

Simple Improvements of Bit-Flipping Decoding

Hiroshi Kamabe Shusaku Kobota

Dept. of Information Science

Gifu University

1-1 Yanagido, Gifu 501-1139, Japan.

Email: kamabe@ieee.org, Shusaku.Kobota@kmb.info.gifu-u.ac.jp

Abstract—A bit-flipping (BF) algorithm is a decoding algorithm for LDPC codes that can be efficiently implemented by electronic circuits. Although its error correcting performance is rather poor, we show that a BF algorithm with bias terms has good error correcting performance. We also introduce two more parameters to improve the performance further. We also propose a variant of candidate bit based BF (CBBF) algorithm by introducing bias terms. We see that these algorithms having bias terms exhibit very good error correcting performances.

Index Terms—low-density parity-check (LDPC) codes, bit-flipping decoding, WBF decoding, CBBF decoding

I. INTRODUCTION

Low-density parity-check (LDPC) codes are linear codes having very sparse parity check matrices. Iterative decoding for LDPC codes using soft information, e.g. sum-product (SP) algorithm, has very good decoding performance [1]. It was shown by computer simulation that if the code length is very long then the performance is very close to the Shannon limit.

There is a simple decoding algorithm called bit-flipping (BF) decoding algorithm, which uses only hard information, and therefore, it can be efficiently implemented by electronic circuits. Since the algorithm does not use any soft information, the performance of the algorithm is inferior to that of the SP algorithm.

Although, a weighted BF (WBF) algorithm can be regarded as a variant of the BF algorithm [7], the WBF algorithm uses not only hard information but also soft information during its decoding process.

A quantized modified bit-flipping (QMBF) is also a variant of the BF algorithm [3]. In the algorithm, if a received signal is relatively close to 0, a bit corresponding to the signal is considered to be unreliable. Thus, a bias term is added to a value which is used in making a decision to flip the bit. It has been shown that the performance of the algorithm is rather better than that of the BF algorithm.

A candidate bit based bit-flipping (CBBF) algorithm is a variant of the BF algorithm [4]. In the algorithm, the reliability of a parity check equation is calculated in addition to the reliability of each bit. The error correcting performance of the CBBF algorithm is also better than the BF algorithm.

In this paper, two decoding algorithms with bias terms are proposed and it is shown that these algorithms are better than the original algorithms having no bias terms.

II. BF DECODING ALGORITHM AND ITS VARIANTS

The BF algorithm is very simple but has a poor performance. Therefore, many variants of the algorithm were proposed to design decoding algorithm with moderate complexity and better performance than the BF algorithm.

Decoding algorithms which we propose here are applied to the binary Gaussian channel with inputs of ± 1 and additive noise.

Let H be a parity check matrix of an LDPC code of size $M \times N$. In this study only regular LDPC codes are used. By j , we mean the column weight. Let $\mathbf{y} = y_1 y_2 \cdots y_N$ be a received signal. Then, a received word $\mathbf{z} = z_1 z_2 \cdots z_N$ is defined by

$$z_i = \begin{cases} 1, & \text{if } y_i \leq 0, \\ 0, & \text{otherwise,} \end{cases}$$

for $i = 1, 2, \dots, N$.

A. BF Decoding Algorithm

A typical BF algorithm is given as follows. The following algorithm is called a maximum detection bit flipping algorithm. It should be noticed that only integer operations are required and therefore the algorithm may be efficiently implemented by an electronic circuit.

The variables used are as follows: l is a count for iteration, $l_{\max}$ is the maximum number of iterations, B is a threshold for the termination condition of the algorithm, $\mathbf{v}$ is a temporal code word, and $\mathbf{v}'$ is a code word for the next iteration. A symbol " $\leftarrow$ " is used to signify a substitution from the right-hand side to the left-hand side.

Step0: $l \leftarrow 0, \quad \mathbf{v} \leftarrow \mathbf{z}$.

Step1: Calculate syndrome $\mathbf{s} = s_1, s_2, \dots, s_M \leftarrow \mathbf{v}H^T$; If $\mathbf{s} = \mathbf{0}$, then the algorithm terminates.

Step2: Calculate e_n by

$$e_n \leftarrow \sum_{m \in M(n)} s_m.$$

Step3: If $\max_j e_j < B$, then the algorithm terminates, where B is usually set to $\lceil j/2 \rceil$.

Step4: Flip all bits v_i with $e_i = \max_k e_k$ and

$$l \leftarrow l + 1.$$

Let $\mathbf{v}'$ be a vector obtained by flipping bits.

Step5: If $l > l_{\max}$, then the algorithm terminates.

Step6: $\mathbf{v} \leftarrow \mathbf{v}'$ and go to Step 1.

The number e_n calculated in Step 2 is the number of unsatisfied check equations including n -th bit. Therefore, if e_n is large, then the n -th bit may be incorrect with high probability.

At Step 4 we can employ a different criterion for finding bits to be flipped. For example, the following step can be possible,

Step4': Flip all bits v_i with $e_i \geq \delta$ and $l \leftarrow l + 1$,

where δ is a predefined integer.

The Step 3 is well defined only for regular LDPC codes. If H is not regular, B may depend on column.

If the number of error bits in a received code word is small enough, then the error bits may disappear as the algorithm proceeds.

B. Weighted bit-flipping (WBF) algorithm

A weighted bit-flipping (WBF) algorithm is an iterative decoding algorithm that is a variant of the BF algorithm[7]. The algorithm invokes floating point arithmetic operations during iterations.

Step 0: $l \leftarrow 0$ and $\mathbf{v} \leftarrow \mathbf{z}$.

Step 1: Calculate syndrome $\mathbf{s} = s_1 s_2 \dots s_M \leftarrow \mathbf{v} H^T$. If $\mathbf{s} = \mathbf{0}$ or $l \geq l_{\max}$, then the algorithm terminates.

Step 2: Calculate $y_{\min}^m$ by

$$y_{\min}^m \leftarrow \min_{n \in N(m)} |y_n| \quad \text{for } m = 1, \dots, M.$$

Step 3: Calculate E_n by

$$E_n \leftarrow \sum_{m \in M(n)} (2s_m - 1)y_{\min}^m \quad \text{for } n = 1, \dots, N.$$

Step 4: Flip all bits v_n with $E_n = \max_{i=1, \dots, N} E_i$. Let $\mathbf{v}'$ be the resulting code word.

Step 5: $l \leftarrow l + 1$ and $\mathbf{v} \leftarrow \mathbf{v}'$. Go to Step 1.

C. QMBF Decoding

QMBF algorithm is a decoding algorithm based on the BF algorithm with bias terms [3]. Since it is very similar to the BF algorithm, only differences between them are given in the following.

Before iterations, bias terms are calculated. Hence Step 0 is modified as follows.

Step 0': $l \leftarrow 0$, $\mathbf{v} \leftarrow \mathbf{Z}$ and define β_n by

$$\beta_n \leftarrow \begin{cases} r, & \text{if } |y_n| < \alpha, \\ 0, & \text{otherwise,} \end{cases}$$

where r is an integer and α is a positive real number.

Step 2 is modified as follows:

Step 2': Calculate e_n by

$$e_n \leftarrow \sum_{m \in M(n)} s_m + \beta_n.$$

In this algorithm, β_n reflects the reliability of y_n . If $|y_n|$ is small then it means y_n contains much noise. Therefore we do not believe the value y_n .

Fig. 1 shows the distribution of an output of a binary bipolar channel. If $|y_n| < \alpha$ then it can be assume that the

amplitude of a noise is relatively large. Hence by adding r to $\sum_{m \in M(n)} s_m$, v_n is flipped more likely.

This idea is different from the BF algorithm and the SP algorithm. In QMBF the initial belief about a received signal has a strong influence on deciding bits to be flipped during the whole process for decoding. In the BF algorithm and the SP algorithm the initial belief about the input signals will disappear as the algorithms proceed.

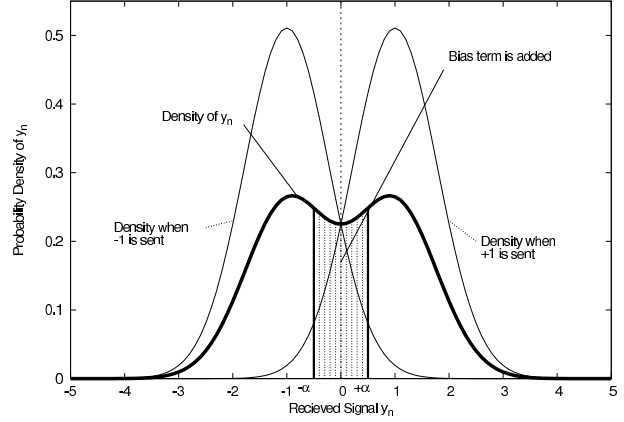


Fig. 1. α and distribution of y_n

D. CBBF algorithm

A candidate bit based bit-flipping (CBBF) decoding algorithm is also a variant of the BF algorithm [4]. The algorithm has a new parameter $\delta > 0$ which is an integer valued threshold for decision on candidate bits. The procedure of the algorithm is given as follows:

Step0: $l \leftarrow 0$ and $\mathbf{v} \leftarrow \mathbf{z}$.

Step1: Calculate syndrome $\mathbf{s} = s_1, \dots, s_M \leftarrow \mathbf{v} H^T$. If $\mathbf{s} = \mathbf{0}$, then terminate the decoding procedure.

Step2: Calculate u_i for $i = 1, 2, \dots, N$

$$u_i = \sum_{m \in M(i)} s_m.$$

(u_i is the number of unsatisfied parity check equations including the i -th bit.) If $u_i > \delta$, then v_i is marked as a candidate bit for bit-flipping.

Step3: Find $u_{\max} = \max_i u_i$. If $u_{\max} \leq \delta$, then terminate the decoding procedure.

Step4: Let c_m be the number of candidate bits included in the m -th parity check equation and calculate $w_m = c_m - 1$ for $m = 1, 2, \dots, M$.

Step5: For $i = 1, 2, \dots, M$ with $u_i = u_{\max}$, calculate r_i by

$$r_i = \sum_{m \in M(i)} w_m.$$

Step6: Flip all bits v_i with $r_i = \min_i r_i$ and $u_i = u_{\max}$. Let $l \leftarrow l + 1$. If $l > l_{\max}$, then terminate the decoding procedure. Go to Step 1.

At Step2, u_i is calculated as well as the BF algorithms. At Step4, w_m is calculated based on u_n . This is one of main differences between BF and the CBBF. This value may mean the reliability of the m -th parity check equation. If w_m is large, then it means the m -th parity check equation contains many candidate bits for flipping, and hence, the reliability of the equation may be low because the equation may tell us nothing about bits to be flipped.

At Step5, r_n is calculated by summing w_m up for $m \in M(n)$. Smaller r_n means that v_n is included in more reliable parity check equations. Therefore, if $u_n = u_{\max}$ and $r_n = r_{\min}$ then it means that v_n is included in many unsatisfied but reliable parity check equations.

III. PROPOSED DECODING ALGORITHMS

In this section two decoding algorithms are proposed, QMBF with two parameters and CBBF with bias terms.

A. QMBF2

A variant of QMBF, called QMBF2, is given in this subsection. In QMBF2, β'_n is used in Step 2 instead of β_n :

$$\beta'_n = \begin{cases} r_1, & \text{if } 0 \leq |y_n| < \alpha_1, \\ r_2, & \text{if } \alpha_1 \leq |y_n| < \alpha_2, \\ 0, & \text{otherwise,} \end{cases}$$

where $r_1 > r_2$ and $\alpha_1 < \alpha_2$. Parameters α_1 and α_2 are positive real values which correspond to α in QMBF.

In QMBF2, two different bias terms are used. Since $\alpha_1 < \alpha_2$, if $|y_n| < \alpha_1$, then y_n can be considered to be less reliable. Hence, for the case $|y_n| < \alpha_1$, β_n is set to r_1 so that z_n is flipped more likely in iteration steps for bit-flipping. It is expected that the performance of the modified algorithm may be better because bias terms are defined more precisely. Fig. 2 shows this idea.

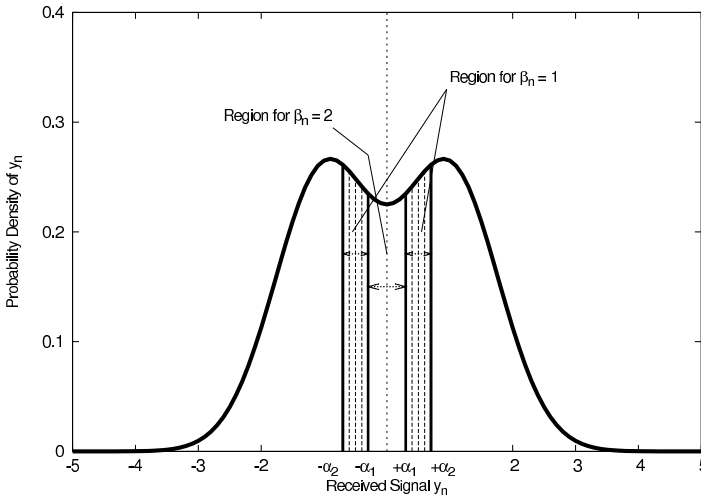


Fig. 2. α_1 , α_2 and Distribution of y_n

B. CBBF with bias terms(QCBF)

Bias terms are also introduced to CBBF and the resulting algorithm is called QCBF.

At the first step of the algorithm, the following bias terms are calculated for $n = 1, 2, \dots, N$:

$$\beta_n = \begin{cases} r, & \text{if } |y_n| < \alpha, \\ 0, & \text{otherwise.} \end{cases}$$

At Step 2 of CBBF, u_i is calculated as follows:

$$u_i = \beta_i + \sum_{m \in M(i)} s_m.$$

The bias terms are introduced according to the same idea explained in the previous subsection. Since bits with bias terms may be flipped with high probability, it can be expected that the error correcting performance will be improved as well as the QMBF.

IV. COMPLEXITY AND PERFORMANCE OF PROPOSED ALGORITHMS

Although the proposed algorithms need calculations for bias terms, calculations needed for the bias terms are virtually comparisons. A circuit implementing the algorithm needs extra memories for bias terms. However, it should be noticed that the values are only small integers which can be represented by 1 or 2 bits. Since the bias terms are constant during the decoding operation, it is expected that the bit-flipping algorithms are easily implemented by electronic circuit and that the algorithms process received signals as quickly as original algorithms.

Since decoding algorithms based on the BF algorithm except for WBF algorithm use no soft information during their decoding processes, the error correcting performance is inferior to SP but the complexity of BF algorithms are very low.

Although CBBF uses no soft information during decoding process, CBBF calculates reliabilities of parity check equations. Therefore, it can be said that CBBF is more complex than QMBF. Since WBF uses soft information during decoding process, WBF is more complex than CBBF.

The following inequalities summarize these observations about complexity: $SP \gg QCBF \simeq CBBF > QMBF2 > QMBF > BF$. The complexities of QMBF and QMBF2 are almost the same. However, their error correcting capabilities are different.

V. SIMULATION RESULTS

In this section simulation results of the decoding algorithms are given.

For each algorithm we repeat the decoding procedure until we obtain 500 error bits, where the maximum number of repetitions of simulation is 10,000,000.

Three LDPC codes in the MacKay's database[5] were used, 1057.244.3.353, 1057.244.4.360, 4095.737.3.101. These code are called LDPC(M1), LDPC(M2) and LDPC(M3), respectively. A code constructed by the progressive edge growth

algorithm [6] was also used. The code rate of the code is 0.5 and is called LDPC(P).

Parameters of these codes are given in Table I.

TABLE I
PARAMETERS OF CODES

Code	length	rate	col. weight
LDPC(M1)	1057	0.77	3
LDPC(M2)	1057	0.77	4
LDPC(M3)	4095	0.82	3
LDPC(P)	2000	0.5	4

In the following, *opt* means that optimized parameters are used.

A. Optimization of parameters

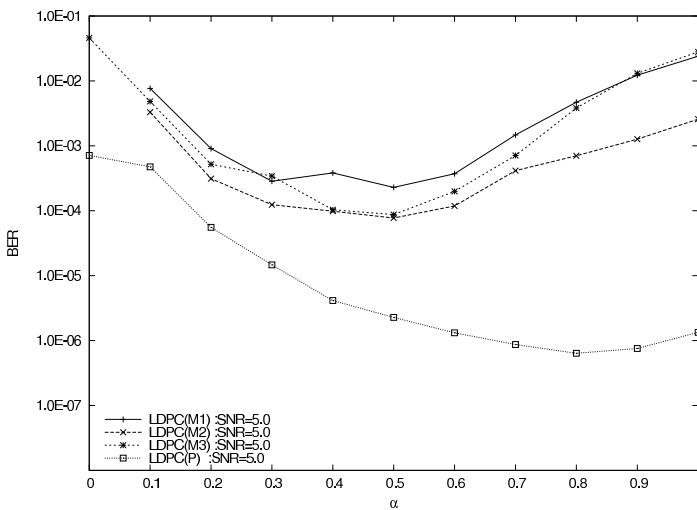


Fig. 3. Optimization of α for QMBF

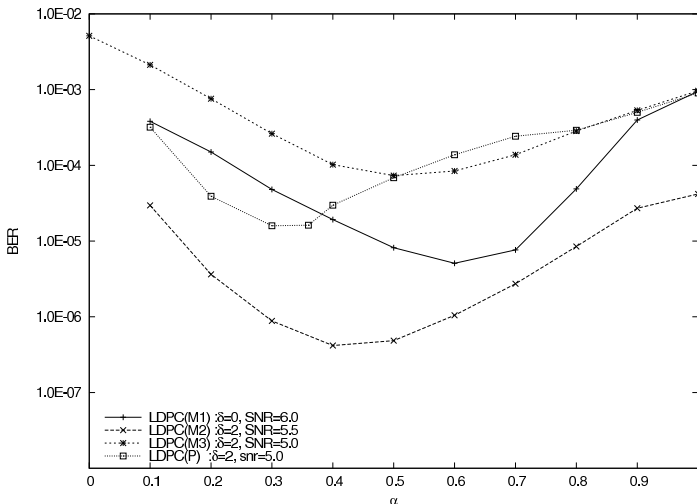


Fig. 4. Optimization of δ for QCBF

Parameters of proposed decoding algorithms are optimized based on simulation results. The criterion for optimization is a BER performance at some SNR. These values of SNR depend

on codes and decoding algorithms. In Table II SNR is shown for each pair of a code and a decoding algorithm.

TABLE II
SNR FOR PARAMETER OPTIMIZATION

	QMBF	QMBF2	QCBF
LDPC(M1)	5.0	5.5	6.0
LDPC(M2)	5.0	5.0	5.5
LDPC(M3)	5.0	5.0	5.0
LDPC(P)	5.0	4.5	5.0

Parameter α of QMBF was optimized as follows. The BER performances were calculated for $\alpha = 0.1, 0.2, \dots, 1.5$, and then, the best value was chosen with respect to BER for each code. The parameters α_1 and α_2 for QMBF2 were optimized as follows. The BER performances were calculated for $\alpha_1 = 0.1, 0.2, \dots, 1.2$ and $\alpha_2 = 0.1, 0.2, \dots, 1.5$ and then the best tuple was chosen with respect to BER for each code. The parameter δ of CBBF was also optimized. The BER performance was calculated for $\delta = 0, 1, \dots, j$ (j is the column weight) and then the best value was chosen with respect to BER. Parameters α and δ of QCBF were optimized over pairs (δ, α) for $\delta = 0, 1, 2, 3$ and $\alpha = 0.1, 0.2, \dots, 1.5$.

Optimized parameters for each decoding algorithm are given in Table III where $l_{\max}$ is the maximum number of iterations in a decoding procedure and B is the threshold for termination of algorithm.

TABLE III
PARAMETERS OF CODES

	α	r	δ	$l_{\max}$	B
BF	-	-	-	45	$\lceil \frac{d}{2} \rceil$
QMBF	optimized	1	-	45	$\lceil \frac{d}{2} \rceil$
QMBF2	optimized	(2,1)	-	45	$\lceil \frac{d}{2} \rceil$
CBBF	-	-	optimized	45	-
QCBF	optimized	(2,1)	optimized	45	-

The maximum iteration number of WBF and SP are 45 and 10, respectively. In the table, symbol “-” means there is no parameter for a decoding algorithm corresponding to the row.

B. Simulation results

Fig. 5, Fig. 6, Fig. 7 and Fig. 8 show simulation results for LDPC(M1), LDPC(M2), LDPC(M3) and LDPC(P), respectively. From these result, decoding algorithms are ordered according to the BER performance.

LDPC(M1)

QMBF2 > QCBF > QMBF > WBF $\simeq$ CBBF

LDPC(M2)

QCBF > QMBF2 > QMBF > WBF > CBBF

LDPC(M3)

QMBF2 > QMBF > QCBF > CBBF > WBF

LDPC(P)

QMBF2 $\simeq$ QMBF > QCBF > CBBF $\simeq$ BF > WBF

It can be seen that QMBF is superior to BF for all four codes. Differences are 1.7db, 1.2db, 1.9db, and 1.0db at BER

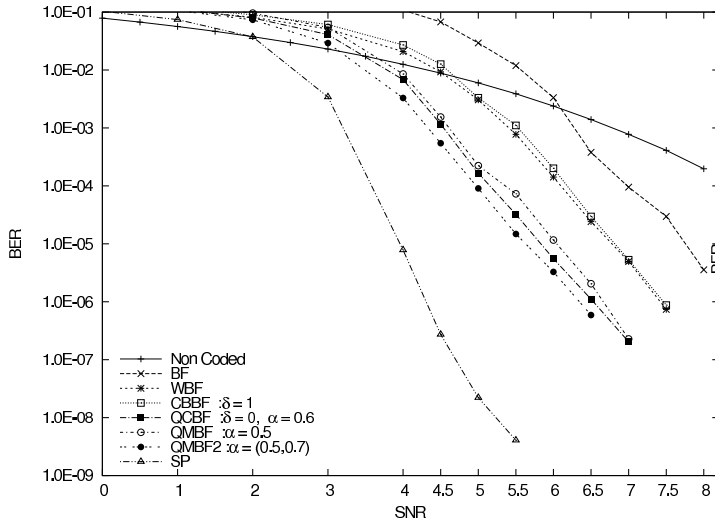


Fig. 5. BER for LDPC(M1)

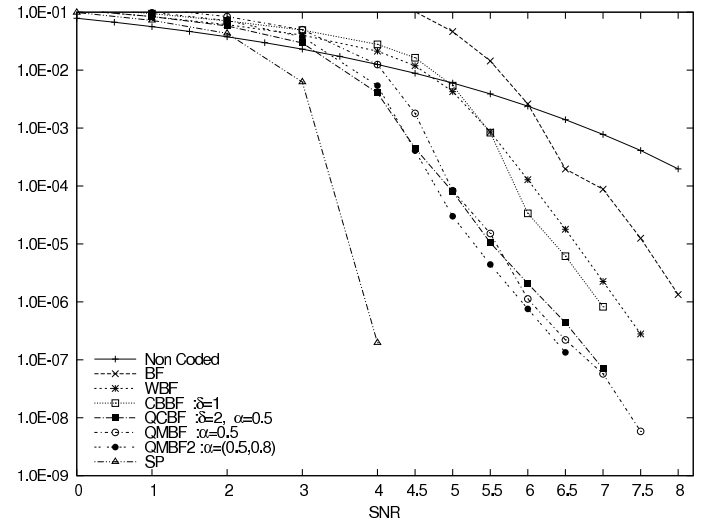


Fig. 7. BER for LDPC(M3)

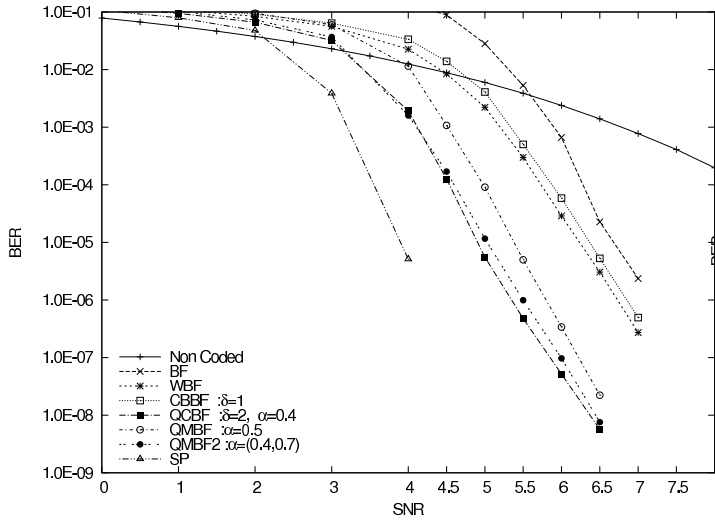


Fig. 6. BER for LDPC(M2)

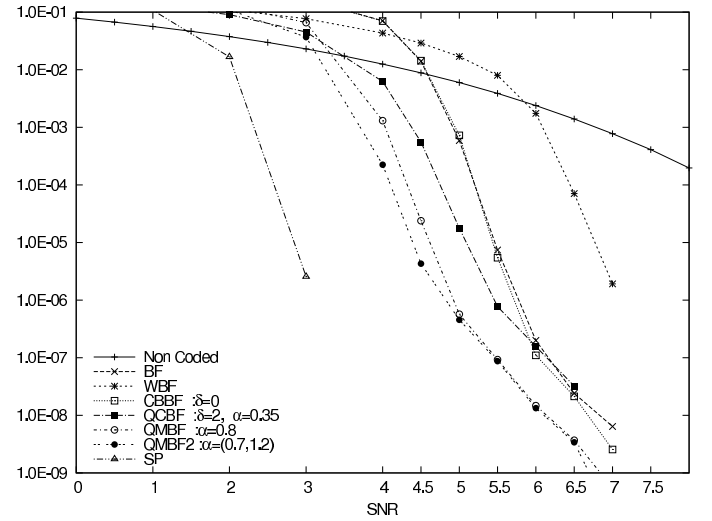


Fig. 8. BER for LDPC(P)

1.0^{-5} for codes LDPC(M1), (M2), (M3) and P, respectively. It should be noticed that the amplitude of improvement by adding bias terms for LDPC(M3) is larger than those for LDPC(M1) and LDPC(M2). This may imply that longer codes will gain larger improvement.

For all codes the BER performance of QMBF2 is better than that of QMBF. Differences are 0.5db, 0.4db, 0.2db and 0.2db at BER 1.0^{-5} , for codes LDPC(M1), (M2), (M3) and P, respectively. This may imply that a careful design of bias terms may result further improvement of performance for LDPC codes with large column sums.

For all codes the BER performance of QCBF is better than that of CBBF. Differences are 1.1db, 1.5db, 1.0db and 0.8db at BER 1.0^{-5} for LDPC(M1), (M2), (M3) and P, respectively.

QMBF and QMBF2 are superior to WBF for all four codes. It should be noticed that WBF is more complex than QMBF and QMBF2.

VI. SUMMARY

A modified bit-flipping algorithm with bias terms (4 parameters) has been proposed (QMBF2). A CBBF with bias terms (QCBF) has also been proposed. It has been shown that both of them have better error correcting performances than original algorithms.

These results may show that it is important to keep the initial belief about received signals during decoding operations. We may apply this idea to other decoding algorithms.

ACKNOWLEDGMENT

This work was partially supported by Storage Research Consortium.

REFERENCES

- [1] D. Mackay, Good error-correcting codes based on very sparse matrices, *IEEE Trans. Inform. Theory*, vol.45, 399–431, 1999.
- [2] R.G. Gallager, *Low-Density Parity-Check Codes*, IRE Trans. Info. Theory IT-8:21-28(1962)

- [3] Ookubo, Kamabe *Improvement of bit-flipping decoding algorithm for High-dimensional parity codes* , SITA2007,Nov. 27-30,2007(in Japanese).
- [4] Guiqiang Dong, Yanan Li, Ningde Xie, Tong Zhang, Huaping Liu, "Candidate bit based bit-flipping decoding algorithm for LDPC codes", ISIT 2009, Seoul, Korea, June 28-July 3, 2009
- [5] David J.C. MacKay FRS, "<http://www.inference.phy.cam.ac.uk/mackay/>", Encyclopedia of Sparse Graph Codes, "<http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>"
- [6] X. Y. Hu, E. Eleftheriou, and D.M. Arnold "Regular and irregular progressive edge-growth tanner graphs" in Proc 2001 Global.Telecommum. Conf., vol.2, San Antonio, TX, Nov.2001, pp.995-1001
- [7] Y. Kou, S. Lin, and M. Fossorier, " Low density parity check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol.47, pp.2711-2736, Nov. 2001.