Multibit-flipping decoding algorithm for low-density parity-check codes

Yingying Peng[1*], Xuegang Ren[1], Defa Hu[2]

[1]*Department of Management and Information Engineering, Hunan University of Chinese Medicine, Changsha 410028, Hunan, China*

[2]*School of Computer and Information Engineering, Hunan University of Commerce, Changsha 410205, Hunan, China*

Received 1 June 2014, www.cmnt.lv

Abstract

Aiming at the Low-Density Parity-Check Codes, a reliability-based multibit-flipping decoding algorithm is proposed in the paper. The multibit-flipping criterion is based on the reliable bit position and the threshold in the flipping-decision (number of flipping bits) can be dynamically adjusted during the decoding process. The proposed algorithm is on the basis of the belief propagation decoding algorithm, and then can be derived from its theory. Compared with the traditional weighted bit-flipping decoder and the multi-bit flipping decoder, the proposed decoder can provide a faster converges faster convergent rate and better performances. Simulation results demonstrate that the proposed algorithm achieves a better balance between performance and complexity.

Keywords: LDPC codes, multibit-flipping algorithm, belief propagation algorithm

1 Introduction

LDPC can adopt many different methods to decode, such as the soft-decision, the hard-decision and the hybrid schemes, etc, where the belief-propagation algorithm in the soft-decision and the minimum-sum algorithm can achieve its excellent performance but its operating complexity is rather high. The bit-flipping algorithm in the hard-decision, such as WBF, MWBF, IMWBF and IMWBF, etc, can achieve a better balance between performance and complexity. The general WBF algorithm and its related algorithms just can flip one bit in each iteration. If the algorithms can flip multi bits in each iteration, the above error can be modified and the delaying of the decoding can be reduced. In the references [8-11], different types of the MBF decoding methods are proposed. In the reference [8], each bit can provide a flipping signal counter and each checking node can count the reliability in each bit. If a certain bit reaches to the pre-set threshold, the error probability of the bit is so high that it must be flipped. The method proposed in the reference [8] should be improved in the reference [9].

Aiming to the high-information bit, if the bit can reach to the pre-set threshold and the flipping of the higher reliable information is delayed, the higher decoding gain can be obtained. In addition, the reference [10] introduces a multi-bit algorithm called the Gradient Descent Bit Flipping method and the algorithm is obtained from the concept of the gradient descent. Compared with WBF algorithm, the group shuffled and group replica shuffled BF (GRSBF) proposed in the reference [10] can effectively reduce the coding iteration times and have the excellent decoding performance. Therefore, compared with the signal-bit flipping method, the multi-bit flipping method can get the faster decoding convergent rate.

Compared with the SP algorithm, the bit flipping algorithm proposed in the above reference still has the obvious disparity in the performance aspect. In addition, the selection of the flipping-decision threshold used in the MBF algorithm [8-11] can be changed with the signal noisy ratio and the coding rate. If the selection of the flipping-decision threshold is not the optimum threshold, the correction of the performance would be significantly depredated. In order to improve the above problem, the paper introduces a reliability-based multibit-flipping decoding algorithm, and its threshold in the flipping-decision can be dynamically adjusted during the decoding process. Additionally, the proposed algorithm is not related to the communication channel and cannot be changed with the signal noise. The simulation results show that the proposed methods can evidently reduce the average iteration times, achieving the excellent performance and having a lower operating complexity.

2 Symbol definition

$(\mathrm{N,K})(d_v, d_c)$ LDPC is defined by the $M \times N$ Parity-Check Code matrix $H = (h_{m.n})$, $1 \le m \le M$, $1 \le n \le N$, where K represents the information length, M represents the numbers of the parity-check codes, N represents the length of the codeword's, d_v represents the numbers of the variable node degree, and dc represents the numbers of the check node degree. The codeword vector $c = (c_1, c_2, .. c_N)(c_n \in \{0,1\})$ should be input in the BPSK modulator and output signal vector is $x = (x_1, x_2, .. x_N)$ so

*** *Corresponding author's*** e-mail: 31581677@qq.com

that the corresponding relations with the codeword's are $x_n = 1 - 2c_n$. Later, the signal vector x should be transmitted through the additive white Gaussian noise whose average value is 0 and the variable number is $\sigma 2 = (2R_c E_b / N_0)^{-1}$, where $R_c = K / N$ and E_b represent the coding rate and the average energy in each information bit respectively. The receiving vector $y = (y_1, y_2, ...y_N)$ can be received in the receiving terminal, and the $z = (z_1, z_2, ...z_N)$ represents the bipolar vector in the hard-decision, that is, $z_n = \mathrm{sgn}(y_n) \in \{+1, -1\} (\mathrm{n} \in [1,\mathrm{N}])$. The set is $N(m) = \{n \mid h_{m,n} = 1\}$ for the check node m connects with all variable nodes, while the other set is $M(n) = \{m \mid h_{m,n} = 1\}$ for all check nodes connects with the variable node n. According to the representation, the condition satisfied with the parity-check code is as follows:

$$\prod_{meN(m)} z_m = 1, \forall m \in [1, M]. \tag{1}$$

If z meets the requirement of the condition in the Equation (1), the decoder is completely right, namely, $(z_1, z_2, ..z_N) \in c$. The m bipolar syndrome can be represented as follows:

$$S_m = \prod_{meN(m)} z_m, \ for\ m \in [1, M], \tag{2}$$

where it satisfies with $s_m \in \{+1, -1\}$.

3 Reliability-based multi-bit flipping algorithm

Although the general BP decoder [1,2] can achieve a better performance, the operating complexity is rather high. Compared with the BP decoder, the traditional BF decoders are very simple with higher performance consumption. The chapter firstly introduces the CIWBF with the hard-decision derived from the BP algorithm with the soft-decision. The algorithm can effectively increase the performance of the traditional BF decoder, and then the Reliability-based multi-bit flipping algorithm can be derived.

The BP algorithm can be represented by the Tanner figure, and the reliable information can be transmitted between the variable nodes and the check nodes. The SP algorithm in the BP algorithm is the most common types. Aiming to the AWGN channel, the information of the initial value can be easily proved as $2y_n/\sigma^2$ by transmitting the variable node to the check node. After the SP algorithm is through the first iteration, the log-likelihood ratio in the n bit can be represented as follows:

$$L(cn \mid y) = \log \frac{\Pr(c_n = 0 \mid y)}{\Pr(c_n = 1 \mid y)} =$$

$$\frac{2y_n}{\sigma^2} + \sum_{meM(n)} \left(\prod_{meN(m)\backslash n} Zi \right) \cdot \varphi \left(\sum_{ieN(m)\backslash n} \varphi \left(\frac{2|y_i|}{\sigma^2} \right) \right) = \tag{3}$$

$$Z_n \left(\frac{2|y_n|}{\sigma^2} + \sum_{meM(n)} S_m \cdot \varphi \left(\sum_{ieN(m)\backslash n} \varphi \left(\frac{2|y_i|}{\sigma^2} \right) \right) \right),$$

where $\varphi(x)$ is equal to $-\ \ln[tan\mathrm{h}(x/2)]$. Aiming at the realization of $\varphi(x)$ function, the realization in the software can be obtained by applying the comparison table, but it is difficult to obtain the realization in the hardware. The reason is that the dynamic range in the $\varphi(x)$ is too large that the $\varphi(x)$ function is difficult to be approximated with many comparison tables. Therefore, the realization in the hardware often causes the decline of the performance, especially in the error floor region. In order to reduce y_i complexity of the $\varphi(x)$ function and avoid the decline of the performance, the CIWBF algorithm [6] makes the $\varphi(x)$ function realize the piecewise linearization, and the first-order polynomial $\varphi p(x)$ can be obtained by the approximating value. Later, the factor in the first-order polynomial $\varphi p(x)$ can be obtained with the minimum square so that the $\varphi p(x)$ is approaching to the $\varphi(x)$. Therefore, the Equation (3) can be approximated as follows:

$$L(c_n \mid y) \approx aZ_n E_n, \tag{4}$$

where α and E_n are constants, and E_n can be represented as follows:

$$E_n = \sum_{meM(n)} S_m w_{m,n}, \tag{5}$$

where $w_{m,n}$ can be represented as follows:

$$w_{m,n} = \left(\frac{1}{|y_m|} \sum_{ieN(m\backslash n)} |y_i| \right). \tag{6}$$

The reliable measuring E_n in the receiving bit Z_n of the hard decision can be obtained by using the Equations (5) and (6). When the E_n is smaller, the more symptom whose Z_n connects with the check node cannot meet the requirement of the condition of the parity-check code. Therefore, Z_n is not realizable. The Z_n is the most possibly error under the condition so that it must be flipped. Each element in the vector $E = (E_1, E_2, E_3, ..., E_N)$ can respectively represent the reliable measurement in each bit Z_n.

In order to flip many bits during the iteration process and fasten the decoding convergent rate, the RBMBF designing method can be explained as follows:

The min E represents the minimum E_n in the reliable measurement vector E, and the definition of the flipping threshold is as follows:

$$E_{th} = E_{\min} - \beta \left| E_{\min} \right| \left(\frac{1}{M} \sum_{ieN(m \backslash n)} S_m \right), \quad (7)$$

where β value can be obtained through the computer imitation.

The set in least reliability position in the E can be defined as follows:

$$LRP(E) = \{ n \mid E_n \geq E_{th}, n \in [1, N] \}, \quad (8)$$

$W = (\beta / M) \left| E_{\min} \right| \sum_{m-1}^{M} S_m$ represents the size of searching the window in the LRP, and the least reliability position should be searched between the E-th and $E_{\min}$. Equation (7) shows that the larger the total value $S = \sum_{m-1}^{M} S_m$ is, the more conditions there are that satisfy the parity-check code in S_1, S_2, ..., S_M. Therefore, the average $E_{avg} = \left(\sum_{m-1}^{M} E_n \right) / N$ in the reliable measurements is larger. Under the condition, large parts of the E_n are farther to the $E_{\min}$ so that the larger window W can be selected to flip the multi-bit. Otherwise, when the total value S is smaller and the more S cannot meet the requirement of the parity-check code, E_{avg} is smaller. Under the condition, large parts of the E_n are closer to the $E_{\min}$ so that the smaller window W can be selected to flip the multi-bit. In the Equation (7), the flipping threshold Eth can be dynamically adjusted with the condition of the decoder during the decoding process in terms of the symptom of the hard-decision z.

The decoding step in the RBMBF is described as follows:

Initiation: the k is set as 0, $z(0)$ is set as z and the maximum iteration times are $k_{\max}$, the $w_{m,n}$ can be computed by using the Equation (6), and the $1 \leq m \leq M$ and $n^* \in N(m)$ can be stored.

1) Computing the symptom $S_m^{(k)} + S^{(k)} = \sum_{m=1}^{M} S_m^{(k)}$. If $S(k)$ equals to M, the decoding is stopped and the output z is used as the codeword's after being decoded.

2) According to the Equation (5), the reliable measurement $E^{(k)} = (E_1^{(k)}, E_2^{(k)}, \ldots, E_N^{(k)})$ in the $Z(k)$ can be computed.

3) The flipping threshold $E_{th}^{(k)}$ can be computed in terms of the Equation (7).

4) Multi unreliable bit position $n^* \in LRP(E(k))$ can be searched and its multi bits can be flipped in the same time in terms of the Equation (8).

5) $k \leftarrow k+1$. If k is larger than $k_{\max}$, the decoding should be stopped the output z is used as the codeword's after being decoded. Otherwise, the process must be back to the step 2.

In the above decoding processes, the purposes of the steps 2-4 are to compute the reliable measurement $E(k)$ searching the most unreliable bit position. Finally, multi unreliable bits should be flipped in the same time in the step 4.

4 Simulation results and performance comparison

4.1 SIMULATION RESULTS

In order to verify the effectiveness of the RBMBF proposed in the paper, the research adopts to the widely used traditional BF algorithm (including WBF [3], IMWBF [5] and GDBF [10]). The mainly compared contents are the computer simulation and the performance comparisons. The projects of the performance comparison are bit error rate, codeword error rate and average iteration times.

The coding method derived from the Mackay [12] is adapted to cause the (2000, 1000) regular LDPC, and its coding rate is $Arc = 1/2$. The maximum iteration times is set as 200, and the parameter needed in the IMWBF is α = 0.4. The BER's, CER's and AIN's performance simulation results in each algorithm are respectively as shown in Figures 1-3.

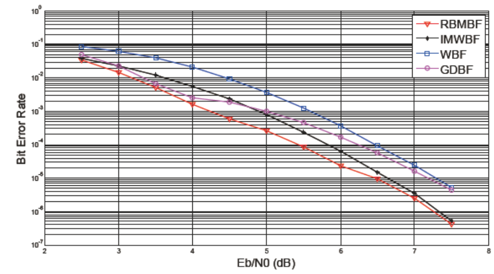


FIGURE 1 the bit error rate performance comparison figure in the (2000, 1000) regular LDPC

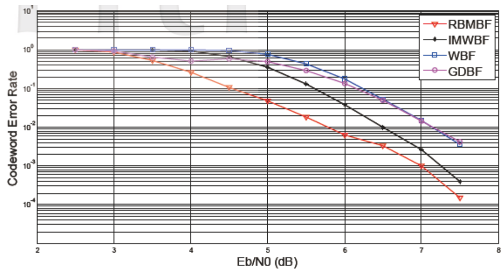


FIGURE 2 the codeword error rate performance comparison figure in the (2000, 1000) regular LDPC

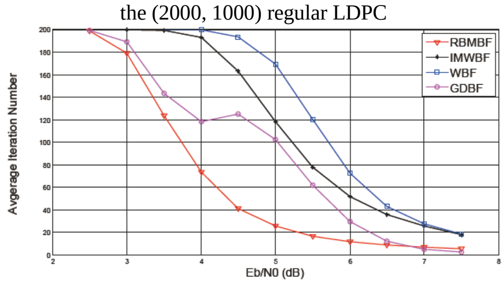


FIGURE 3 the average iteration times comparison figure in the (2000, 1000) regular LDPC

Figure 1 shows that the bit error rate performance comparison in the RBMBF, IMWBF, WBF, GDBF and other algorithms. Compared with the IMWBF, WBF and

GDBF algorithm, the proposed RBWBF can respectively realize the 0.8dB， 0.2dB and 0.7dB coding gain when the BER is 10−5. The figure 2 shows that the codeword error performance comparison. Compared with the IMWBF, WBF and GDBF algorithm, the proposed RBWBF can respectively realize the 1 dB, 0.5 dB and 1 dB coding gain when the CER is 3×10^{-3}.The figure 3 shows the average iteration times during the decoding process. Compared with the IMWBF, WBF and GDBF algorithm, the proposed RBWBF can respectively reduce the84%, 78% and 74% average iteration times when the E_b/N_0 is 5dB. The simulation results show that RBMBF can achieve a better correction performance in the bit error rate aspect compared with other three traditional bit flipping algorithms, the same as in the average iteration time's aspect.

In addition, the method derived from the Mackay [12] is also adapted to cause the (1920, 640) irregular LDPC, and its coding rate is $R_c = 1/3$. The maximum iteration times is set as 200, and the parameter needed in the IMWBF is $\alpha = 0.4$. The BER's, CER's and AIN's performance simulation results in each algorithm are respectively as shown in the Figures 4-6.

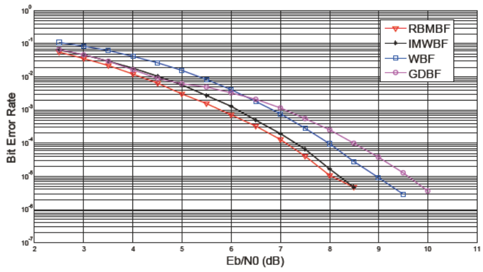


FIGURE 4 the bit error rate performance comparison figure in the (1920, 640) irregular LDPC

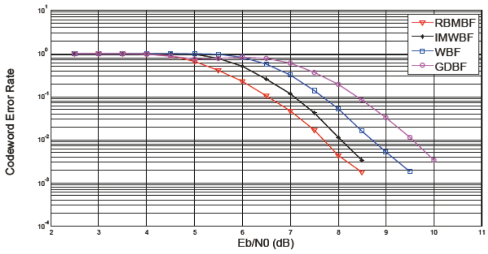


FIGURE 5 the codeword error rate performance comparison figure in the (1920, 640) irregular LDPC

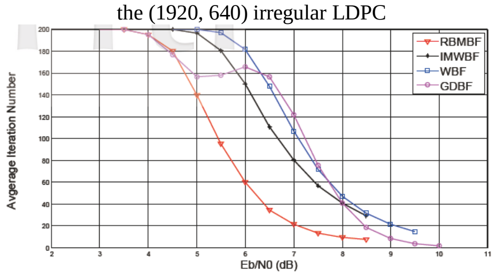


FIGURE 6 the average iteration times comparison figure in the (1920, 640) irregular LDPC

Figure 4 shows that the bit error rate performance comparison in the RBMBF, IMWBF, WBF, GDBF and other algorithms. Compared with the IMWBF, WBF and GDBF algorithm, the proposed RBWBF can respectively realize the 1dB, 0.2dB and 1.6dB coding gain when the BER is 10^{-5}. The figure 5 shows that the codeword error performance comparison. Compared with the IMWBF, WBF and GDBF algorithm, the proposed RBWBF can respectively realize the 1.1dB， 0.4dB and 1.9dB coding gain when the CER is 3×10^{-3}.The Figure 6 shows the average iteration times during the decoding process. Compared with the IMWBF, WBF and GDBF algorithm, the proposed RBWBF can respectively reduce the 66%, 60% and 63% average iteration times when the E b/N 0 is 6dB. Compared with other three algorithms, RBMBF can achieve a better balance between the performance and the complexity.

4.2 THE ANALYSIS COMPARISON OF THE OPERATING COMPLEXITY

The initial solution before the iteration decoding need compute $w_{m,n}$ value in the Equation (6). There are Md_v values in total should be computed and stored. The explanation for the operating complexity in each iteration decoding is as follows:

As to each $n^* \in LRP(\mathrm{E}(k))$ bit in the decoding step 2, the d_v which has $s_m, s_m^{(k+1)} \leftarrow -s_m^{(k)}, m \in M(n^*)$ symptom values need to be updated during the process of one iteration decoding. Later, as to all $n \in N(\mathrm{m})$ in each $m \in M(n^*)$ which needs to update the $E_n^{(k+1)}$, the $E_n^{(k+1)} \leftarrow E_n^{(k)} - 2s_m^{(k)} w_{m,n}$ needs to be operated. There are $L(k)$ bits needs to be flipped in the N bits during the period of each iteration. Therefore, the updating of flipping function $E_n^{(k+1)}$ needs $L(k)d_vd_c$ additions. In order to compute $E_{th}^{(k)}$ in the decoding step 3, an addition and two multiply operations are needed. $L(k)$ bits in the N bit needs to be found out in the decoding step 4, and its bit position is $n^* \in LRP\left(E(\mathrm{k})\right)$ so that it needs N comparators. RBMBF algorithm needs $N + L^{(k)} d_v d_c + 1$ additions in total. IMWBF algorithm needs $N - 1 + d_v d_c$ additions in total during the period of each iteration for the numbers of the adders are larger than the numbers of the comparators and the comparators can be regarded as the adders.

The computing explanation of the whole iterating complexity in the decoding process is as follows: in order to simplify the computing process, RBMBF algorithm and IMWBF algorithm can be assumed to find out the error bits and be flipped in the decoding process. Aiming at the decoders in the RBMBF algorithm, NBE and NI represent the numbers of the receiving error bits and the needed iteration times respectively. The total numbers of the flipping error bits in the RBMBF decoder and the IMWBF decoder are $N_{BE} = \sum_{k=1}^{N_I} L^{(k)}$ and NBE respectively.

Therefore, the whole operating complexities in the IMWBF and the RBMBF are $N_{BE} = N - 1 + d_v d_c$ and $\sum_{k=1}^{N_I}(N + L^{(k)} d_v d_c + 1) = N_I(N+1) + N_{BE} d_v d_c$ respectively. NI is smaller than NBE according to the Figure 3 and the Figure 6. Therefore, it proves that the whole operating complexity $N_I(N+1) + N_{BE} d_v d_c$ in the RBMBF is smaller than the whole operating complexity $N_{BE}(N - 1 + d_v d_c)$ in the IMWB. The whole comparison tables to the decoding operating complexity is as shown in the Table 1.

TABLE 1 the operating complexity comparison during the decoding

Algorithm	RBMBF	IMWBF [5]
Addition/each iteration	$N + L^{(k)} d_v d_c + 1$	$N - 1 + d_v d_c$
The whole additions	$N_I(N+1) + N_{BE} d_v d_c$	$N_{BE} = (N - 1 + d_v d_c)$

Therefore, although the RBMBF increases a little operating complexity in each iteration process, the iteration times in the multi-bit flipping RBMBF is smaller than the iteration times in the signal-bit flipping IMWBF so that the RBMBF algorithm can obviously reduce the whole operating complexity.

5 Conclusions

In order to speed up decoding convergent speed, the paper proposes a multi-bit flipping RBMBF algorithm. The numbers of the flipping bits can be dynamically adjusted with the channels and the iterations in each iteration decoding process. Aiming at the decoding procedure in the regular and irregular LDPC, RBMBF can have faster decoding convergent rate and have a better performance compared with the traditional multi-bit flipping algorithm GDBF. Compared with other BF algorithms, the RBMBF can reach a better balance between the decoding complexity and the actual performance. The RBMBF algorithm can be properly applied in the real-time and high-channel decoder for it has a faster convergent rate, a lower operating complexity and other advantages.

Acknowledgments

This work was supported by Department of Science & Technology, Hunan Province, Project Serial No.2013GK3143.

This work was supported by Project Funded by Department of Education, Hunan Province, Project Serial No.13B079.

This work was supported by the National Natural Science Foundation of China (Grant No: 61202464).

References

[1] Gallager R G 1962 Low-density parity-check codes *IRE Transactions on Information Theory* **7**(8) 21-8

[2] Fossorier M P C, Mihaljevic M J, Imai H 1999 *IEEE Transactions on Communications* **3**(47) 673-80

[3] Kou Y, Lin S, Fossorier M P C 2001 *IEEE Transaction on Information Theory* **11**(47) 2711-36

[4] Zhang J, Fossorier M P C 2004 *IEEE Communications Letters* **3**(8) 165-7

[5] Jiang M, Zhao C, Shi Z, Chen Y 2005 *IEEE Communications Letters* **9**(9) 814-6

[6] Chen T C 2012 A channel independent weighted bit-flipping decoding algorithm for LDPC codes *IET Communications* **11**(6) 2968-73

[7] Li G, Li D, Wang Y, Sun W 2010 Hybrid decoding of finite geometry low-density parity-check codes *IET Communications* **7**(4) 1238-46

[8] Wu X, Zhao C, You X 2007 Parallel weighted bit-flipping decoding *IEEE Communications Letters* **8**(11) 671-3

[9] Li G, Feng G 2009 Improved parallel weighted bit-flipping decoding algorithm for LDPC codes *IET Communications* **1**(3) 91-9

[10] Wadayama T, Nakamura K, Yagita M, Funahashi Y, Usami S, Takumi I 2010 *IEEE Transactions on Communications* **6**(58) 1610-4

[11] Zhang J, Yedidia J S, Fossorier M P C 2007 Low-latency decoding of EG LDPC codes *Journal of Lightwave Tech*nologies **9**(25) 2879-86

[12] Mackay D J C 1999 *IEEE Transactions on Information Theory* **5**(45) 399-431

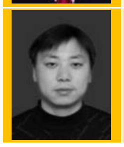
Authors

Yingying Peng, February 1982, China.

Current position, grades: researcher at Hunan University of Chinese Medicine, China.
University studies: bachelor degree in information science and technology from Hunan Normal University, China in 2004.
Scientific interests: information security and cloud computing.

Xuegang Ren, September 1986, China.

Current position, grades: experimental division at Hunan University of Chinese Medicine, China.
University studies: bachelor degree in computer science and technology at Hunan University of Chinese Medicine, China in 2007.
Scientific interests: information security and computer network management.



Defa Hu, October 1980, China.

Current position, grades: researcher at Hunan University of Commerce, China.
University studies: PhD degree in computer science and technology at Hunan University, China in 2010.
Scientific interests: information security and image processing.