiments.

Second, we can calculate the offset factor β_{mo} . From Eq. 5, β is the difference between output of MS and OMS algorithms. Here we use Eq.(6) to replace the output of the OMS algorithm and get this adaptive offset factor β_{mo} .

$$\beta_{mo} = |L_{mn,MS}^{(k)}| - |L_{mn,MNMS}^{(k)}| \quad (7)$$

Where $L_{mn,MS}^{(k)}$ is the k th iterative check-node update value of the MS algorithm and $L_{mn,MNMS}^{(k)}$ is the

k th iterative check-node update value of the modified NMS algorithm.

Finally, we can get the check-node computation equation for our MOMS in Eq.(8).

$$L_{mn,MOMS}^{(k)} = \text{sign}(L_{mn,MS}^{(k)}) \max(|L_{mn,MS}^{(k)}| - \beta_{mo}, 0) \quad (8)$$

The computational complexity of check-node update equation in the MS algorithm is determined by the number of comparison and addition operations in Eq.(3). Assuming the check-node degree is d_c , $d_c + \lceil \log d_c \rceil - 2$ addition operations are needed. For the NMS and OMS algorithms, two extra multiplication operations or two extra addition operations are needed for the normalized factor $1/\alpha$ or the offset factor β respectively. The algorithm proposed in Res.[8] requires one left shift operation and one look-up table of length 8 against the OMS algorithm. And the size of the look-up table is determined by Res.[9] and stored before the iterative decoding.

Table I lists the check-node update computational complexity of these algorithms[8]. Our MOMS algorithm requires two more multiplication operations and P more addition operations against the OMS algorithm. And compared with the NMS algorithm, our MOMS algorithm only requires $P+2$ more addition operations. When P is a small integer (P is set to be 2 in our project), the extra requirement is neglectable. The comparison between the computational complexity of our MOMS and the algorithm in Res.[8] is dependent on specific parameters[9].

Table1. The computation complexity of check-node equation with MS, OMS, NMS, Res.[8] and MOMS

Algorithm	Multip	Addition	Others
Min-Sum Algorithm	0	$d_c + \lceil \log d_c \rceil - 2$	
OMS Algorithm	0	$d_c + \lceil \log d_c \rceil$	
NMS Algorithm	2	$d_c + \lceil \log d_c \rceil - 2$	
Algorithm in [8]	0	$d_c + \lceil \log d_c \rceil$	*
MOMS Algorithm	2	$d_c + \lceil \log d_c \rceil + p$	

*One left shift operation and one look-up table of length 8

4 Simulation Results

In the simulation we use both regular^[2] and irregular LDPC codes^[10] to test our MOMS algorithm.

First, (1008, 504)^[2] regular LDPC codes with code rate 1/2, row weight 3 and column weight 6 are used in the simulation. The codes are transmitted on AWGN channel after BPSK modulation. The maximum number of iterations is set to be 100. For the OMS algorithm, the optimal value of the offset factor is set to be 0.2.

The performance of MOMS algorithm and several other algorithms (which are labeled as Min-Sum, OBP, algorithm in Res.[8], MOBP and BP algorithms) are shown and compared in Fig. 2. From simulation results we can see, MOMS algorithm has better performance than the MS, OMS and Res.[8] algorithms and slightly worse than the BP algorithm. For example, when BER is 10⁻⁵, MOMS algorithm is 0.4dB, 0.25dB and 0.1dB better than the MS, OMS and Res.[8] algorithms respectively and only 0.1dB worse than the BP algorithm.

Then in the simulation, we use a long irregular LDPC code constructed from Res.[10] with code rate 1/2 and length 8000. Other conditions are the same as the previous simulation. From simulation results in Fig. 3 we can see MOMS algorithm outperforms the MS, OMS and Res.[8] algorithms and is only slightly worse than BP algorithm. For example, when BER is

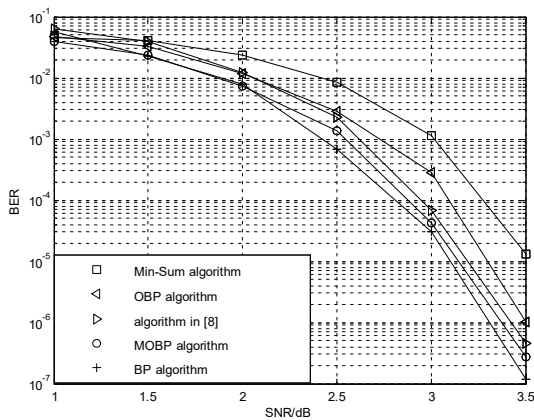


Fig.2 Block error rate of the (1008, 504) regular LDPC code with BP, MOBP, Res.[8], OBP and MS algorithms

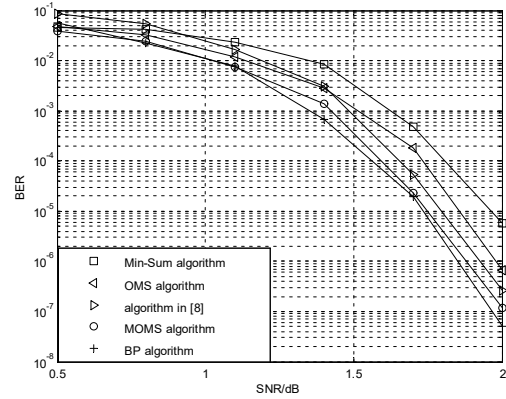


Fig.3 Block error rate of the irregular LDPC code with BP, MOBP, Res.[8], OBP and MS algorithms

10⁻⁵, MOMS algorithm is 0.3dB, 0.2dB and 0.1dB better than the MS, OMS and Res.[8] algorithms respectively and only 0.05dB worse than the BP algorithm.

Then in the simulation, we use a long irregular LDPC code constructed from Res.[10] with code rate 1/2 and length 8000. Other conditions are the same as the previous simulation. From simulation results in Fig. 3 we can see MOMS algorithm outperforms the MS, OMS and Res.[8] algorithms and is only slightly worse than BP algorithm. For example, when BER is 10⁻⁵, MOMS algorithm is 0.3dB, 0.2dB and 0.1dB better than the MS, OMS and Res.[8] algorithms respectively and only 0.05dB worse than the BP algorithm.

From the simulation results in Fig. 2 and Fig. 3, our MOMS algorithm can achieve considerable performance gain over the other algorithms and only requires minor additional computation complexity.

5 Conclusions

This paper proposes a modified Offset Min-Sum decoding algorithm in which the offset factor is adjusted adaptively with the previous check-node computation outputs. Simulation result shows that, for both regular and irregular LDPC codes, our modified algorithm can achieve better decoding performance with only minor increased computational complexity required.



References

- [1] R. G. Gallager. Low-density parity check codes[J]. IRE Trans. on Information Theory, 1962, IT-8: 21-28.
- [2] D. J. C. Mackay. Good error correcting codes based on very sparse matrices[J]. IEEE Trans. on Inform. Theory, 1999, 45: 399-431.
- [3] D. J. C. MacKay and R. M. Neal. Near Shannon limit performance of low-density parity-check codes[J]. Electronics Letter, 1996, 32(18): 1645-1646.
- [4] M. P. C. Fossorier, M. Mihaljevic, and H. Imai. Reduced complexity iterative decoding of low density parity check nodes based on belief propagation[J]. IEEE Trans. on Commun., 1999, 47(5): 673-680.
- [5] E. Eleftheriou, T. Mittelholzer, and A. Dholakia. Reduced-complexity decoding algorithm for low-density parity-check codes[J]. Electronics Letter, 2001, 37(2): 102-104.
- [6] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi. On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes[J]. IEEE Trans. on Commun., 2005, 53(4): 549-554.
- [7] J. Chen and M. P. C. Fossorier. Density evolution for two improved BP-based decoding algorithm of LDPC codes[J]. IEEE Communication Letters, 2002, 6(5): 208-210.
- [8] Ming Jiang, Chunming Zhao, Li Zhang, et al. Adaptive offset min-sum algorithm for low-density parity check codes[J]. IEEE Communication Letters, 2006, 10(6): 483-485.
- [9] S. L. Howard, C. Schlegel, and V. C. Gaudet. A degree-matched check node approximation for LDPC decoding[A]. in Proc. IEEE Int. Symp. on Information Theory[C]. Adelaide, Australia, 2005, 4-9: 1131-1135.
- [10] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes[J]. IEEE Trans. on Inform. Theory, 2001, 47(2): 617-637.