PAPER *Special Section on Information Theory and Its Applications*

Adaptive Decoding Algorithms for Low-Density Parity-Check Codes over the Binary Erasure Channel

Gou HOSOYA[†a)], **Hideki YAGI**[††], **Manabu KOBAYASHI**[†††], ***Members***, ***and*** **Shigeichi HIRASAWA**[††††], ***Fellow***

SUMMARY Two decoding procedures combined with a belief-propagation (BP) decoding algorithm for low-density parity-check codes over the binary erasure channel are presented. These algorithms continue a decoding procedure after the BP decoding algorithm terminates. We derive a condition that our decoding algorithms can correct an erased bit which is uncorrectable by the BP decoding algorithm. We show by simulation results that the performance of our decoding algorithms is enhanced compared with that of the BP decoding algorithm with little increase of the decoding complexity.
key words: *low-density parity-check code, belief-propagation decoding, binary erasure channel, stopping set*

1. Introduction

The combination of low-density parity-check (LDPC) codes with a belief propagation (BP) decoding algorithm over the binary erasure channel (BEC) has high decoding performance with low decoding complexity [1], [2]. The decoding complexity of the BP decoding algorithm for LDPC codes is proportional to the code length. It is well known that the BP decoding algorithm over the BEC cannot succeed when a subset of erased bit positions forms a stopping set [3]. The stopping set, which influences on the performance of the BP decoding algorithm over the BEC, is closely related to loops in the Tanner graph of LDPC codes.

To overcome a decoding failure caused by a stopping set and enhance the performance of the BP decoding algorithm, two approaches have been taken. The first approach is adding redundant rows and columns for a given parity-check matrix of a code to improve the performance of the BP decoding. This approach has been taken by K. Kasai et al. [7], S. Sankaranarayanan and B. Vasic [10], and N. Kobayashi et al. [11]. The second one is performing an adaptive procedure after the BP decoding algorithm fails in decoding. This approach has been taken by H. Pishro-Nik and F. Fekri [4], and B.N. Vellambi and F. Fekri [12]. The decoding algorithms in [4] and [12] guess the values of some erased bits to correct other erased bits.

A primary difference of these two approaches is as follows: The first approach needs the procedure of constructing a redundant parity-check matrix only once beforehand. Although the performance of the BP decoding algorithm with a redundant parity-check matrix is better than that with an original one, from the experimental results it is only effective for codes of short length [10]. On the other hand, the second approach needs to perform an adaptive decoding procedure, and this procedure is executed for each received sequence. The performance of these improved BP decoding algorithms is significantly enhanced from that of the BP decoding algorithm for codes with various lengths [2]. However these algorithms might output a wrong codeword as a decoding result.

The maximum likelihood decoding (MLD) with the Gaussian elimination (GE) is optimal, and performance gap between the BP decoding and MLD is significantly large [5]. D. Burshtein and G. Miller have proposed an MLD algorithm which is a combination of the BP decoding algorithm and the GE [8]. They have shown that MLD for LDPC codes can be efficiently performed by combining the BP decoding algorithm and the GE. Unfortunately the decoding complexity of this algorithm is proportional to the cube of the number of erased bits.

In this paper, by taking the second approach we develop two decoding algorithms of LDPC codes over the BEC which require no guessing procedures. Our decoding algorithms are also adaptive ones which perform after the BP decoding algorithm fails without producing a wrong codeword. We derive a condition that our decoding algorithms can correct an erased bit which is uncorrectable by the BP decoding algorithm. We show by simulation results that the performance of the decoding algorithms is significantly enhanced from that of the BP decoding algorithm with little increase of the decoding complexity when the erasure probability of a channel is small.

This paper is organized as follows. In Sect. 2, we describe LDPC codes, decoding for the BEC, and the BP decoding algorithm. In Sect. 3, we explain new decoding algorithms. We mention related works of our decoding algorithms in Sect. 3.1. The first decoding algorithm is presented in Sect. 3.2 and we give some properties of this decoding algorithm in Sect. 3.3. Similarly, the second decoding algorithm is presented in Sect. 3.4. An overview of the whole procedures of both algorithms is explained in Sect. 3.5. Fi-

Manuscript received January 16, 2009.
Manuscript revised April 26, 2009.
[†]The author is with the Faculty of Science and Engineering, Waseda University, Tokyo, 169-8555 Japan.
[††]The author is with Center for Frontier Science and Engineering, the University of the Electro-Communications, Chofu-shi, 182-8585 Japan.
[†††]The author is with the Faculty of Engineering, Shonan Institute of Technology, Fujisawa-shi, 251-8511 Japan.
[††††]The author is with the Waseda Research Institute for Science and Engineering, Waseda University, Tokyo, 169-8555 Japan, and is with Cyber University, Japan.
a) E-mail: hosoya@m.ieice.org
DOI: 10.1587/transfun.E92.A.2418

nally, some simulation results and discussions are presented in Sect. 4 and the conclusion is given in Sect. 5.

2. Preliminaries

2.1 LDPC Codes

Let $H = [H_{mn}]$, $m \in [1, M]$, $n \in [1, N]$, be a parity-check matrix of an LDPC code whose row and column lengths are M and N, respectively[†]. Let $\boldsymbol{c} = (c_1, c_2, \ldots, c_N) \in F_2^N$ be a codeword of an LDPC code where F_2 denotes the binary Galois field with elements $\{0, 1\}$. Then we have $\boldsymbol{c}H^{\mathrm{T}} = \mathbf{0}$ for all $\boldsymbol{c}$. Let ρ_i and λ_i denote the fractions of element ones in H of rows and columns of Hamming weight i, respectively. Let $d_{\max}$ and $c_{\max}$ be the maximum weights of rows and columns, respectively. Let $\rho(x) = \sum_{i=2}^{d_{\max}} \rho_i x^{i-1}$ and $\lambda(x) = \sum_{i=2}^{c_{\max}} \lambda_i x^{i-1}$ be the weight distribution functions of rows and columns in H, respectively. The number of rows, M, in H is given by $M = N \int_0^1 \rho(x)dx / \int_0^1 \lambda(x)dx$. In this paper, we deal with binary $C(N, \lambda(x), \rho(x))$ LDPC codes. The design rate R' of $C(N, \lambda(x), \rho(x))$ LDPC codes is given by $R' = 1 - \frac{M}{N}$. The actual rate R of the codes satisfies $R \geq R'$ since H is not necessarily a full rank matrix.

We define a loop of length $2L$, $L \geq 2$, in H as a closed path consisting of the elements one in H at positions (m_1, n_1), (m_1, n_2), (m_2, n_2), …, (m_L, n_L), and (m_L, n_1) where both $m_1, m_2, \ldots, m_L$ and $n_1, n_2, \ldots, n_L$ are distinct. We call the minimum length of loops in H the *girth* of H.

2.2 Decoding for the BEC

We assume a codeword $\boldsymbol{c}$ is transmitted through the BEC. The codeword $\boldsymbol{c}$ is disturbed by an erased sequence $\boldsymbol{e} = (e_1, e_2, \ldots, e_N) \in \{0, \epsilon\}^N$ from the channel where ϵ denotes an erasure, and the decoder receives a sequence $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$. The addition of a binary bit and the erased bit are defined as $0 + \epsilon = \epsilon$ and $1 + \epsilon = \epsilon$. Therefore the received bits are either erased (unknown) or known bits.

Let $\mathcal{N} = [1, N]$ be an index set of the code bits or that of the columns in H. Each element in both sets corresponds to each other. For some index set $\chi \subseteq \mathcal{N}$, let $\boldsymbol{c}_\chi$ and H_χ denote a subsequence of $\boldsymbol{c}$ and a submatrix of H indexed by χ, respectively. In this paper, we express the column positions in H_χ by the same indices of H. For example, let $\mathcal{N} = [1, 8]$ and $\chi = \{2, 4, 6, 8\}$, hence the position of the leftmost column of H_χ is indexed by not 1 but 2. Let $\overline{\chi} = \mathcal{N} \setminus \chi$ be the complement set of χ.

From the definition of a parity-check matrix H, we can write

$$\boldsymbol{c}H^{\mathrm{T}} = \boldsymbol{c}_{\mathcal{E}}H_{\mathcal{E}}^{\mathrm{T}} + \boldsymbol{c}_{\overline{\mathcal{E}}}H_{\overline{\mathcal{E}}}^{\mathrm{T}} = \mathbf{0}, \tag{1}$$

where $\mathcal{E} \subseteq \mathcal{N}$ denotes the index set of erased bits. Since $\boldsymbol{c}_{\overline{\mathcal{E}}}$, $H_{\overline{\mathcal{E}}}$, and $H_{\mathcal{E}}$ are known to a decoder, we can rewrite Eq. (1) as

$$\boldsymbol{c}_{\mathcal{E}}H_{\mathcal{E}}^{\mathrm{T}} = \boldsymbol{c}_{\overline{\mathcal{E}}}H_{\overline{\mathcal{E}}}^{\mathrm{T}} = \boldsymbol{s}, \tag{2}$$

where $\boldsymbol{s} = (s_1, s_2, \ldots, s_M) \in \{0, 1\}^M$ is a syndrome sequence obtained by calculating $\boldsymbol{c}_{\overline{\mathcal{E}}}H_{\overline{\mathcal{E}}}^{\mathrm{T}}$. Therefore decoding for the BEC is to calculate the erased (unknown) sequence $\boldsymbol{c}_{\mathcal{E}}$ from the simultaneous equations $\boldsymbol{c}_{\mathcal{E}}H_{\mathcal{E}}^{\mathrm{T}} = \boldsymbol{s}$. Since $\boldsymbol{c}$ is a codeword, $\boldsymbol{c}_{\mathcal{E}}H_{\mathcal{E}}^{\mathrm{T}} = \boldsymbol{s}$ has at least one solution. If this equation has multiple solutions, then it cannot be corrected, which results in error detection. MLD for the BEC can correct the received sequence optimally by using the GE while the decoding complexity is proportional to the cube of the number of erased bits [8].

2.3 BP Decoding Algorithm [2]

We define the following sets for all (m, n), $m \in [1, M]$, $n \in [1, N]$, such that $H_{mn} = 1$:

$$\mathcal{A}(m) = \{n \mid H_{mn} = 1\}, \ \mathcal{B}(n) = \{m \mid H_{mn} = 1\}.$$

Since this paper focuses on LDPC codes, the size of $\mathcal{A}(m)$ and $\mathcal{B}(n)$ are assumed to be small values. Let $\boldsymbol{s}^{\mathrm{B}} = (s_1^{\mathrm{B}}, s_2^{\mathrm{B}}, \ldots, s_M^{\mathrm{B}})$ and $\mathcal{E}_{\mathrm{B}}$ be a syndrome sequence and a set of erased bit positions during an execution of the BP decoding algorithm, respectively. The values of $\boldsymbol{s}^{\mathrm{B}}$ and $\mathcal{E}_{\mathrm{B}}$ are initialized to $\boldsymbol{s}^{\mathrm{B}} := \boldsymbol{s}$ and $\mathcal{E}_{\mathrm{B}} := \mathcal{E}$ at the beginning of the algorithm, respectively. For some $\mathcal{E}_{\mathrm{B}} \subseteq \mathcal{N}$, let $\Psi_{\mathrm{B}}(m)$, $m \in [1, M]$, be an index set of the column positions of element ones at row m in $H_{\mathcal{E}_{\mathrm{B}}}$ during an execution of the BP decoding algorithm. The values of $\Psi_{\mathrm{B}}(m)$ are initialized to $\Psi_{\mathrm{B}}(m) := \{\mathcal{A}(m) \cap \mathcal{E}\}$ at the beginning of the algorithm.

We can rewrite $\boldsymbol{c}_{\mathcal{E}}H_{\mathcal{E}}^{\mathrm{T}} = \boldsymbol{s}$ as

$$\sum_{i \in \Psi_{\mathrm{B}}(m)} c_i = s_m^{\mathrm{B}}, \qquad m \in [1, M]. \tag{3}$$

Note that s_m^{B} is obtained by calculating

$$s_m^{\mathrm{B}} = \sum_{i \in \mathcal{A}(m) \setminus \Psi_{\mathrm{B}}(m)} c_i. \tag{4}$$

From Eq. (4), each iteration of the BP decoding algorithm can correct an erased bit c_i, $i \in \Psi_{\mathrm{B}}(m)$, iff $|\Psi_{\mathrm{B}}(m)| = 1$. In other words, the BP decoding algorithm can correct an erased bit c_i whenever a row of weight one exists in $H_{\mathcal{E}_{\mathrm{B}}}$. The algorithm continues the procedure of calculating erased bits until all the erased bits are corrected or there is no $m \in [1, M]$ satisfying $|\Psi_{\mathrm{B}}(m)| = 1$.

The BP decoding algorithm over the BEC constitutes the following procedures:

[BP Decoding Algorithm over the BEC]

B1 (Initialization): Set $\boldsymbol{s}^{\mathrm{B}} := \boldsymbol{s}$ and $\mathcal{E}_{\mathrm{B}} := \mathcal{E}$. For $m \in [1, M]$, set $\Psi_{\mathrm{B}}(m) := \{\mathcal{A}(m) \cap \mathcal{E}_{\mathrm{B}}\}$.

B2 (Decision of Decoding Failure): If there exists $m \in [1, M]$ such that $|\Psi_{\mathrm{B}}(m)| = 1$, then go to B3. Otherwise the algorithm fails.

B3 (BP Step): For any $m \in [1, M]$ such that $|\Psi_{\mathrm{B}}(m)| = 1$,

[†]For two integers i and j ($i \leq j$), $[i, j]$ denotes the set of integers from i to j.

perform the followings:

B3-1 (Correct an Erased Bit): Let i be an integer such that $\Psi_B(m) = \{i\}$, set $c_i := s^B_m$, $\mathcal{E}_B := \mathcal{E}_B \setminus \{i\}$, and $\Psi_B(m) := \Psi_B(m) \setminus \{i\}$.

B3-2 (Row Operation): For all $m' \in \mathcal{B}(i) \setminus \{m\}$, set $\Psi_B(m') := \Psi_B(m') \setminus \{i\}$ and $s^B_{m'} := s^B_{m'} + s^B_m$.

B4 (Decision of Decoding Success): If $\mathcal{E}_B \neq \emptyset$, then go to B2. Otherwise the algorithm succeeds. □

The BP decoding algorithm over the BEC fails when a set of the erased bit positions contains a stopping set.

Definition 1 (Stopping set [3]): For some $\mathcal{S} \subseteq \mathcal{E}$, $\mathcal{S} \neq \phi$, we call $\mathcal{S}$ a stopping set if the weights of all rows in a submatrix $H_{\mathcal{S}}$ are zero or at least two. □

The above definition is slightly modified from the original one in [3], since we do not consider a non-empty stopping set. At the step B2 in the BP decoding algorithm, the algorithm stops when there does not exist $m \in [1, M]$ satisfying $|\Psi_B(m)| = 1$. At this time, weights of all rows in $H_{\mathcal{E}_B}$, are zero or at least two. Therefore $\mathcal{E}_B$ is a stopping set[†] $\mathcal{S}$.

Since the parity-check matrix H of LDPC codes is sparse, the weight of each row has a small value. From the above fact, the BP decoding algorithm can perform for LDPC codes effectively with the decoding complexity proportional to the number of ones in $H_{\mathcal{E}}$. Hence the BP decoding algorithm is efficient for LDPC codes.

3. Proposed Decoding Algorithms

In this section, we develop two decoding algorithms of LDPC codes over the BEC which perform adaptive decoding procedures after the BP decoding algorithm stops in failure.

3.1 Relation with Other Methods

As was mentioned in Sect. 1, to overcome a decoding failure caused by a stopping set, two approaches have been studied. This paper takes the second approach and the proposed decoding algorithms considered here perform an adaptive procedure after the BP decoding algorithm fails [4], [9], [12].

An adaptive procedure is conducted for each transmitted codeword. The decoding performance of these decoding algorithms is significantly improved compared with that of the BP decoding algorithm for codes of various lengths [2]. The decoding algorithms by Pishro-Nik and Fekri [4], and Vellambi and Fekri [12] guess the values of erased bits which are not corrected by the BP decoding algorithm. Clearly the performance of these algorithms depends on the number of guessed bits and the way of choosing these bits. Unfortunately, it is possible for these decoding algorithms to generate a wrong codeword as a decoding result.

Our two decoding procedures do not take guessing procedures and hence it does not result in "mis-correction." They use the rows of weight greater than one to eliminate the loops which satisfy a certain condition and make it possible to go back to the BP decoding algorithm.

3.2 Decoding Algorithm A

Recall that when the BP decoding algorithm stops in failure, the set of erased bit positions is represented by $\mathcal{E}_B$. Let $\boldsymbol{s}^P = \boldsymbol{c}_{\overline{\mathcal{E}}_B} H^T_{\overline{\mathcal{E}}_B}$. The decoding algorithm A tries to solve a simultaneous equation $\boldsymbol{c}_{\mathcal{E}_B} H^T_{\mathcal{E}_B} = \boldsymbol{s}^P$. From Definition 1, weights of all rows in $H_{\mathcal{E}_B}$ are zero or at least two since $\mathcal{E}_B$ is a stopping set $\mathcal{S}$.

3.2.1 Overview of the Algorithm

The strategy of the decoding algorithm A is to make the situation where we can go back to the BP decoding algorithm by producing the row of weight one. First we choose a row of weight two and then select one column position of the element one between them. Next this row is linearly combined with the rows having the element one in a column position that we select. If we perform row operations by using the row of weight greater than two, weight of a resulting row in general becomes larger than that of former ones. On the other hand, if we perform row operations by using the row of weight two, weight of a resulting row remains equal or decreases by two. Moreover this procedure is efficiently performed for LDPC codes.

3.2.2 Procedure of Decoding Algorithm A

Let $\boldsymbol{s}^P = (s^P_1, s^P_2, \ldots, s^P_M)$ and $\mathcal{E}_P$ be a syndrome sequence and a set of the erased bit positions during an execution of the decoding algorithm A, respectively. The values of $\boldsymbol{s}_P$ and $\mathcal{E}_P$ are initialized to $\boldsymbol{s}^P := \boldsymbol{s}^B$ and $\mathcal{E}_P := \mathcal{E}_B$ at the beginning of the procedure, respectively. Let $\mathcal{E}^S_P \subseteq \mathcal{E}_P$ and $\mathcal{M}_P \subseteq [1, M]$ be a subset of the erased bit positions $\mathcal{E}_P$ and a subset of the row positions $[1, M]$, respectively. For a given $\mathcal{E}_P \subseteq \mathcal{N}$, let $\Psi_P(m)$, $m \in [1, M]$, be an index set of column positions of the element one at a row m in $H_{\mathcal{E}_P}$. The values of $\Psi_P(m)$ are initialized to $\Psi_P(m) := \{\mathcal{A}(m) \cap \mathcal{E}_P\}$. Note that $|\Psi_P(m)| \neq 1$ for all m. For $n \in \mathcal{E}_P$, let $\Delta_P(n)$ be an index set of row positions of the element one at a column n of $H_{\mathcal{E}_P}$. The values of $\Delta_P(n)$ are initialized to $\Delta_P(n) := \mathcal{B}(n)$.

At first, we choose a row m such that $|\Psi_P(m)| = 2$ where $\Psi_P(m) = \{i_1, i_2\}$. The row m can be written as $c_{i_2} = c_{i_1} + s^P_m$, meaning that the erased bit c_{i_2} can be represented by only c_{i_1} so that we perform row operation using the row m with the rows of $m' \in \Delta_P(i_2) \setminus \{m\}$. Here we can choose either i_1 or i_2 in any order from $\Psi_P(m)$ at this step, since this choice does not influence on the decoding result. More precious reason is mentioned after the explanation of the algorithm. If the row m' satisfies $\Psi_P(m') \supseteq \{i_1, i_2\}$, then the size of resulting $\Psi_P(m')$ decreases by two. Clearly the erased bit c_{i_2} and a row m are not needed in the subsequent row operations until the

[†]Note that at this time, $\mathcal{E}_B$ equals to the set of erased bit positions which cannot be corrected by the BP decoding algorithm.

erased bit c_{i_1} is corrected, so we use a set of the rows $\mathcal{M}_P(:= \mathcal{M}_P \setminus \{m\})$ and a set of erased bits at $\mathcal{E}_P^S(:= \mathcal{E}_P^S \setminus \{i_2\})$ for the next row operation. By decreasing $|\Psi_P(m')|$, this procedure happens to make $|\Psi_P(m')| = 1$. Then we can proceed the BP decoding algorithm. We continue these procedures until all erased bits are corrected or there are no rows $m \in \mathcal{M}_P$ of the weight one or two.

The decoding algorithm A consists of the BP decoding and the procedure A. The main difference between the BP decoding algorithm and the procedure A is the step P3.

[Procedure A]

P1 (Initialization): Set $\mathcal{E}_P := \mathcal{E}_B$ and $\mathcal{E}_P^S := \mathcal{E}_P$. For $m \in [1, M]$, set $\Psi_P(m) := \{\mathcal{A}(m) \cap \mathcal{E}_B\}$. For $n \in \mathcal{E}_P$, set $\Delta_P(n) := \mathcal{B}(n)$. Set $\mathcal{M}_P := \{m \mid m \in [1, M], |\Psi_P(m)| \geq 2\}$.

P2 (Decision of Decoding Failure): If there exists $m \in [1, M]$ such that $|\Psi_P(m)| = 1$, then go to P4. If $|\mathcal{M}_P| = 0$, then the algorithm fails. If there exists $m \in \mathcal{M}_P$ such that $|\Psi_P(m)| = 2$, then go to P3. Otherwise, the algorithm fails.

P3 (Decoding A Step): For all $m \in \mathcal{M}_P$ such that $|\Psi_P(m)| = 2$, perform the followings:

- **P3-1 (Row Operation):** For all $m' \in \Delta_P(j) \setminus \{m\}$ where $\Psi_P(m) = \{i, j\}$, set $\Psi_P(m') := \{\Psi_P(m') \cup \Psi_P(m)\} \setminus \{\Psi_P(m') \cap \Psi_P(m)\}$ and $s_{m'}^P := s_{m'}^P + s_m^P$. No matter of how to choose i and j from $\Delta_P(m)$. If $|\Psi_P(m')| = 0$, then set $\mathcal{M}_P := \mathcal{M}_P \setminus \{m'\}$.
- **P3-2 (Updating Column Index Sets):** Set $\Delta_P(i) := \{\Delta_P(i) \cup \Delta_P(j)\} \setminus \{\Delta_P(i) \cap \Delta_P(j)\}$, $\mathcal{M}_P := \mathcal{M}_P \setminus \{m\}$, and $\mathcal{E}_P^S := \mathcal{E}_P^S \setminus \{j\}$.

P4 (BP Step): For all $m \in [1, M]$ such that $|\Psi_P(m)| = 1$, perform the followings.

- **P4-1 (Correct an Erased Bit):** For i such that $\Psi_P(m) = \{i\}$, set $c_i := s_m^P$, $\mathcal{E}_P := \mathcal{E}_P \setminus \{i\}$, and $\Psi_P(m) := \Psi_P(m) \setminus \{i\}$.
- **P4-2 (Row Operation):** For all $m' \in \Delta_P(i) \setminus \{m\}$, set $\Psi_P(m') := \Psi_P(m') \setminus \{i\}$, $s_{m'}^P := s_{m'}^P + s_m^P$, and $\mathcal{E}_P^S := \mathcal{E}_P^S \setminus \{i\}$.

P5 (Decision of Decoding Success): If $\mathcal{E}_P \neq \emptyset$, then go to P2. Otherwise the algorithm succeeds. □

Clearly row operation at P3-1 is equivalent to combining two columns i and j. Therefore choosing either i_1 or i_2 from $\Psi_P(m)$ does not influence on a result of the decoding.

From the above algorithm, we have the following lemma:

Lemma 1: During an execution of the decoding algorithm A, $|\Psi_P(m)|$ remains equal or decreases by two for $m \in \mathcal{M}_P$ in step P3. □

Note that performing the decoding algorithm A for $H_{\mathcal{E}_P}$ is equivalent to perform the decoding algorithm A for the condensed matrix H^P defined as follows.

Definition 2 (Condensed Matrix): Let $H^P = [H_{mn}^P]$, $m \in \mathcal{M}_P$, $n \in \mathcal{E}_P^{S\dagger}$. We set $H_{mn}^P = 1$ if some (m, n) satisfies $n \in \Psi_m^P$, otherwise we set $H_{mn}^P = 0$. This matrix H^P is called the condensed matrix. □

The size of a condensed matrix H^P, $|\mathcal{M}_P| \times |\mathcal{E}_P^S|$, is smaller than or equal to the original matrix $H_{\mathcal{E}_B}$. It is obvious from the previous argument that once all erased bits in positions $\mathcal{E}_P^S$ are corrected, remaining erased bits in positions $\mathcal{E}_P \setminus \mathcal{E}_P^S$ can also be corrected. We explain this reason using the following example.

Example 1: We perform decoding algorithm A for a matrix

$$H_{\mathcal{E}_P} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

if $\mathcal{E}_P = \{1, 2, 3, 4\}$ and $\mathcal{M}_P = \{1, 2, 3, 4\}$. This time, $H^P = H_{\mathcal{E}_P}$ and $\mathcal{E}_P^S = \mathcal{E}_P$. Then we choose the second row and perform row operations. Here we choose a bit position $i_2 = 2$ ($i_1 = 3$) and substitute $c_2 = c_3 + s_2^P$ into rows $\Delta_P(2) = \{1, 2, 4\} \setminus \{2\} = \{1, 4\}$. By this procedure, $H_{\mathcal{E}_P}$ and H^P are transformed as follows:

$$H_{\mathcal{E}_P} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad H^P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix},$$

with $\mathcal{M}_P = \{1, 2, 3, 4\} \setminus \{2\} = \{1, 3, 4\}$ and $\mathcal{E}_P^S = \{1, 2, 3, 4\} \setminus \{2\} = \{1, 3, 4\}$. □

In the above example, H^P of the second row and the second column are ignored. Note that an erased bit c_2 will never be evaluated after an erased bit c_1 becomes a known bit. Therefore performing decoding algorithm A for $H_{\mathcal{E}_P}$ to correct erasures in $\mathcal{E}_P$ is equivalent to performing it for H^P to correct those in $\mathcal{E}_P^S$.

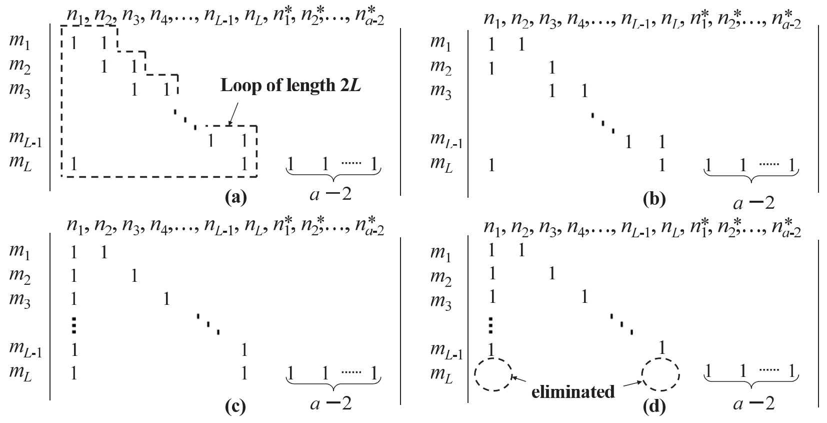
Remark 1: The decoding algorithm A fails when $|\mathcal{M}_P| = 0$. This case implies that the rank of the condensed matrix is smaller than the number of its columns and the remaining erased bits cannot be corrected even by the MLD. The algorithm also fails when weights of all rows in H^P are greater than or equal to three. In this case, the steps P3 or P4 in the algorithm cannot be performed afterward. □

3.3 Some Properties for Decoding Algorithm A

We show a condition such that the decoding algorithm A at step P3 can correct erased bits. Note that a parity-check matrix may have the duplicated row vectors, which do not influence on the performance over the BEC, so we ignore all but one of the same rows.

Lemma 2[6]: Assume that both rows and columns of a parity-check matrix H have the minimum weight at least two. For some stopping set $\mathcal{S}$, the matrix $H_{\mathcal{S}}$ has at least one loop. □

†For avoiding a confusion, we assume that the index sets of row and column positions of the condensed matrix H^P are the same as those of the original matrix H.



$n_1, n_2, n_3, n_4, \ldots, n_{L-1}, n_L, n_1^*, n_2^*, \ldots, n_{a-2}^*$
m_1
m_2
m_3
m_{L-1}
m_L
Loop of length $2L$
$a-2$
(a)
(b)
(c)
(d)
eliminated

Fig. 1 An example of the DC in Definition 3 and an assumption in the proof of Lemma 2. **(a):** The positions (m_1, n_1), (m_1, n_2), (m_2, n_2), …, (m_L, n_L), and (m_L, n_1) form a loop of length $2L$. **(b):** The result after performing the row operation for rows of m_1 and m_2. **(c):** The result before performing row operation for m_{L-1} and m_L. **(d):** The result after performing row operation for m_{L-1} and m_L and obtain $\Psi_{\mathrm{P}}(m_L) = \mathcal{N}^*$.

The above lemma shows that if there are no loops in a submatrix of H, then the index set of bit positions of this submatrix is not a stopping set. The key idea of the decoding algorithm A is to eliminate loops in H^{P}. The row operations in the step P3 sometimes yield $\Psi_{\mathrm{P}}(m)$ for $m \in \mathcal{M}_{\mathrm{P}}$ such that $|\Psi_{\mathrm{P}}(m)| = 1$ and we can calculate an erased bit at position i, $\Psi_{\mathrm{P}}(m) = \{i\}$, by BP decoding (the step P4).

From Lemma 1, the decoding algorithm A does not increase $|\Psi_{\mathrm{P}}(m)|$ for all $m \in M$. We then show a sufficient condition such that this algorithm can decrease $|\Psi_{\mathrm{P}}(m)|$ by two at the step P3. We first define a condition as follows:

Definition 3 (Diminishable Condition): When there exist all distinct $m_1, m_2, \ldots, m_L \in \mathcal{M}_{\mathrm{P}}$ and all distinct $n_1, n_2, \ldots, n_L, n_1^*, n_2^*, \ldots, n_{a-2}^* \in \mathcal{E}_{\mathrm{P}}^{\mathrm{S}}$, $a > 2$, such that $\Psi_{\mathrm{P}}(m_i) = \{n_i, n_{i+1}\}$, $i \in [1, L-1]$ for $L \geq 2$ and $\Psi_{\mathrm{P}}(m_L) = \{n_1, n_L, n_1^*, n_2^*, \ldots, n_{a-2}^*\}$ where $|\Psi_{\mathrm{P}}(m_L)| = a$, we say Diminishable Condition (DC) is satisfied. □

An example satisfying DC is shown in Fig. 1(a). From the definition of DC, rows $m_1, m_2, \ldots, m_L$ and columns $n_1, n_2, \ldots, n_L$ form a loop of length $2L$. From Lemma 3.3, the matrix $H_{\mathcal{E}_{\mathrm{B}}}$ always has at least one loop and may satisfy DC.

Lemma 2: At step P3, the decoding algorithm A can decrease $|\Psi_{\mathrm{P}}(m_L)|$ from a to $a-2$ by row operations using the rows $m_1, m_2, \ldots, m_{L-1}$ when the DC in Definition 3 is satisfied.

Proof: We perform the row operation by linearly combining a row m_1 with a row m_2 both of which have the element one at position $j = n_2$. Then $\Psi_{\mathrm{P}}(m_2)$ is modified from $\{n_2, n_3\}$ to $\{n_1, n_3\}$. We can see such an example in Fig. 1(a) and this result in Fig. 1(b). Likewise, we perform the row operation for rows m_2, m_3, …, m_{L-1} in order and obtain $\Psi_{\mathrm{P}}(m_L) = \{n_1^*, n_2^*, \ldots, n_{a-2}^*\}$. □

From Lemma 1, we can decrease $|\Psi_{\mathrm{P}}(m_L)|$ from a to $a-2$. We can see the result by before performing row operation for the rows m_{L-1} and m_L in Fig. 1(c) and after performing it in Fig. 1(d). When $a = 3$, we can decrease $|\Psi_{\mathrm{P}}(m_L)|$ from 3 to 1 and this fact immediately derives the following corollary.

Corollary 1: At step P3, the decoding algorithm A can correct an erased bit if the DC with $a = 3$ is satisfied. □

Next we show a necessary condition of decreasing $|\Psi_{\mathrm{P}}(m_L)|$ from a to $a-2$. Here we define the number of steps as the number of executions of row operations.

Lemma 3: If the DC with $a > 2$ and $L \geq 2$ does not satisfy for some step l during the decoding algorithm A, the algorithm always fails a decoding.

Proof: Assume that the DC does not hold at the step l but does hold at the step $l+1$. Assume that we perform row operation with some row m' such that $\Psi_{\mathrm{P}}(m') = \{n_i, n_{i'}\}$ for $i \in [2, L-1]$ and assume that $n_{i'}$ is eliminated from $\mathcal{E}_{\mathrm{P}}^{\mathrm{S}}$ such that $\mathcal{E}_{\mathrm{P}}^{\mathrm{S}} := \mathcal{E}_{\mathrm{P}}^{\mathrm{S}} \setminus \{n_{i'}\}$ at the step l. Note that the cases where $i = 1$ or $i = L$ are omitted in this proof, but it is easily derived if the cases of $i \in [2, L-1]$ holds. Then we have $\Psi_{\mathrm{P}}(m_{i-1}) = \{n_{i-1}, n_i\}$, $\Psi_{\mathrm{P}}(m_i) = \{n_i, n_{i+1}\}$ for distinct m_{i-1}, m_{i-2} and distinct n_{i-1}, n_i, n_{i+1} at step $l+1$ by substituting $c_{n_{i'}} = c_{n_i} + s_{m_{i'}}^{\mathrm{P}}$ into rows m_{i-1} and m_i. To satisfy DC at step $l+1$, $\Psi_{\mathrm{P}}(m_{i-1}) \supset \{n_{i-1}\}$ and $\Psi_{\mathrm{P}}(m_i) \supset \{n_{i+1}\}$ hold with $|\Psi_{\mathrm{P}}(m_{i-1})| = 2$ and $|\Psi_{\mathrm{P}}(m_i)| = 2$ at step l. This means that each $\Psi_{\mathrm{P}}(m_{i-1})$ and $\Psi_{\mathrm{P}}(m_i)$ has either n_i or $n_{i'}$. The values of $\Psi_{\mathrm{P}}(m_{i-1})$ and $\Psi_{\mathrm{P}}(m_i)$ take either the following four cases (1) – (4) such that (1) $\Psi_{\mathrm{P}}(m_{i-1}) = \{n_{i-1}, n_i\}$ and $\Psi_{\mathrm{P}}(m_i) = \{n_i, n_{i+1}\}$, (2) $\Psi_{\mathrm{P}}(m_{i-1}) = \{n_{i-1}, n_i\}$ and $\Psi_{\mathrm{P}}(m_i) = \{n_{i'}, n_{i+1}\}$, (3) $\Psi_{\mathrm{P}}(m_{i-1}) = \{n_{i-1}, n_{i'}\}$ and $\Psi_{\mathrm{P}}(m_i) = \{n_i, n_{i+1}\}$, and (4) $\Psi_{\mathrm{P}}(m_{i-1}) = \{n_{i-1}, n_{i'}\}$ and $\Psi_{\mathrm{P}}(m_i) = \{n_{i'}, n_{i+1}\}$. But all of these cases, DC is satisfied at the step l which contradicts the assumption. Then the lemma is proved. □

Note that the decoding algorithm A may continue the decoding process for a while even after the DC is not satisfied. It finally results in failure and H^{P} satisfies the conditions in Remark 1 at that moment.

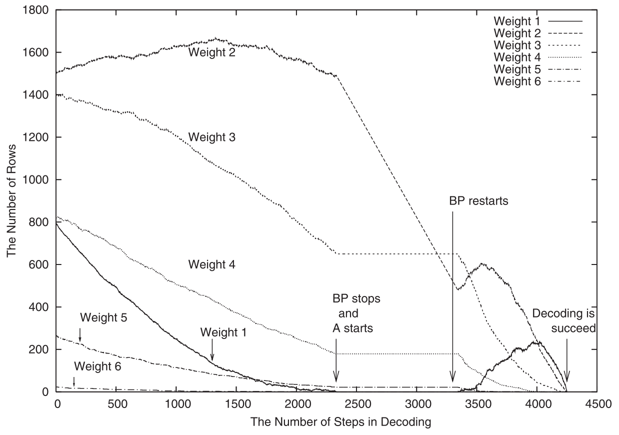


Fig. 2 The transition of the number of rows of weight $i \in [1, 6]$ for some received sequence during an execution of the decoding algorithm A with a code of $N = 10000$, $\lambda(x) = x^2$, and $\rho(x) = x^5$.

From the above argument, we have the following theorem.

Theorem 1: Without increasing the row weight in all rows of H^{P} (or equivalently $H_{\mathcal{E}_{\mathrm{P}}}$), row weight a, $a > 2$, can be decreased by two in the decoding algorithm A iff the DC holds.

Proof: Obvious from Lemmas 1, 2, and 3.

The decoding algorithm A is just a substitution procedure and does not guess the value of any erased bit. Therefore it does not produce a wrong estimate for any erased bits. Then we have the following theorem:

Theorem 2: The bit error performance of the decoding algorithm A is better than or equal to that of the BP decoding algorithm.

Proof: Obvious from the above discussion. □

Figure 2 shows transition of the number of rows of weight $i \in [1, 6]$ for some received sequence during the decoding algorithm A with a code of $N = 10000$, $\lambda(x) = x^2$, and $\rho(x) = x^5$. We can verify that the number of rows of weight two and three dominates the whole rows. The number of weight one rows decreases monotonically as the BP decoding algorithm proceeds, and it takes zero at 2331 step. This implies that the set of remaining erased bit positions is a stopping set. After 2331 step, the procedure A is executed by using the rows of weight two, and thereafter the number of rows of weight two decreases monotonically until a row of weight one is produced at 3345 step. After 3345 step, all the remaining erased bits can be corrected by the BP decoding algorithm. For other received sequences as well as this example, the decoding algorithm A produces a few number of rows of weight one can correct all the remaining erased bits.

3.4 Decoding Algorithm B

3.4.1 Overview of the Decoding Algorithm

The following observation can be made when the decoding algorithm A stops in failure:

Observation 1: The procedure A may produce new loops in H^{P} which do not exist in $H_{\mathcal{E}}$. □

Observation 1 is obvious from the fact that the procedure A changes the positions of elements one by row operations using a row of weight two in the step P3. As a result the algorithm has a possibility to produce new loops of shorter length compared with those in an original matrix $H_{\mathcal{E}}$.

Remark 1 and Observation 1 provide us with a prospect to continue decoding effectively after the decoding algorithm A stops in failure. To continue the decoding after the decoding algorithm A, we extend the idea of the decoding algorithm A such that by using rows of small weight and we produce row of weight one or two by the row operations. From Remark 1, there is no choice but to use rows of weight three after the decoding algorithm A terminates, since the minimum weight of rows in H^{P} is at least three. However, the row operations for a row of weight three and a row of weight b, $b \geq 3$, with one overlapped elements will increase the weight of a resulting row from b to $b+1$, which makes it difficult for the decoding algorithm A to resume. We show an example of such cases with $b = 3$ below:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

The weight of second row of the above matrix increases from three to four. On the other hand, if two rows of weight three and b with two overlapped elements one exist, i.e., these two rows form a loop of length four, the row operation decreases the weight of the resulting row from b to $b - 1$. We show an example when $b = 3$ below:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The second row of the above matrix decreases its weight from three to two.

We here consider the loops of length four since the aim is producing a new row of smaller weight with a reasonable complexity[†].

Let us summarize a new decoding algorithm called the decoding algorithm B. Let $H^{\mathrm{Q}} := H^{\mathrm{P}}$ be the condensed matrix when the decoding algorithm A stops in failure. Then we find, if exist, the pairs of rows which contains a loop of length four and one of whose weight is three. We perform the row operation by linearly combining each pair of rows obtained in the previous step to produce new rows of smaller weight. If a row of weight one or two is produced

[†]The complexity of searching loops of length $2T$, $T \geq 2$, is exponential in T.

in H^{Q}, then we go back to the decoding algorithm A for H^{Q}. If the decoding algorithm A stops in failure, we again proceed the same procedure for searching loops. These procedures iterate pre-determined times I or until all erased bits are corrected. If we cannot find any loops, then the decoding algorithm fails. Note that a special care must be taken not to increase the decoding complexity for the decoding. Therefore, we do not search loops in the pairs of rows which do not change from the previous iteration.

3.4.2 Procedure of Decoding Algorithm B

Let $\boldsymbol{s}^{\mathrm{Q}} = (s_1^{\mathrm{Q}}, s_2^{\mathrm{Q}}, \ldots, s_M^{\mathrm{Q}})$ and $\mathcal{E}_{\mathrm{Q}}$ be a syndrome sequence and a set of the erased bit positions during an execution of the procedure B, respectively. The values of $\boldsymbol{s}^{\mathrm{Q}}$ and $\mathcal{E}_{\mathrm{Q}}$ are initialized to $\boldsymbol{s}^{\mathrm{Q}} := \boldsymbol{s}^{\mathrm{P}}$ and $\mathcal{E}_{\mathrm{Q}} := \mathcal{E}_{\mathrm{P}}$ at the beginning of the procedure, respectively. Let $\mathcal{E}_{\mathrm{Q}}^{\mathrm{S}} \subseteq \mathcal{E}_{\mathrm{Q}}$ and $\mathcal{M}_{\mathrm{Q}} \subseteq [1, M]$ be a set of the condensed erased bit positions and a set of the condensed row positions, respectively. For some $\mathcal{E}_{\mathrm{Q}} \subseteq \mathcal{N}$, let $\Psi_{\mathrm{Q}}(m)$, $m \in [1, M]$, be an index set of column positions of the element one at row m in $H_{\mathcal{E}_{\mathrm{Q}}}$ during the procedure B. The values of $\Psi_{\mathrm{Q}}(m)$ are initialized to $\Psi_{\mathrm{Q}}(m) := \Psi_{\mathrm{P}}(m)$. For $n \in \mathcal{E}_{\mathrm{Q}}$, let $\Delta_{\mathrm{Q}}(n)$ be an index set of row positions of the element one in a column n of $H_{\mathcal{E}_{\mathrm{Q}}}$. The values of $\Delta_{\mathrm{Q}}(n)$ are initialized to $\Delta_{\mathrm{Q}}(n) := \Delta_{\mathrm{P}}(n)$.

The decoding algorithm B consists of the BP decoding, the procedure A, and the procedure B, which is performed after the decoding algorithm A stops in failure. The main difference between the procedures A and B is the step Q3, which performs searching loops of length four (the step Q3-1) and eliminates them from the matrix H^{Q} (the step Q3-2).

[Procedure B]

Q1 (Initialization): Set $\mathcal{E}_{\mathrm{Q}} := \mathcal{E}_{\mathrm{P}}$, $\mathcal{E}_{\mathrm{Q}}^{\mathrm{S}} := \mathcal{E}_{\mathrm{P}}^{\mathrm{S}}$, $\mathcal{M}_{\mathrm{Q}} := \mathcal{M}_{\mathrm{P}}$, and $H^{\mathrm{Q}} := H^{\mathrm{P}}$. Set $I \geq 1$ be some constant integer. For all $m \in [1, M]$, set $\Psi_{\mathrm{Q}}(m) := \Psi_{\mathrm{P}}(m)$ and $s_m^{\mathrm{Q}} := s_m^{\mathrm{P}}$. For $n \in \mathcal{E}_{\mathrm{Q}}^{\mathrm{S}}$, set $\Delta_{\mathrm{Q}}(n) := \Delta_P(n)$. Set $l := 1$.

Q2 (Decision of Decoding Procedures): If there exists $m \in [1, M]$ such that $|\mathcal{E}_{\mathrm{Q}}(m)| = 1$, then go to Q5. If $|\mathcal{M}_{\mathrm{Q}}| = 0$, then the algorithm fails. If there exists $m \in \mathcal{M}_{\mathrm{Q}}$ such that $|\Psi_{\mathrm{Q}}(m)| = 2$, then go to Q4. If there exists $m \in \mathcal{M}_{\mathrm{Q}}$ such that $|\Psi_{\mathrm{Q}}(m)| = 3$, then go to Q3. Otherwise, the algorithm fails.

Q3 (Decoding B Step): If $l \geq I$, then the algorithm fails. Otherwise, perform the following procedures:

Q3-1 (Searching Loops): Set $k := 0$ and go to Q3-1-1.

Q3-1-1 (Main Procedure): For a pair of rows m_0 and m_1 such that $|\Psi_{\mathrm{Q}}(m_0)| = 3$ or $|\Psi_{\mathrm{Q}}(m_1)| = 3$ where $m_0 \in \mathcal{M}_{\mathrm{Q}}$ and $m_1 \in \{m > m_0 \mid m \in \Delta_{\mathrm{Q}}(n_0), n_0 \in \Psi_{\mathrm{Q}}(m_0)\}$, confirm whether $|\Psi_{\mathrm{Q}}(m_0) \cap \Psi_{\mathrm{Q}}(m_1)| \geq 2$ holds or not. If this equation holds, then set $\mathcal{L}_k := \{m_0, m_1\}$ and $k := k + 1$. Note that when $l > 1$, the above procedure is performed for only m_0 and m_1 such that both $\Psi_{\mathrm{Q}}(m_0)$ and $\Psi_{\mathrm{Q}}(m_1)$ are updated at Q3-2 in the iteration $l - 1$.

Q3-1-2 (Decision of Procedure Continue): If step Q3-1-1 is performed for all $m_0 \in \mathcal{M}_{\mathrm{Q}}$, then set $s_l := k$ and go to Q3-2. Otherwise go to Q3-1-1.

Q3-2 (Elimination of Loops) If $s_l = 0$, then the algorithm fails. Otherwise set $k := 0$ and go to Q3-2-1.

Q3-2-1 (Row Operation): For a pair of rows m_0 and m_1 such that $\{m_0, m_1\} = \mathcal{L}_k$, if both $\Delta_{\mathrm{Q}}(m_0)$ and $\Delta_{\mathrm{Q}}(m_1)$ are not updated at step Q3-2-1 in the iteration l, then set $\Psi_{\mathrm{Q}}(m') := \{\Psi_{\mathrm{Q}}(m_0) \cup \Psi_{\mathrm{Q}}(m_1)\} \setminus \{\Psi_{\mathrm{Q}}(m_0) \cap \Psi_{\mathrm{Q}}(m_1)\}$ where $m' = m_j$, $j = \arg\max_{i \in \{0,1\}} |\Psi_{\mathrm{Q}}(m_i)|$[†]. Set $s_{m'}^{\mathrm{Q}} := s_{m_0}^{\mathrm{Q}} + s_{m_1}^{\mathrm{Q}}$ and $k := k + 1$, and go to Q3-2-2. Otherwise go to Q3-2-3.

Q3-2-2 (Updating Column Index Sets): For all $n' \in \Psi_{\mathrm{Q}}(m_0) \cap \Psi_{\mathrm{Q}}(m_1)$, set $\Delta_{\mathrm{Q}}(n') := \Delta_{\mathrm{Q}}(n') \setminus \{m'\}$. For any $n'' \in \Psi_{\mathrm{Q}}(\overline{m'}) \setminus \{\Psi_{\mathrm{Q}}(m_0) \cap \Psi_{\mathrm{Q}}(m_1)\}$ where $\overline{m'} \in \mathcal{L}_k \setminus \{m'\}$, set $\Delta_{\mathrm{Q}}(n'') := \Delta_{\mathrm{Q}}(n'') \cup \{m'\}$.

Q3-2-3 (Decision of Procedure Continue): If $k = s_l$, then go to Q4. Otherwise go to Q3-2-1.

Q4 (Decoding A Step): For all $m \in \mathcal{M}_{\mathrm{Q}}$ such that $|\mathcal{E}_{\mathrm{Q}}(m)| = 2$, perform the following procedures:

Q4-1 (Row Operation): For any $m' \in \Delta_{\mathrm{Q}}(j) \setminus \{m\}$ where $\Psi_{\mathrm{Q}}(m) = \{i, j\}$, set $\Psi_{\mathrm{Q}}(m') := \{\Psi_{\mathrm{Q}}(m') \cup \Psi_{\mathrm{Q}}(m)\} \setminus \{\Psi_{\mathrm{Q}}(m') \cap \Psi_{\mathrm{Q}}(m)\}$ and $s_{m'}^{\mathrm{Q}} := s_{m'}^{\mathrm{Q}} + s_m^{\mathrm{Q}}$. No matter of how to choose i and j from $\Delta_{\mathrm{Q}}(m)$. If $|\Psi_{\mathrm{Q}}(m')| = 0$, then set $\mathcal{M}_{\mathrm{Q}} := \mathcal{M}_{\mathrm{Q}} \setminus \{m'\}$.

Q4-2 (Updating Column Index Sets): Set $\Delta_{\mathrm{Q}}(i) := \{\Delta_{\mathrm{Q}}(i) \cup \Delta_{\mathrm{Q}}(j)\} \setminus \{\Delta_{\mathrm{Q}}(i) \cap \Delta_{\mathrm{Q}}(j)\}$, $\mathcal{M}_{\mathrm{Q}} := \mathcal{M}_{\mathrm{Q}} \setminus \{m\}$, and $\mathcal{E}_{\mathrm{Q}}^{\mathrm{S}} := \mathcal{E}_{\mathrm{Q}}^{\mathrm{S}} \setminus \{j\}$.

Q5 (BP Step): For all $m \in [1, M]$ such that $|\Psi_{\mathrm{Q}}(m)| = 1$, perform the following procedures:

Q5-1 (Correct an Erased Bit): For any i such that $\Psi_{\mathrm{Q}}(m) = \{i\}$, set $c_i := s_m^{\mathrm{Q}}$, $\mathcal{E}_{\mathrm{Q}} := \mathcal{E}_{\mathrm{Q}} \setminus \{i\}$, and $\Psi_{\mathrm{Q}}(m) := \Psi_{\mathrm{Q}}(m) \setminus \{i\}$.

Q5-2 (Row Operation): For any $m' \in \Delta_{\mathrm{Q}}(i) \setminus \{m\}$, set $\Psi_{\mathrm{Q}}(m') := \Psi_{\mathrm{Q}}(m') \setminus \{i\}$, $s_{m'}^{\mathrm{Q}} := s_{m'}^{\mathrm{Q}} + s_m^{\mathrm{Q}}$, and $\mathcal{E}_{\mathrm{Q}}^{\mathrm{S}} := \mathcal{E}_{\mathrm{Q}}^{\mathrm{S}} \setminus \{i\}$.

Q6 (Decision of Decoding Success): If $\mathcal{E}_{\mathrm{Q}} \neq \emptyset$, then go to Q2. Otherwise the algorithm succeeds. □

The procedure B tries to generate row vectors of weight one or two by linearly combining some rows of the condensed matrix H^{Q}. To show some properties of this algorithm, we give the following corollary as an immediate consequence of Lemma 2.

Corollary 2: The decoding algorithm B can correct some erased bits when the DC with $a = 3$ is satisfied for the new rows of weight two produced at step Q3 of the algorithm. Here H^{Q} corresponds to H^{P} in Lemma 2. □

An example that the decoding algorithm B correct an erased bit is depicted in Fig. 3.

The decoding algorithm B is also a substitution procedure and does not guess the value of any erased bit. There-

[†]If there exist two m', then choose arbitrarily one of them.

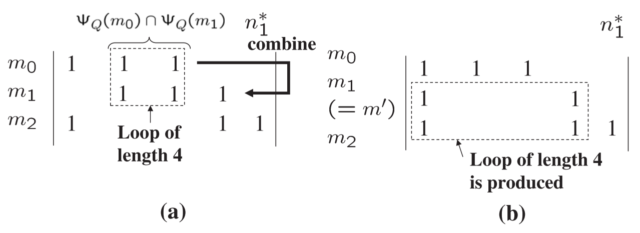


Fig. 3 An example that the decoding algorithm B corrects an erased bit n_1^*. **(a):** Rows m_0 and m_1 such that $\mathcal{L}_t = \{m_0, m_1\}$ for $t = [1, s_l]$ with $|\Psi_Q(m_0)| = 3$ and $|\Psi_Q(m_1)| = 3$ in the condensed matrix H^Q. **(b):** These rows are linearly combined and a new row $m'(= m_1)$ with $|\Psi_Q(m')| = 2$ is generated. In a result, a set of rows m_1 and m_2 forms a loop of length four and satisfies the DC of Lemma 2 with $a = 3$ and $L = 2$. From the above fact, an erased bit n_1^* can be corrected by the decoding algorithm B.

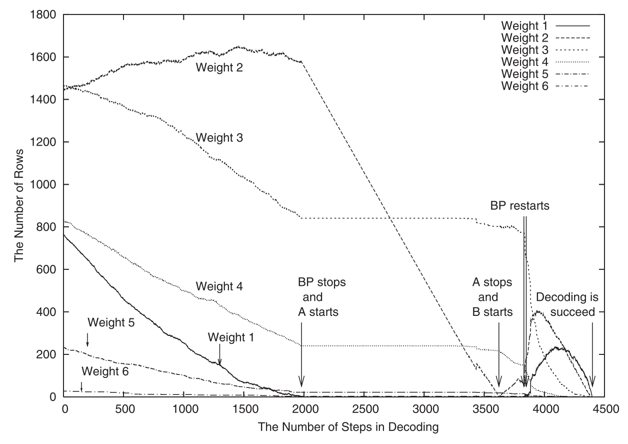


Fig. 4 The transition of the number of rows of weight $i \in [1, 6]$ for some received sequence during the decoding algorithm B with a code of $N = 10000$, $\lambda(x) = x^2$, and $\rho(x) = x^5$.

fore it does not produce a wrong estimate for any erased bits. Then we have the following theorem:

Theorem 3: The bit error performance of the decoding algorithm B is better than or equal to that of the decoding algorithm A.

Proof: It is easy to see that the decoding algorithm B has a larger decoding space than the decoding algorithm A, since it only executes after the decoding algorithm A fails in decoding. □

Figure 4 shows the transition of the number of rows of weight $i \in [1, 6]$ for some received sequence during the decoding algorithm B with a code of $N = 10000$, $\lambda(x) = x^2$, and $\rho(x) = x^5$. After 1977 step, the procedure A is executed and the number of rows of weight two decreases monotonically, and it reduces to zero at 3622 step. This implies that decoding algorithm A stops in failure since the weights of remaining rows become at least three. After 3622 step, the procedure B is executed to try to produce rows of weight two or one by using the rows of weight three. Therefore after 3622 step, the number of rows of weight three resume decreasing. At 3831 and 3836 steps, rows of weight one are produced and later all the remaining erased bits can be

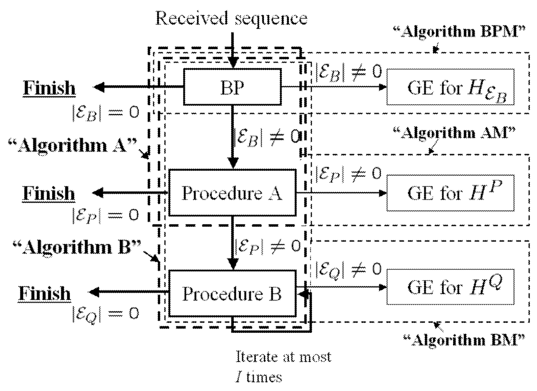


Fig. 5 Overall procedures of the decoding.

corrected by the BP decoding algorithm[†].

3.5 Overall Procedure of Decoding

We summarize an overall procedure of the decoding algorithm in Fig. 5. For a given received sequence, first the BP decoding is performed. If the BP decoding algorithm fails in decoding, we can choose termination of the BP decoding algorithm, execution of the procedure A, or execution of Gaussian Elimination (GE) [8]. We call the combinations of BP decoding algorithm with GE the algorithm BPM. If the decoding algorithm A fails in decoding, we can choose termination of the decoding algorithm A, execution of the procedure B at most I times, or execution of GE. We call the combinations of decoding algorithm A with GE the decoding algorithm AM. If the decoding algorithm B fails in decoding, we can choose termination of the decoding algorithm B or execution of GE. We call a combination of the algorithm B and GE the decoding algorithm BM.

By using the decoding algorithms AM and BM, we can usually perform MLD efficiently than by using the decoding algorithm BPM, since in these cases GE is performed for the condensed matrices (H^P or H^Q) whose columns (the number of erased bits) are less than or equal to the matrix $H_{\mathcal{E}_B}$.

Note that all of the mentioned algorithms do not result in mis-correction, therefore decoding result is either "corrected" or "error detected" but not in "error."

4. Simulation Results

4.1 Conditions for Simulation

In order to verify the performance of the proposed decoding algorithms, we show some simulation results. We use four codes denoted by C_i for $i \in [1, 4]$ whose parameters are shown in Table 1. For each parameter of codes, we generate three parity-check matrices from different seeds of random generators. In order to show the effectiveness of the decoding algorithm B, the girth of parity-check matrices of all the

[†]At 3832 step the BP decoding algorithm again stops in failure and then the procedure A resumes.

Table 1 The parameters $(N, \lambda(x), \rho(x))$ of the codes and their designed rate R' used for the simulation.

code	N	$\lambda(x)$	$\rho(x)$	R'
C_1	1000	x^2	x^5	0.5
C_2	10000	x^2	x^5	0.5
C_3	1000	x^2	x^{11}	0.75
C_4	10000	x^2	x^{11}	0.75

codes are restricted to six. Hence all codes used in the simulation have no loops of length four.

We compare the bit erasure rate (BER) and the average number of total binary operations[†] of the BP decoding algorithm [2] (denoted by "algorithm BP"), the decoding algorithm A (denoted by "algorithm A"), the decoding algorithm B (denoted by "algorithm B"), and the MLD.

For each decoding algorithm, we transmit at most 10^7 codewords over the BEC with channel erasure probability p until 300 received words are failed in decoding.

4.2 Decoding Results

Figure 6 shows the decoding performance for C_1 and C_2 and Fig. 7 shows that for C_3 and C_4. The horizontal axis and the vertical axis represent the erasure probability p of the BEC and the BER of decoding, respectively. For codes C_1 and C_3 in Figs. 6 and 7, we also show the BER of the MLD, respectively.

Figures 8 and 9 show the average numbers of total binary operations of each decoding algorithm for C_1 and C_3, respectively. Using the same codes, Tables 2 and 3 show the ratios of the average number of binary operations for searching loops in the algorithm B to that of all the procedures in the decoding. For evaluating these values in Tables 2 and 3, we averaged the number of binary operations over the received sequence which cannot be corrected by the algorithm A.

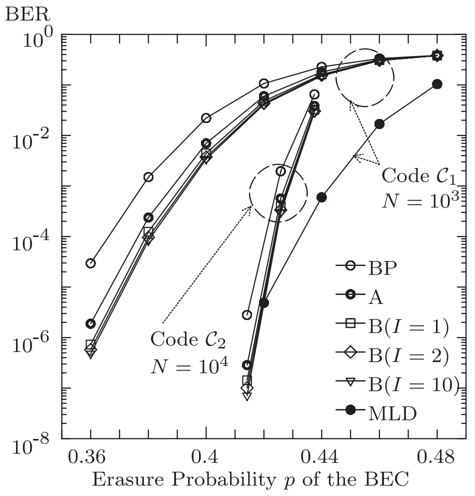


Fig. 6 Decoding performance for codes C_1 and C_2.

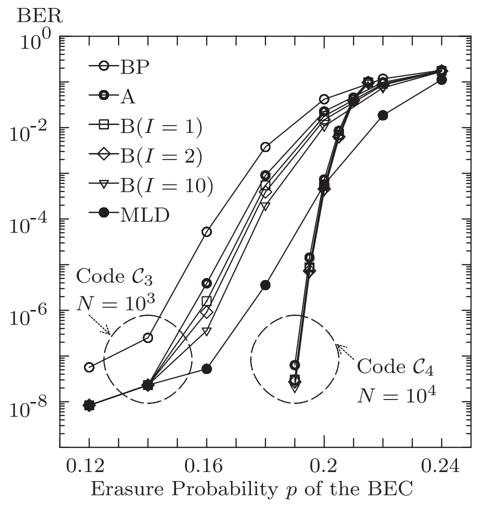


Fig. 7 Decoding performance for codes C_3 and C_4.

4.3 Discussions

4.3.1 For Codes C_1 and C_2 with $R' = 0.5$

(1) The case for small p

From Fig. 6, the BER of the algorithm B becomes smaller as I takes a large value, but this increment gets smaller as well. The performance difference between the algorithm BP and the algorithms A and B becomes large as the code length N increases. For the case of code C_1 with $p = 0.36$ in Fig. 6, the BER of the algorithm A is 6.4×10^{-2} times as large as that of the algorithm BP. The BERs of the algorithm B with $I = 1$ and 10 are 2.5×10^{-2} and 1.6×10^{-2} times lower than that of the algorithm BP, respectively. The BERs of the algorithm B are almost equal when $I \geq 2$. The performance difference between the BP based decoding and the MLD is quite large.

Figure 8 indicates that both algorithms A and B require almost the same average number of total binary operations as the algorithm BP, although the BERs of these algorithms are different as shown in Figs. 6. We explain the reasons as follows: The procedure A and the procedure B are executed

[†]Binary operations in this evaluation consist of AND and Exclusive OR operations. The number of binary operations are evaluated in the following 1) – 4):

1) Calculation of $\boldsymbol{s} = \boldsymbol{c}_{\overline{\mathcal{E}}} H_{\overline{\mathcal{E}}}^{\mathrm{T}}$ in Eq. (2)
2) Row operations of B3-2 to evaluate $\Psi_{\mathrm{B}}(m')$ and $s_{m'}^{\mathrm{B}}$
3) Row operations of P3-1 and P4-2 to evaluate $\Psi_{\mathrm{P}}(m')$ and $s_{m'}^{\mathrm{P}}$
4) Row operations of Q3-1-1, Q3-2-1, Q4-1, and Q5-2 to evaluate $\Psi_{\mathrm{Q}}(m_0) \cap \Psi_{\mathrm{Q}}(m_1)$, $\Psi_{\mathrm{Q}}(m')$, and $s_{m'}^{\mathrm{Q}}$

For the BP decoding algorithm, we evaluate the cases 1) and 2). For the decoding algorithm A, we evaluate from 1) to 3). For the decoding algorithm B, we evaluate from 1) to 4). We averaged these numbers over all the received sequence. For the calculation of $a + b$, $a, b \in \{0, 1\}$, we count a binary operation unless $a = 0$ and $b = 0$ hold at the same time. Since we apply row operations to sparse matrices, we only count binary operations for positions whose element is non-zero (this is possible because only the positions of 1's are stored, rather than the full binary representation).

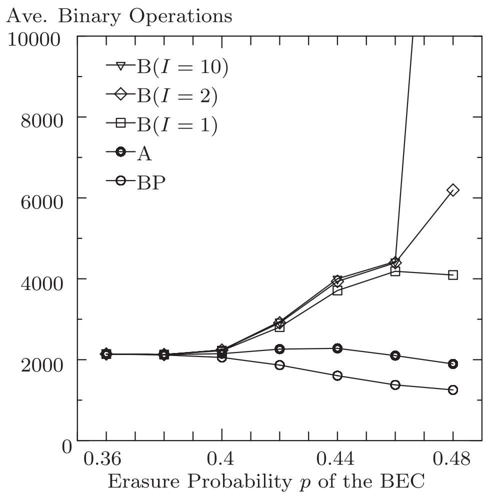


Fig. 8 The average number of the total binary operations of each decoding algorithm for code C_1. The average number of total binary operations of the algorithm B with I = 10 at p = 0.48 attains 22783.

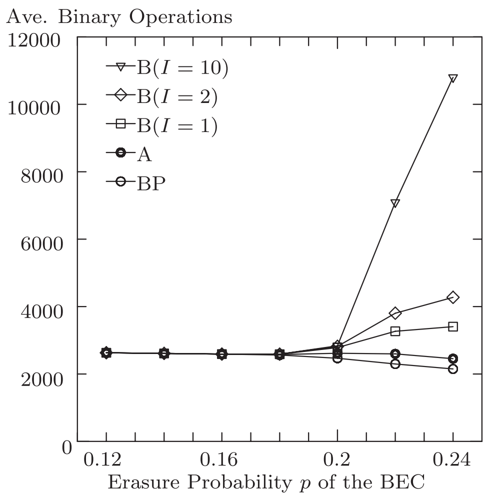


Fig. 9 The average number of the total binary operations of each decoding algorithm for code C_3.

Table 2 The ratio of the average number of binary operations for searching loops in the algorithm B to that of all the procedures in the decoding using code C_1.

p	I			
	1	2	5	10
0.36	0.417	0.442	0.447	0.448
0.38	0.416	0.447	0.455	0.456
0.4	0.419	0.453	0.461	0.462
0.42	0.432	0.469	0.477	0.477
0.44	0.461	0.494	0.500	0.500
0.46	0.496	0.518	0.520	0.520
0.48	0.517	0.679	0.840	0.913

Table 3 The ratio of the average number of binary operations for searching loops in the algorithm B to that of all the procedures in the decoding using code C_3. When p = 0.12 and 0.14, the algorithm A always stops decoding with $|\mathcal{M}_P| = 0$ which implies that the algorithm B cannot be performed and these facts are expressed as "–."

p	I			
	1	2	5	10
0.12	–	–	–	–
0.14	–	–	–	–
0.16	0.320	0.400	0.402	0.401
0.18	0.361	0.472	0.471	0.468
0.2	0.362	0.480	0.479	0.476
0.22	0.244	0.370	0.558	0.695
0.24	0.265	0.413	0.630	0.769

when the algorithm BP and the algorithm A fail in decoding, respectively. Therefore if the algorithm BP succeeds in decoding, then it does not need to continue the decoding and the procedure A is not executed. If the word erasure rate (WER) of the algorithm BP is low, then the procedures A and B are rarely executed. This fact leads a big gap of the BER performance between the algorithm A (B) and the algorithm BP with a slight increase of the binary operations when p is small. For example, the WER of the algorithm BP when $p = 0.16$ is 1.38×10^{-4} and the average number of the binary operations for the algorithm BP and the procedure A[†] are 2140.0 and 977.0, respectively. Then we have the average number of the total binary operations of the algorithm A by calculating $2140.0 + 977.0 \times 1.38 \times 10^{-4} \approx 2140.1$.

From Table 2, we verified that the number of operations for searching loops dominates much time in the overall decoding at various values of p. Once the procedure B is executed, the number of these operations drastically increases but thereafter not become large as I increases, since the algorithm performs the effective search for loops. Note that the algorithm B is rarely executed compared with the algorithm BP for small p, since most trials of decoding terminate in succeed at the algorithm BP or A. Therefore the increment of the decoding operations of the algorithm B does not affect the overall decoding operations for small p as shown in Fig. 8.

(2) The case for large p

From Figs. 6 and 8, although the average number of total binary operations of the algorithms A and B increases, the BERs of all the algorithms are almost the same. Since the number of rows of small weight becomes small and both algorithms do not satisfy the DCs in Lemma 2 and Corollary 2 as p increases, the BERs of all the algorithms are almost the same. From Fig. 8, as the number of performing the procedure B becomes large (equivalently as p increases), the number of operations for the decoding also increases. Moreover this number of operations increases as I becomes large. This is because (i) searching loops dominates much time in decoding, and (ii) the number of rows of weight three does not decrease monotonically as the decoding proceeds.

We then discuss a relationship between the decoding complexity and the value of p. The complexities of the algorithm BP and the algorithm A do not always increase with p. Their complexities increase as the number of rows of weight

[†]The average number of the binary operations for the procedure A is averaged over the received sequences which cannot be corrected by the algorithm BP.

one and/or two becomes large in the matrix H^{P}, which is often for small p. On the contrary, these algorithms cannot continue the decoding process when there do not exist such rows. As p becomes large, the number of rows of weight one and two becomes small, and in result, so do the decoding complexities.

4.3.2 For Codes C_3 and C_4 with $R' = 0.75$

From Fig. 7, the performance difference between the algorithm BP and the algorithm A and that between the algorithm BP and the algorithm B for C_3 and C_4 are smaller than those for the case of C_1 and C_2.

For the case of the code C_3 with $p = 0.12$ and $p = 0.14$, the BER of both algorithms A and B are the same as that of MLD (this fact can be verified from Table 3). This is because all the decoding trials terminate during the execution of the algorithms BP and A. We confirmed that the size of the resulting condensed matrix is zero when the algorithm A stops in failure, so the decoding could not proceed further (This case is mentioned in Remark 1.). Since the algorithm BP stops in failure with a few remaining erased bits (about six bits on average) in these cases, only a small number of adaptive binary operations (about 21 operations on average) for procedure A is required to achieve the same performance as MLD. Although the performance differences between the algorithm BP and those of the algorithms A and B become large as N increases, this difference is not larger than the case for the codes with $R' = 0.5$. Similar to the case for the codes with $R' = 0.5$, once the procedure B is executed, the number of operations for searching loops dominates the whole procedures of decoding as shown in Table 3.

4.3.3 Comparison with Other Methods

(1) Decoding Algorithm A

From Lemma 2, the algorithm A can correct an erased bit by eliminating an arbitrary length of loops in the condensed matrix H^{P}. Since this algorithm can correct an erased bit in some step of the algorithm when the DC with $a = 3$ is satisfied, the algorithm is valid for loops of arbitrary length $2L$. Hence it is valid for LDPC codes with large girth. We verified that for some received sequence, it could eliminate loops of length 20 for the code C_1 without searching such loops.

Previously, methods using an elimination technique of loops by concatenating the redundant rows and columns for a parity-check matrix H in advance, have been investigated [7], [10], [11]. In [11], a similar condition for correcting some erased bits by this concatenation has been derived. However this method has the following disadvantages compared with the algorithm A. The complexity for searching loops in H is exponential in the length of loops, which makes it difficult for this approach to have better performance as the algorithm A. Hence this approach is not suitable for LDPC codes with large girth. Due to these reasons, it has no choice but to search loops with only short length, and the performance improvement of this method is limited compared with the algorithm A. Moreover if more than one loop exists in some row, this method cannot eliminate all of these loops.

The performance of the previous approach for guessing erased bits as some values [4], [12] depends on a way of choosing these bits, guessed values, and the number of them. By guessing erased bits, these algorithms may have a miss-correction which has never been arised for our approach.

(2) Decoding Algorithm B

We use LDPC codes of its girth six. So the performance improvement obtained by the algorithm B arises from newly produced loops of length four by the algorithm A (See Observation 1). Although the decoding algorithms of the former works [7], [10], [11] are effective for LDPC codes with short length of loops, this algorithm is effective for those loops of arbitrary length.

4.4 Decoding Complexity of MLD

As described in Sect. 3.5, the algorithms AM and BM perform GE efficiently by using the condensed matrix H^{P} and H^{Q}, respectively. Hence the decoding complexity of GE strongly depends on the size of these matrices. We compare the average number of total binary operations of the decoding algorithm BPM (denoted by "algorithm BPM"), the decoding algorithm AM (denoted by "algorithm AM"), and the decoding algorithm BM (denoted by "algorithm BM"). As a reference, we also show the average number of total binary operations of performing only GE from the beginning (denoted by "OGE").

Figure 10 shows the average number of total binary operations of these decoding algorithms for the code C_1. Since the When p is relatively small, the average numbers of total binary operations of the algorithm AM and that of the algorithm BM are almost the same as that of the algorithm

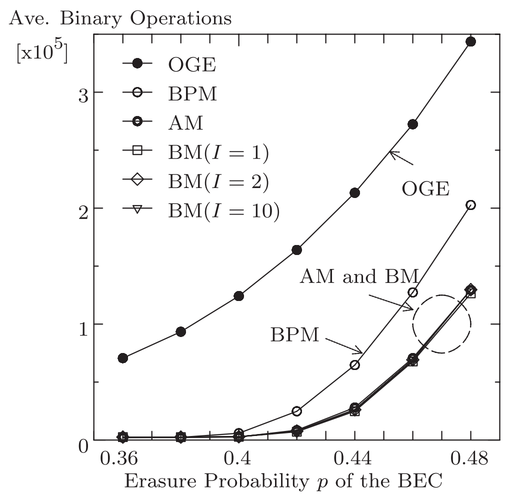


Fig. 10 The average number of total binary operations of each MLD algorithm for C_1.

Table 4 The ratio of the average number of binary operations for GE to that of all the procedures in the decoding for code C_1.

p	BPM	AM	BM		
			$I = 1$	$I = 2$	$I = 10$
0.36	0.930	0.819	0.704	0.709	0.718
0.38	0.940	0.845	0.745	0.752	0.759
0.4	0.952	0.870	0.783	0.786	0.788
0.42	0.968	0.913	0.844	0.844	0.844
0.44	0.980	0.947	0.899	0.897	0.895
0.46	0.989	0.982	0.952	0.950	0.950
0.48	0.994	0.985	0.968	0.911	0.849

Table 5 The average number of columns of each matrix for code C_1 after the algorithms BP, A, and B stops in failure. The original number of $H_{\mathcal{E}}$ is $pN = 1000p$.

p	$H_{\mathcal{E}_B}$	H^P	H^Q ($I = 1$)	H^Q ($I = 2$)	H^Q ($I = 10$)
0.36	212.4	82.5	85.0	88.1	92.9
0.38	223.6	91.3	92.0	97.2	101.4
0.4	237.2	99.0	100.7	104.1	108.2
0.42	262.3	117.3	116.7	120.6	123.9
0.44	290.3	138.2	137.8	139.9	142.8
0.46	335.4	176.6	173.7	175.3	176.1
0.48	383.4	229.8	227.4	227.2	227.2

BPM. On the contrary, as p increases, these numbers of the algorithm AM and the algorithm BM are lower than those of OGE and the algorithm BPM. Since most decoding trials are failed by the procedure B for large p, the number of performing each procedure is almost the same. To see the decoding complexity of GE in detail, Table 4 shows the ratio of the average number of binary operations for GE to that of all the procedures in decoding[†]. It can be verified that the number of binary operations of GE combined with all the algorithms dominates almost all the procedures in the decoding.

Table 5 shows the average numbers of columns of each matrix for C_1. From this table, the numbers of columns of H^{P} and H^{Q} are smaller than those of $H_{\mathcal{E}}$ and $H_{\mathcal{E}_B}$, and these numbers are almost the same for H^{P} and H^{Q}. Therefore it is obvious that the algorithms AM and BM can perform GE more efficiently than the algorithm BPM (and than OGE, naturally). Note that the average number of columns for H^{Q} when $I = 10$ are larger than those when $I = 1$, since the algorithm BPM only performs GE for the received sequence with large number of erasures which cannot be corrected by the decoding algorithm B with $I = 10$.

5. Conclusion and Further Work

We have developed new iterative decoding algorithms of LDPC codes over the BEC. Both algorithms perform substitution procedures of the matrix H and can eliminate the loops in a submatrix of H whose column positions are indexed by a stopping set. In Theorem 1, we have shown a correctable condition for an erased bit that cannot be corrected by the BP decoding algorithm. The derived condition is valid for the arbitrary length of loops. From simulation results, BERs of our decoding algorithms are lower than that of the BP decoding algorithm when the channel erasure probability takes small value.

Characterizing a relationship between decoding complexity and the value of p seems difficult but a challenging problem. Development of an analytical method of the performance of the proposed decoding algorithms is needed. For example, the residual degree distribution approach has been developed [2] to analytically track the behavior of row weight at each decoding step in the BP decoding algorithm. In Figs. 2 and 4, although we have shown the transition of weight of rows based on the simulation results. Evaluation of their performance by applying such approaches [2],[13, Theorem 3.106] to our algorithms is desired.

[†]For evaluating these numbers of the algorithm BPM, AM, and BM in Table 4, we averaged the number of binary operations over the received sequences which cannot be corrected by the algorithm BP, A, and B, respectively.

Acknowledgments

The authors wish to thank the reviewers for their thorough and conscientious reviewing for the paper. The authors would like to thank Prof. M. Goto at Waseda University and Prof. B. Kurkoski at the University of Electro-Communications for their valuable comments and discussions. This work is partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (Start-up), No. 20860074 and (B), No. 20760249.

References

[1] R.G. Gallager, "Low density parity check codes," IRE Trans. Inf. Theory, vol.8, pp.21–28, Jan. 1962.

[2] M. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, "Efficient erasure correcting codes," IEEE Trans. Inf. Theory, vol.47, no.2, pp.569–584, Feb. 2001.

[3] C. Di, D. Proietti, I.E. Telatar, T.J. Richardson, and R.L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," IEEE Trans. Inf. Theory, vol.48, no.6, pp.1570–1579, June 2002.

[4] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary-erasure channel," IEEE Trans. Inf. Theory, vol.50, no.3, pp.439–454, March 2004.

[5] D. Burshtein and G. Miller, "Asymptotic enumeration methods for analyzing LDPC codes," IEEE Trans. Inf. Theory, vol.50, no.6, pp.1115–1131, June 2004.

[6] T. Tian, C. Jones, J.D. Villasenor, and R.D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," IEEE Trans. Commun., vol.52, no.8, pp.1242–1247, Aug. 2004.

[7] K. Kasai, T. Shibuya, and K. Sakaniwa, "A code-equivalent transformation removing cycles of length four in Tanner graphs," IEICE Technical Report, IT2004-42, Sept. 2004.

[8] D. Burshtein and G. Miller, "An efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel," IEEE Trans. Inf. Theory, vol.50, no.11, pp.2837–2844, Nov. 2004.

[9] G. Hosoya, T. Matsushima, and S. Hirasawa, "A decoding method of low-density parity-check codes over the binary erasure channel," Proc. 27th Symposium on Information Theory and its Applications (SITA2004), pp.263–266, Gifu, Japan, Dec. 2004.

[10] S. Sankaranarayanan and B. Vasić, "Iterative decoding of linear block codes: A parity-check orthogonalization approach," IEEE Trans. Inf. Theory, vol.51, no.9, pp.3347–3353, Sept. 2005.
[11] N. Kobayashi, T. Matsushima, and S. Hirasawa, "Transformation of a parity-check matrix for a message-passing algorithm over the BEC," IEICE Trans. Fundamentals, vol.E89-A, no.5, pp.1299–1306, May 2006.
[12] B.N. Vellambi and F. Fekri, "Results on the improved decoding algorithm for low-density parity-check codes over the binary erasure channel," IEEE Trans. Inf. Theory, vol.53, no.4, pp.1510–1520, April 2007.
[13] T. Richardson and R. Urbanke, Modern coding theory, Cambridge University Press, 2008.

Gou Hosoya was born in Yokohama, Japan, on Dec. 14, 1979. He received the B.E. degree, M.E. degree, and D.E. degree in Industrial and Management Systems Engineering from Waseda University, Tokyo, Japan, in 2002, 2004, and 2008, respectively. From 2008, he has been a research associate in the Department of Industrial and Management Systems Engineering at Waseda University. His research interests are coding theory and information theory. He is a member of the IEEE, the Society of Information Theory and Its Applications.

Hideki Yagi was born in Yokohama, Japan, on Oct. 14, 1975. He received the B.E. degree, M.E. degree, and Dr.E. degree in Industrial and Management Systems Engineering from Waseda University, Tokyo, Japan, in 2001, 2003 and 2005, respectively. From 2005 to 2008, he was with Media Network Center, Waseda University as a research associate and an Assistant Professor. He is currently an assistant professor in Center for Frontier Science and Engineering at the University of Electro-Communications, Tokyo, Japan. His research interests are information theory and coding theory. He is a member of the Society of Information Theory and its Applications and IEEE.

Manabu Kobayashi was born in Yokohama, Japan, on Oct. 30, 1971. He received the B.E. degree, M.E. degree and Dr.E. degree in Industrial and Management Systems Engineering form Waseda University, Tokyo, Japan, in 1994, 1996 and 2000, respectively. From 1998 to 2001, he was a research associate in Industrial and Management Systems Engineering at Waseda University. He is currently an associate professor of the Department of Information Science at Shonan Institute of Technology, Kanagawa, Japan. His research interests are coding and information theory and data mining. He is a member of the Society of Information Theory and Its Applications, Information Processing Society of Japan and IEEE.

Shigeichi Hirasawa was born in Kobe, Japan, on Oct. 2, 1938. He received the B.S. degree in mathematics and the B.E. degree in electrical communication engineering from Waseda University, Tokyo, Japan, in 1961 and 1963, respectively, and the Dr.E. degree in electrical communication engineering from Osaka University, Osaka, Japan, in 1975. From 1963 to 1981, he was with the Mitsubishi Electric Corporation, Hyogo, Japan. From 1981 to 2009, he was a professor of the School of Science and Engineering, Waseda University, Tokyo, Japan. He is currently a research consultant at the Waseda Research Institute for Science and Engineering, Waseda University, and a consultant at Cyber University. In 1979, he was a Visiting Scholar in the Computer Science Department at the University of California, Los Angeles (CSD, UCLA), CA. He was a Visiting Researcher at the Hungarian Academy of Science, Hungary, in 1985, and at the University of Trieste, Italy, in 1986. In 2002, he was also a Visiting Faculty at CSD, UCLA. From 1987 to 1989, he was the Chairman of the Technical Group on Information Theory of IEICE. He received the 1993 Achievement Award and the 1993 Kobayashi-Memorial Achievement Award from IEICE. In 1996, he was the President of the Society of Information Theory and Its Applications (Soc. of ITA). His research interests are information theory and its applications, and information processing systems. He is an IEEE Life Fellow, and a member of Soc. of ITA, and the Information Processing Society of Japan.