LETTER *Special Section on Information Theory and Its Applications*

Modified Algorithm on Maximum Detected Bit Flipping Decoding for High Dimensional Parity-Check Code

Yuuki FUNAHASHI[†a)], ***Student Member***, **Shogo USAMI**[†], **Ichi TAKUMI**[††], ***Members***, ***and*** **Masayasu HATA**[†††], ***Fellow***

SUMMARY We have researched high dimensional parity-check (HDPC) codes that give good performance over a channel that has a very high error rate. HDPC code has a little coding overhead because of its simple structure. It has hard-in, maximum detected bit flipping (MDBF) decoding that has reasonable decoding performance and computational cost. In this paper, we propose a modified algorithm for MDBF decoding and compare the proposed MDBF decoding with conventional hard-in decoding.
key words: *error correcting code, high dimensional parity-check code, hard-in decoding*

1. Introduction

In recent years, low-density parity-check (LDPC) codes [1]–[3], that are made from sparse graphs, have attracted much attention in coding theory and can achieve near Shannon limit performance [2]. We have researched high dimensional parity-check codes (HDPC) [4] that give good performance for channels over a very high error rate. HDPC code is constructed using a combination of single parity-check codes, and it is easy to implement the code, as well as the EG-LDPC code [5], [6], on hardware because of the simple structures. It can be said that HDPC code is subset of LDPC code. It is useful to consider a specific decoding algorithm for the structure of HDPC code.

HDPC code can use hard-in decoding to take advantage of its simple structure. Maximum detected bit flipping (MDBF) decoding is known as a hard-in decoding method for HDPC codes with reasonable decoding performance and computational cost. In this paper, we propose a modified algorithm based on MDBF decoding to both improve decoding performance and reduce computational costs. Then we perform a computer simulation to compare the proposed method with original MDBF decoding methods. Moreover we also perform bit flipping (BF) [1] decoding, which is used in hard-in decoding for Gallager's LDPC to compare them.

Manuscript received January 23, 2006.
Manuscript revised April 11, 2006.
Final manuscript received June 15, 2006.
[†]The authors are with the Dept. of Info. Eng., Meijo Univ., Nagoya-shi, 468-8502 Japan.
[††]The author is with Nagoya Inst. of Tech., Nagoya-shi, 466-8555 Japan.
[†††]The author is with Chubu Univ., Kasugai-shi, 487-8501 Japan.
a) E-mail: m0632024@ccmailg.meijo-u.ac.jp
DOI: 10.1093/ietfec/e89–a.10.2670

2. High Dimensional Parity-Check Code

2.1 Property of Code

HDPC code [4] is denoted as νD$m\mu$ code, when ν is dimension and μ is size. νD$m\mu$ code has $(\mu-1)^\nu$ information bits and $N=\mu^\nu$ codeword length including $M=N-(\mu-1)^\nu$ parity-check bits, and so the transmission rate is $R=(1-\frac{1}{\mu})^\nu$ and minimum distance between codeword is 2^ν.

2.2 A Parity-Check Matrix

HDPC code has low-density parity-check matrix, so HDPC code is a subset of LDPC code [1], [3]. Mackay shows the condition of a good LDPC code [2] as follows:

1. Parity-check matrix H generated by starting from an all-zero matrix and randomly flipping not necessarily distinct bits in each column.
2. Add to above condition, matrix H generated by randomly creating weight columns.
3. Add to above conditions, matrix H generated with weight per column and (as near as possible) uniform weight per row.
4. Add to above conditions, matrix H generated with weight per column and uniform weight per row, and no two columns having overlap greater than 1.
5. Add to above conditions, matrix H further constrained so that its bipartite graph has large girth.

HDPC code has good error correcting capability, since the code has parity-check matrix that is satisfied by 4th condition. For example, 3Dm2 code has the following:

$$H=\begin{array}{c} \\ x'_1\\x'_2\\x'_3\\x'_4\\x'_5\\x'_6\\x'_7\\x'_8\\x'_9\\x'_{10}\\x'_{11}\\x'_{12}\end{array}\begin{array}{c}\begin{array}{cccccccc}x_1&x_2&x_3&x_4&x_5&x_6&x_7&x_8\end{array}\\ \left(\begin{array}{cccccccc}
1&1&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0\\
0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&1&1\\
1&0&1&0&0&0&0&0\\
0&1&0&1&0&0&0&0\\
0&0&0&0&1&0&1&0\\
0&0&0&0&0&1&0&1\\
1&0&0&0&1&0&0&0\\
0&1&0&0&0&1&0&0\\
0&0&1&0&0&0&1&0\\
0&0&0&1&0&0&0&1
\end{array}\right)\end{array}. \quad (1)$$

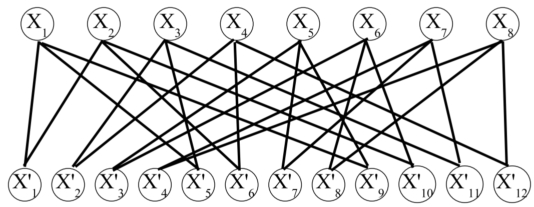

Fig. 1 Tanner graph of parity-check matrix of 3Dm2 HDPC code.

Then, the Tanner graph in Fig. 1 shows parity-check matrix of 3Dm2 HDPC code.

3. Conventional Hard-in Decoding Methods for HDPC

In this section, the conventional hard-in decoding methods of adaptive threshold (AT) decoding, MDBF decoding, and BF decoding are shown.

In this paper, when codeword length is N and length of parity is M, let $X = (x_1, x_2, \cdots, x_N)$ be transmitted codeword and $Y = (y_1, y_2, \cdots, y_N)$ be the corresponding received sequence over binary symmetric channel(BSC). Y becomes into $D = (d_1, d_2, \cdots, d_N)$ through decoding process as one of follows:

- $D = X$ (Correct decoding),
- D becomes not any codeword(Error detect),
- D becomes any other codeword except X (Erroneous decoding).

Word error rate is defined as the sum of the error detect rate and erroneous decoding rate. $D_i = (d_{i_1}, d_{i_2}, \cdots, d_{i_N})$ represents the temporary decoded codeword in the decoding process of i-th iteration, $B_i = (b_{i_1}, b_{i_2}, \cdots, b_{i_N})$ represents flip bits set of i-th iterations, $I_{\max}$ represents maximum number probability of iterations.

3.1 AT Decoding

1. $D_0 \leftarrow Y, i \leftarrow 0$.
2. D_i is checked by parity-check lines.
3. If all parity-check lines in D_i is correct, then $D \leftarrow D_i$ and finish decoding.
4. E_{i_j} is counted number over parity check lines which detects error and includes j-th bit.
5. If E_{i_j} is greater than threshold Th_i, then $B_{i_j} \leftarrow 1$, else $B_{i_j} \leftarrow 0$.
6. $D_{i+1} \leftarrow$XOR(D_i, B_i).
7. If $i < I_{\max}$, then $i \leftarrow i + 1$ and go to 2, else $D \leftarrow D_i$ and detect an error.

3.2 MDBF Decoding

1. $D_0 \leftarrow Y, i \leftarrow 0$.
2. D_i is checked by parity-check lines.
3. If all parity-check lines in D_i is correct, then $D \leftarrow D_i$ and finish decoding.
4. Each bit is counted number over belonging parity check lines which detects error.
5. If $E_{i_j} = \max_k E_{i_k}$, then $B_{i_j} \leftarrow 1$, else $B_{i_j} \leftarrow 0$.
6. $D_{i+1} \leftarrow$XOR(D_i, B_i).
7. If $i < I_{\max}$, then $i \leftarrow i + 1$ and go to 2, else $D \leftarrow D_i$ and detect an error.

3.3 BF Decoding [1], [6]

Gallager proposed BF decoding for reasonable hard-in decoding for LDPC codes. H represents $N \times M$parity-check matrix. Let $S = (s_1, s_2, \cdots, s_M)$ be the syndrome of D_i:

$$s_j = \sum_{h=1}^{N} d_{i_h} \cdot H_{hj}\ (mod\ 2). \tag{2}$$

If D_i is a codeword, S will be $\mathbf{0}$. BF decoding algorithm following.

1. $D_0 \leftarrow Y, i \leftarrow 0$.
2. S is evaluated by Eq. (3).
3. If $S = \mathbf{0}$, then $D \leftarrow D_i$ and finish decoding.
4. $E_{i_j} \leftarrow \sum_{h=1}^{M} s_h \cdot h_{jh} = 1$.
5. E_{i_j} is greater than threshold, then $B_{i_j} \leftarrow 1$, else $B_{i_j} \leftarrow 0$.
6. $D_{i+1} \leftarrow$ XOR (D_i, B_i).
7. If $i < I_{\max}$, then $i \leftarrow i + 1$ and go to 2, else $D \leftarrow D_i$ and detect an error.

3.4 Comparison

MDBF decoding is an improvement on AT decoding. AT decoding is a problem because the threshold of each turn Th_i cannot be determined by systematic means. There is no problem decoding threshold in MDBF decoding since the threshold is determined as the maximum E_{i_j}. In a fixed threshold situation, when Th_i is constant in the decoding process, decoding performance of BF decoding is equivalent to that of AT decoding. However, AT decoding is specific to HDPC code at least in the viewpoint of computational cost.

4. Modified Algorithm on Maximum Detected Bit Flipping Decoding

We propose a modified algorithm based on MDBF decoding for both improving performance and reducing computational cost.

In this section, Hamming distance A, B is denoted as $d_H(\mathrm{A}, \mathrm{B})$.

4.1 Modification to Improve Performance

4.1.1 Decoding Radius

Whatever $d_H(D_i, Y)$ is, the decoding process in the conventional method continues. Therefore, the decoding result might have less likelihood from received word. We propose to bound $d_H(D_i, Y)$. When $d_H(D_{i+1}, Y)$ is greater than decoding radius r, the flip bit set B_i is limited by approaching Y in that round. In the algorithm the following process Step A. is added to the process after Step 4.:

A. If $d_H(\text{XOR}(D_i, B_i), Y) > r$,
$B_i \leftarrow \text{AND}(B_i, \text{XOR}(D_i, Y))$.

4.1.2 Bound Distance Check

The farther distance $d_H(D, Y)$ is, the higher likelihood of Erroneous decoding there is. We propose to bound $d_H(D, Y)$. When $d_H(D, Y)$ is greater than threshold of bound distance t, then decoding results in Error detect. In the following process Step B. is added to the algorithm after Step 7.:

B. If $d_H(D, Y) > t$, then detect an error.

4.2 Modification to Reduce Computational Cost

There is a never-ending case of decoding in an infinite iteration, because of loop in the decoding state. It is the simulation where a temporary decoded codeword in the decoding process of $(i + 1)$-th iteration is equal to the temporary decoded codeword in decoding process of $(i - 1)$-th iteration. So, we propose to stop decoding if the loop is detected. In the following process Step C. is added to the algorithm after Step 6.:

C. If $D_{i+1} = D_{i-1}$, then stop decoding and detect an error.

Similarly, a decoding process may not be finished if $D_{i+1} = D_{i-j}(j = 2, 3, \cdots)$, but these situations hardly occur.

5. Research on Modified Algorithm

The parameters of r and t play an important role in the improvement of the decoding performance. We compute decoding performance for 3 dimensional HDPC codes which are different sizes of 3Dm3, 3Dm5, 3Dm10 transmitted over a BSC on computer simulation using various r and t.

First, Fig. 2 shows correct rate gain from the original MDBF decoding to the proposed using decoding radius 9,10 and 11 of 3Dm3 with respect to crossover probability, where correct rate gaun means the increment of correct decoding rate by setting decoding radius and bound distance in MDBF decoding. When r is 10, the code has the most gain. Figures 3, 4 show correct rate gain using decoding radius 13, 15, 17, 19 and 21 of 3Dm5 and 20, 22, 24, 26 and 28 of 3Dm10 with respect to crossover probability, respectively. When r is 17, 26, the code has the most gain, respectively.

Second, Fig. 5 shows correct rate gain and erroneous rate gain using bound distance 7, 8 and 9 decoding radius 10 of 3Dm3 with respect to crossover probability.

When t is 7, erroneous rate gain is so large, but correct rate gain is too large to use in decoding. When t is 9, correct rate gain is small, but erroneous rate gain is also small. Therefore bound distance $t = 8$ is the most desirable for 3Dm3 code.

Figures 6, 7 show gain to change t of 3Dm5 and 3Dm10 with respect to crossover probability for MDBF decoding, respectively. When t is 15, 22, the code has most gain, respectively.

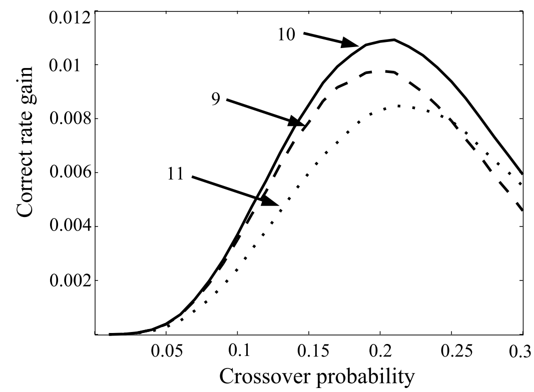

Fig. 2 Correct rate gain using decoding radius 9, 10 and 11 of 3Dm3 with respect to crossover probability.

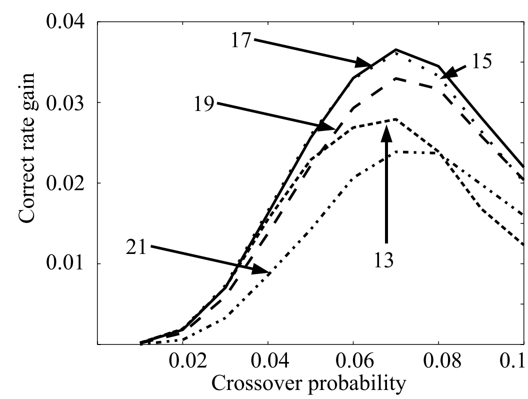

Fig. 3 Correct rate gain using decoding radius 13, 15, 17, 19 and 21 of 3Dm5 with respect to crossover probability.

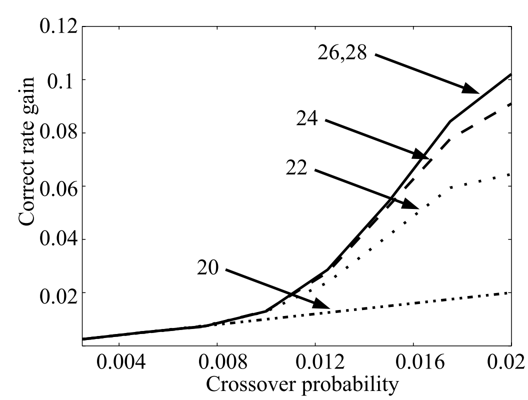

Fig. 4 Correct rate gain using decoding radius 20, 22, 24, 26 and 28 of 3Dm10 with respect to crossover probability.

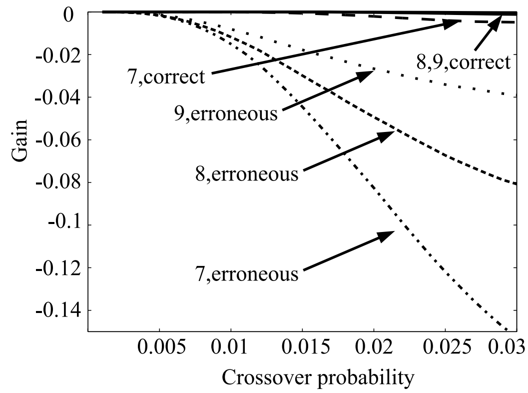

Fig. 5 Correct rate gain and erroneous rate gain using bound distance 7, 8 and 9 decoding radius 10 of 3Dm3 with respect to crossover probability.

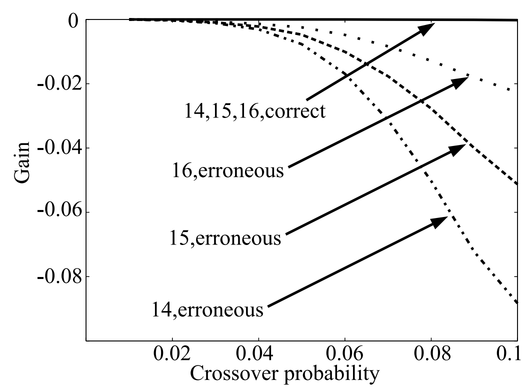

Fig. 6 Correct rate gain and erroneous rate gain using bound distance 14, 15 and 16 decoding radius 17 of 3Dm5 with respect to crossover probability.

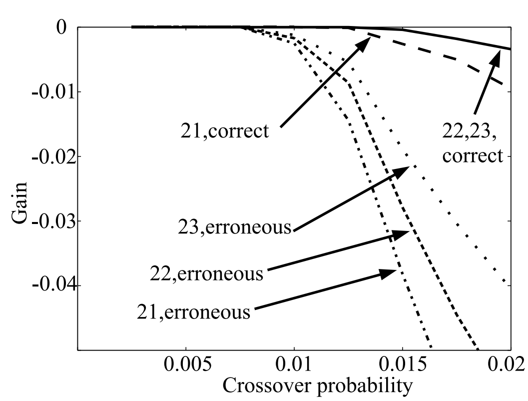

Fig. 7 Correct rate gain and erroneous rate gain using bound distance 21, 22 and 23 decoding radius 26 of 3Dm10 with respect to crossover probability.

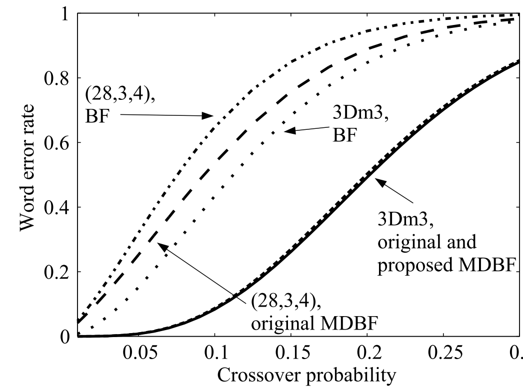

Fig. 8 Word error rate of 3Dm3 code and (28,3,4) code with respect to crossover probability.

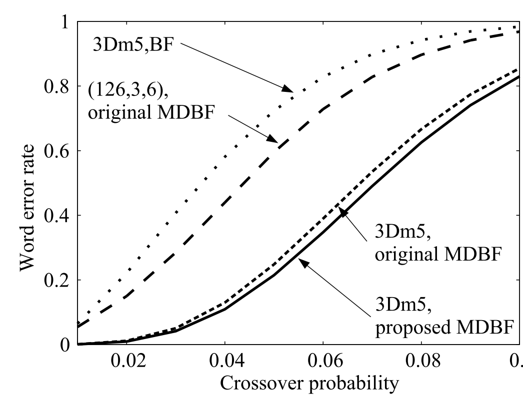

Fig. 9 Word error rate of 3Dm5 code and (126,3,6) code with respect to crossover probability.

6. Comparison with Conventional Decoding Methods

In this section, we compare proposed MDBF decoding with original MDBF decoding. We compute 3 dimensional HDPC codes which are different size of 3Dm3, 3Dm5, 3Dm10 transmitted over a BSC on computer simulation. And, we use decoding radius r and bound distance t in these simulations, which are optimum values of word error rate in pilot simulation. Moreover, we compare HDPC codes with Galager's LDPC codes with nearly transmitted rate and codeword length. Gallager's LDPC codes satisfy Gallager's design [1], [7].

According to Mackay [2], [7], Regular-LDPC code which column weight is 3 have good decoding performance for sum-product decoding. (n, w_c, w_r) Gallager's LDPC code means codeword length n, column weight w_c and row weight w_r. The threshold of BF decoding on HDPC code and LDPC code is fixed to 3 in our simulation, because of higher decoding performance than at a fixed threshold of 2.

6.1 Decoding Performance

First, decoding performance are shown. Figure 8 shows word error rate with respect to crossover probability on 3Dm3 HDPC code and (28,3,4) LDPC code for MDBF decoding and BF decoding. The HDPC code for original MDBF decoding, HDPC code for proposed MDBF decoding with $r = 10$ and $t = 8$. Is shown in Fig. 8 HDPC code for BF decoding, and LDPC code for BF decoding. LDPC code for MDBF decoding always has superior performance than BF decoding. Figures 9, 10 show word error rate with respect to crossover probability on 3Dm5 and 3Dm10 HDPC code, respectively and (126,3,6) and (1001,3,11) LDPC code, respectively. In Figs. 9, 10 proposed MDBF decoding simulated with the parameters $r = 17, t = 15$ and $r = 26, t = 22$, respectively. According to Figs. 8, 9, 10 it seems the HDPC code for the proposed MDBF decoding has the best performance in all ranges. The longer the codeword length becomes, the closer the performance of the LDPC code for MDBF decoding approaches the HDPC code. We can reduce the word error rate by virtue of process Step A.

Second, the performance improvement from the original MDBF decoding is shown. Figures 11, 12, 13 show block erroneous rate and word error rate with respect to crossover probability of original and proposed MDBF decoding on 3Dm3, 3Dm5 and 3Dm10, respectively. In Figs. 11, 12, 13, it seems that both the word error rate and block erroneous rate of proposed method are decreased from the original method. Therefore, we can say the proposed method is superior to the original method. We can reduce the block erroneous rate by virtue of process Step B.

6.2 Computational Cost

Table 1 shows the average number probability of bit operations per iteration. Proposed methods have less computational costs than original methods. Cost reduction from

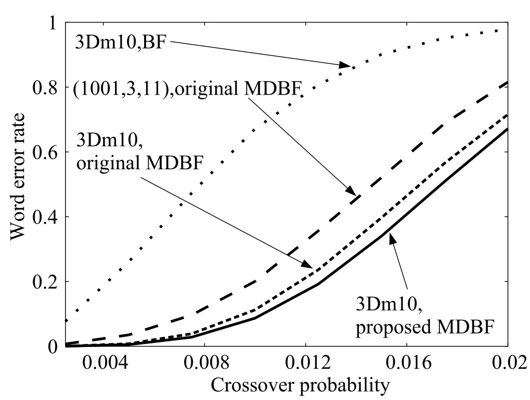

Fig. 10 Word error rate of 3Dm10 code and (1001,3,11) code with respect to crossover probability.

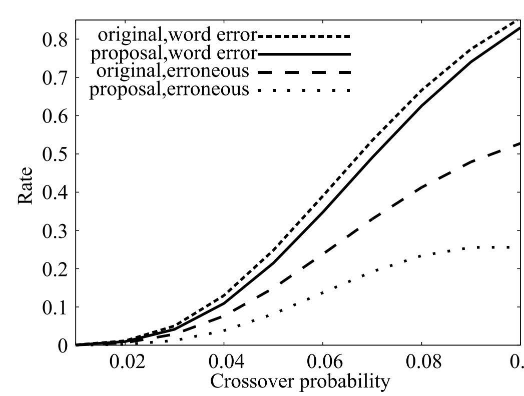

Fig. 12 Block erroneous rate and word error rate of 3Dm5 code with respect to crossover probability.

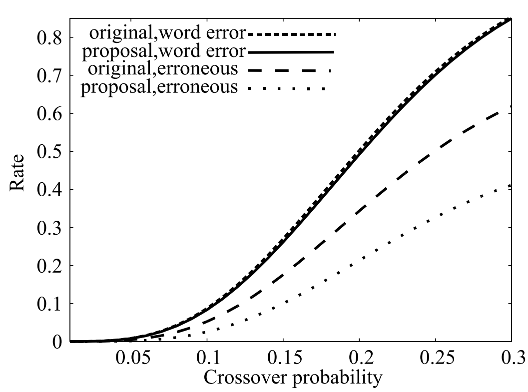

Fig. 11 Block erroneous rate and word error rate of 3Dm3 code with respect to crossover probability.

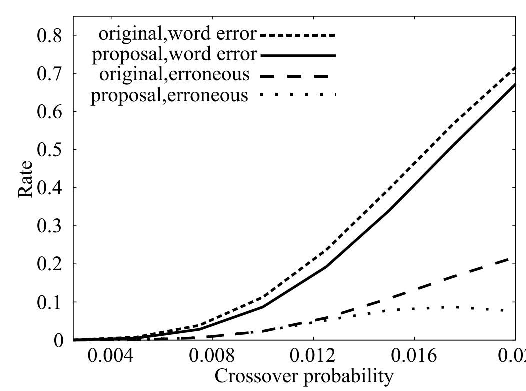

Fig. 13 Block erroneous rate and word error rate of 3Dm10 code with respect to crossover probability.

Table 1 Average number of bit operation on 3Dm3, 3Dm5 and 3Dm10 HDPC code for original and proposed MDBF decoding.

		$I_{\max}$	average number of achieving $I_{\max}$	average number of bit operations per iteration	mean times of iteration	average bit operations	cost rate on proposal / original
3Dm3	original	20	0.107	298.8	4.722	1411	
	proposal	20	0.004	368.0	3.024	1112	0.789
3Dm5	original	20	0.141	1334	5.943	7928	
	proposal	20	0.012	1724	3.876	6684	0.843
3Dm10	original	100	0.184	10390	6.547	68000	
	proposal	100	0.06	13180	3.844	50680	0.745

original to proposed methods is 15% to 25%. In general, there is trade-off between decoding performance and computational cost. But, the proposed method improves both decoding performance and computational cost. According to Table 1, the success of reducing the averages is attributable to process Step C.

7. Conclusion

In this paper, we proposed a modified algorithm for MDBF decoding that improves both decoding performance and computational cost. We introduced two parameters, decoding radius r and bound distance t, to increase correct decoding and reduce erroneous decoding. Moreover, decoding cost reduction was successful by stopping the decoding when a loop state was detected. When codeword length is less than about 1000, 3 dimensional HDPC codes show better decoding performance than Gallager's LDPC codes.

In future works, we hope to propose a method to find the optimum parameters relating to codeword length, transmission rate and channel environment using this systematic method.

References

[1] R.G. Gallager, Low-Density Parity-Check Codes, Research Monograph series, Cambridge, MIT Press, 1963.
[2] D.J.C. Mackay, "Good error-correcting codes based on very sparse matirices," IEEE Trans. Inf. Theory, vol.45, no.3, pp.399–431, 1999.
[3] T. Wadayama, Low-Density Parity-Check Codes and Its Decoding, Triceps, 2002.
[4] T. Fukuda, Y. Hamasuna, I. Takumi, M. Hata, and T. Nakanishi, "An iterative decoding of high dimensional parity code," IEICE Trans. Fundamentals, vol.E86-A, no.10, pp.2473–2482, Oct. 2003.
[5] S. Lin and D.J. Cosrello, Error Control Coding, Second ed., Person Prentice Hall, Upper Saddle River, New Jerser 07458, U.S.A., 2004.
[6] J. Zhang, J.S. Yedidia, and M.P.C. Fossoirier, "Low latency decoding of EG LDPC codes," TR-2005-036, 2005.
[7] D.J.C. Mackay, Information Therory, Inference, and Learning Algorithms, Cambridge University Press, 2003.