PAPER

Message Passing Decoder with Decoding on Zigzag Cycles for Non-binary LDPC Codes*

Takayuki NOZAKI[†a)], Kenta KASAI[††b)], *Members*, *and* Kohichi SAKANIWA[††c)], *Fellow*

SUMMARY In this paper, we propose a message passing decoding algorithm which lowers decoding error rates in the error floor regions for non-binary low-density parity-check (LDPC) codes transmitted over the binary erasure channel (BEC) and the memoryless binary-input output-symmetric (MBIOS) channels. In the case for the BEC, this decoding algorithm is a combination with belief propagation (BP) decoding and maximum a posteriori (MAP) decoding on zigzag cycles, which cause decoding errors in the error floor region. We show that MAP decoding on the zigzag cycles is realized by means of a message passing algorithm. Moreover, we extend this decoding algorithm to the MBIOS channels. Simulation results demonstrate that the decoding error rates in the error floor regions by the proposed decoding algorithm are lower than those by the BP decoder.

key words: *non-binary LDPC code, error floor, memoryless binary-input output-symmetric, belief propagation decoder, maximum a posteriori decoding*

1. Introduction

Gallager invented low-density parity-check (LDPC) codes [1]. Due to the sparseness of the parity check matrices, LDPC codes are efficiently decoded by the belief propagation (BP) decoder. Optimized LDPC codes can exhibit performance very close to the Shannon limit [2]. Davey and MacKay [3] have found that non-binary LDPC codes can outperform binary ones. In this paper, we assume the non-binary LDPC codes defined over $\mathbb{F}_{2^m}$ transmitted over the memoryless binary-input output-symmetric (MBIOS) channels, where $\mathbb{F}_{2^m}$ is the finite field of order 2^m.

A non-binary LDPC code over $\mathbb{F}_{2^m}$ is defined by the null space of a sparse parity-check matrix over $\mathbb{F}_{2^m}$. A Tanner graph for a non-binary LDPC code is represented by a bipartite graph with variable nodes, check nodes and labeled edges. Each LDPC code is represented by Tanner graphs. It is known that Tanner graphs of optimized irregular non-binary LDPC codes contain variable nodes of degree two [4].

The curve of the decoding error rate for a finite code length LDPC code is divided into two regions called *waterfall region* and *error floor region*, or simply waterfall and error floor. In the waterfall, the decoding error rate drops off steeply as the function of channel error probability. The waterfall is mainly caused by the large weight errors. In the error floor, the decoding error rate has a gentle slope. The error floor is mainly caused by the small weight errors. In this paper, we investigate the decoding error rates in the error floors for non-binary LDPC codes.

A *zigzag cycle* is a *simple cycle*, or a *circuit* such that variable nodes are of degree two in the Tanner graph. Since the zigzag cycles cause small weight errors, the decoding error rates in the error floors are mainly caused by zigzag cycles. Hence, in this paper, we focus on the decoding errors in the zigzag cycles to lower the decoding error rates in the error floors.

There are two approaches to lowering the decoding error rates in the error floors: by optimizing the code and by improving the decoder. By optimizing labels in the zigzag cycles in non-binary LDPC codes, the decoding error rates in the error floors were lowered by [5], [6]. On the other hand, there are no message passing (MP) decoders lowering the decoding error rates in the error floors.

In this paper, we propose an MP decoding algorithm reducing the decoding errors in the zigzag cycles. Firstly, we investigate the maximum a posteriori (MAP) decoding algorithm for the non-binary LDPC codes over the binary erasure channels (BECs). From this result, we show that MAP decoding on the zigzag cycles is realized by means of an MP algorithm for the BEC. Secondly, we propose an MP decoding algorithm by combining with BP decoding and decoding on the zigzag cycles for the non-binary LDPC codes over the BECs. Thirdly, we extend the proposed MP decoding algorithm to the MBIOS channels. Finally, we show by simulation results that the decoding error rates in the error floors by the proposed decoding algorithm are lower than those by the BP decoder.

The remainder of this paper is organized as follows: Sect. 2 defines non-binary LDPC codes and introduces BP and MAP decoding algorithms for the non-binary LDPC codes. Section 3 reviews a MAP decoding algorithm on the zigzag cycles over the BEC. Section 4 proposes a decoding algorithm which reduces the decoding erasure rates in the error floors for non-binary LDPC codes over the BEC. Section 5 extends the algorithm proposed in Sect. 4 to the MBIOS channels. Simulation results in Sect. 6 demonstrate that the proposed decoding algorithm lowers the decoding error rates in the error floors.

Manuscript received November 19, 2013.

[†] The author is with the Dept. of Information Systems Creation, Kanagawa University, Yokohama-shi, 221-8686 Japan.

[††] The authors are with the Dept. of Communications and Computer Engineering, Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

[*] The material in this paper was presented in part at IEEE International Symposium on Information Theory (ISIT2013).

a) E-mail: nozaki@kanagawa-u.ac.jp
b) E-mail: kenta@comm.ce.titech.ac.jp
c) E-mail: sakaniwa@comm.ce.titech.ac.jp

DOI: 10.1587/transfun.E97.A.975

2. Preliminaries

This section defines non-binary LDPC codes and introduces channel models. Moreover, we recall BP and MAP decoding algorithms for the non-binary LDPC codes over the MBIOS channel and the BEC, respectively. This section introduces some notations throughout used in this paper.

2.1 Non-binary LDPC Codes

A non-binary LDPC code over $\mathbb{F}_{2^m}$ is defined by the null space of a sparse parity-check matrix $H = (h_{i,j}) \in \mathbb{F}_{2^m}^{M \times N}$:

$$\left\{ \boldsymbol{x} \in \mathbb{F}_{2^m}^N \mid H\boldsymbol{x}^T = \boldsymbol{0}^T \in \mathbb{F}_{2^m}^M \right\}.$$

Note that N is called *symbol code length.*

In this paper, we consider a non-binary LDPC code represented by a Tanner graph $(\mathsf{V} \cup \mathsf{C}, \mathsf{E})$, where $\mathsf{V} = \{v_1, v_2, \dots, v_N\}$, $\mathsf{C} = \{c_1, c_2, \dots, c_M\}$ and E are the sets of the variable nodes, the check nodes and the labeled edges, respectively. The v-th variable node and the c-th check node are connecting to an edge labeled as $h_{c,v} \in \mathbb{F}_{2^m} \setminus \{0\}$ if $h_{c,v} \neq 0$.

The LDPC codes defined by Tanner graphs with the variable nodes of degree d_v and the check nodes of degree d_c are called $(d_\mathrm{v}, d_\mathrm{c})$-regular LDPC codes. Each irregular LDPC code is characterized by a pair of degree distributions $\Lambda(x) = \sum_i \Lambda_i x^i$ and $P(x) = \sum_i P_i x^i$, where Λ_i (resp. P_i) represents the number of the variable nodes (resp. check nodes) of degree i. Notice that $N = \sum_i \Lambda_i$ and $M = \sum_i P_i$. It is known that optimized irregular non-binary LDPC codes contain variable nodes of degree two [4]. In particular, $(2,d_c)$-regular non-binary LDPC codes exhibit good decoding performance among other LDPC codes for $2^m \geq 64$.

2.2 Channel Model

Let α be a primitive element of $\mathbb{F}_{2^m}$. Once a primitive element α of $\mathbb{F}_{2^m}$ is fixed, each symbol is given by an m-bit representation [7, p.110]. We denote the m-bit representation of $\gamma \in \mathbb{F}_{2^m}$, by $b(\gamma) = (b_1(\gamma), b_2(\gamma), \dots, b_m(\gamma))$.

For integers a, b, we denote the set of integers between a and b, as $[a; b]$. Note that $[a; b] = \emptyset$ if $a > b$. By using the m-bit representation, we regard each codeword $\boldsymbol{x} = (x_1, x_2, \dots, x_N)$ in a non-binary LDPC code as a binary codeword, i.e., a codeword is represented by $(x_{1,1}, x_{1,2}, \dots, x_{N,m})$, where $x_{i,j} = b_j(x_i)$ for $i \in [1; N]$ and $j \in [1; m]$. Hence, the codewords in the non-binary LDPC codes defined over $\mathbb{F}_{2^m}$ are able to be transmitted over binary channels. We denote the received word as $\boldsymbol{y} = (y_{1,1}, y_{1,2}, \dots, y_{N,m})$.

A binary input channel is called *memoryless* if

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \textstyle\prod_{i=1}^N \prod_{j=1}^m p(y_{i,j} \mid x_{i,j}).$$

We transform the binary input alphabets $\{0, 1\}$ into $\{+1, -1\}$. With some abuse of notation, we make no distinction between $\{0, 1\}$ and $\{+1, -1\}$. A memoryless binary-input channel is called *output-symmetric* if

$$p(y \mid x) = p(-y \mid -x).$$

We assume the transmission over the MBIOS channel. Examples of the MBIOS channels include the BEC and the binary additive white Gaussian noise (BAWGN) channel.

2.3 BP Decoding Algorithm for MBIOS Channel

In the BP decoding algorithm, each edge in a Tanner graph transfers a message. The messages arising in the BP decoder for non-binary LDPC codes over $\mathbb{F}_{2^m}$ are vectors of length 2^m. Let $\Psi_{v,c}^{(\ell)}$ (resp. $\Phi_{c,v}^{(\ell)}$) be the message from the v-th variable node (resp. c-th check node) to the c-th check node (resp. v-th variable node) at the ℓ-th iteration.

(1) Initialization

Set $\ell = 0$. Recall that N and M represents the number of variable nodes and check nodes in the Tanner graph, respectively. For $v \in [1; N]$, let $C_v = (C_v(0), C_v(1), \dots, C_v(\alpha^{2^m-2}))$ denote the initial message of the v-th variable node. For $\gamma \in \mathbb{F}_{2^m}$, the component $C_v(\gamma)$ of the initial message is derived from the channel outputs as follows:

$$C_v(\gamma) = \textstyle\prod_{i=1}^m \Pr(Y_{v,i} = y_{v,i} \mid X_{v,i} = b_i(\gamma)).$$

Let $\mathcal{N}_\mathrm{v}(i)$ (resp. $\mathcal{N}_\mathrm{c}(j)$) be the set of indices of the check nodes (resp. variable nodes) adjacent to the i-th variable node (resp. the j-th check node). Set for all $c \in [1; M]$ and $v \in \mathcal{N}_\mathrm{c}(c)$,

$$\Phi_{c,v}^{(0)} = (2^{-m}, 2^{-m}, \dots, 2^{-m}).$$

(2) Iteration and Decision

The convolution of two vectors Ψ_1 and Ψ_2 is defined as

$$[\Psi_1 \oplus \Psi_2](x) := \textstyle\sum_{y \in \mathbb{F}_{2^m}} \Psi_1(y) \Psi_2(x - y).$$

To simplify the notation, we define $\bigoplus_{i \in [1;k]} \Psi_i := \Psi_1 \oplus \Psi_2 \oplus \cdots \oplus \Psi_k$. Define

$$\mathrm{argmax}_{x \in \mathbb{F}_{2^m}} \Psi(x) := \{x \in \mathbb{F}_{2^m} \mid \Psi(x) \geq \Psi(y)\ \forall y \in \mathbb{F}_{2^m}\}.$$

The BP decoding algorithm is described as follows:

1. In the v-th variable node, the message $\Psi_{v,c}^{(\ell)}$ is given by

$$\Psi_{v,c}^{(\ell)}(x) = \xi^{-1} C_v(x) \textstyle\prod_{c' \in \mathcal{N}_\mathrm{v}(v) \setminus \{c\}} \Phi_{c',v}^{(\ell)}(x),$$

for $x \in \mathbb{F}_{2^m}$, where ξ is a normalization factor such that $1 = \sum_{x \in \mathbb{F}_{2^m}} \Psi_{v,c}^{(\ell)}(x)$.

2. In the c-th check node, the message $\Phi_{c,v}^{(\ell+1)}$ is calculated as, for $x \in \mathbb{F}_{2^m}$

$$\check{\Psi}_{v,c}^{(\ell)}(x) = \Psi_{v,c}^{(\ell)}(h_{c,v}^{-1} x), \quad \check{\Phi}_{c,v}^{(\ell+1)} = \textstyle\bigoplus_{v' \in \mathcal{N}_\mathrm{c}(c) \setminus \{v\}} \check{\Psi}_{v',c}^{(\ell)},$$
$$\Phi_{c,v}^{(\ell+1)}(x) = \check{\Phi}_{c,v}^{(\ell+1)}(h_{c,v} x).$$

3. In the v-th variable node, the decoding result $D_v^{(\ell)} \subseteq \mathbb{F}_{2^m}$ is calculated from

$$D_v^{(\ell)} = \mathrm{argmax}_{x \in \mathbb{F}_{2^m}} C_v(x) \textstyle\prod_{c \in \mathcal{N}_\mathrm{v}(v)} \Phi_{c,v}^{(\ell)}(x).$$

If ℓ reaches the maximum number of iterations $L_{\max}$ or the decoding results $(D_i^{(\ell)})_{i=1}^N$ satisfy the parity check constraints, the decoding result D_v is set as $D_v \leftarrow D_v^{(\ell)}$ for all $v \in [1; N]$ and the algorithm terminates. Otherwise, set $\ell \leftarrow \ell + 1$ and go to Step 1.

Let $\hat{x}_v^{\mathrm{BP}}$ be the v-th output of BP decoder. If $|D_v| = 1$, $\hat{x}_v^{\mathrm{BP}}$ is set as the unique element of D_v. Otherwise, set $\hat{x}_v^{\mathrm{BP}} = ?$. If $\hat{x}_v^{\mathrm{BP}} = x_v$, the v-th symbol is correctly decoded, where x_v is the v-th transmitted symbol. Otherwise, the v-th variable node has a decoding error.

2.4 BP Decoding Result for BEC

In this section, we review the BP decoding algorithm for the non-binary LDPC codes over the BECs. In the case for the BEC, the messages arising in the BP decoder satisfy a property such that all the non-zero entries in a message are equal [8, Lemma 2]. Hence, the v-th symbol is correctly decoded if and only if $|D_v| = 1$.

We denote the set of the indices of the variable nodes with decoding erasures, by $\mathcal{D}$, namely, $\mathcal{D} := \{i \in [1; N] \mid |D_i| > 1\}$. Define $\bar{\mathcal{D}} := [1; N] \setminus \mathcal{D}$. In words, $\bar{\mathcal{D}}$ represents the set of indices of the variable nodes which are correctly decoded. For $j \in [1; M]$, we define the *syndrome* s_j for the j-th check node as

$$s_j := \textstyle\sum_{i \in \mathcal{N}_{\mathrm{c}}(j) \cap \bar{\mathcal{D}}} h_{j,i} \hat{x}_i^{\mathrm{BP}} \in \mathbb{F}_{2^m}.$$

Notice that the empty sum $\sum_{i \in \emptyset} a_i$ is equal to zero for any $a_1, \ldots, a_k \in \mathbb{F}$.

We refer to the subgraph constructed by the variable nodes with decoding erasures as the *residual graph* [9] after BP decoding. More precisely, a residual graph $(\mathsf{V}_{\mathcal{D}} \cup \mathsf{C}_{\mathcal{D}}, \mathsf{E}_{\mathcal{D}})$ is a subgraph of the Tanner graph such that $\mathsf{V}_{\mathcal{D}} = \{v_i \mid |D_i| \neq 1\}$, $\mathsf{E}_{\mathcal{D}}$ is the set of all the edges connecting to $\mathsf{V}_{\mathcal{D}}$ and $\mathsf{C}_{\mathcal{D}} = \{c_j \mid \exists v_i \in \mathsf{V}_{\mathcal{D}} \text{ s.t. } j \in \mathcal{N}_{\mathrm{v}}(i)\}$.

A stopping set $\mathcal{S}$ is a set of variable nodes such that all the neighbors of $\mathcal{S}$ are adjacent to $\mathcal{S}$ at least twice. It is known that the set of variable nodes with decoding erasures forms a stopping set.

2.5 MAP Decoding Algorithm for BEC

In this paper, we consider the *symbol-wise* MAP decoding using the decoding rule

$$D_i^{\mathrm{MAP}}(\boldsymbol{y}) = \operatorname{argmax}_{\gamma \in \mathbb{F}_{2^m}} p_{X_i|\boldsymbol{Y}}(\gamma \mid \boldsymbol{y}),$$

for a given received word $\boldsymbol{y} \in \boldsymbol{Y} = Y^{mN}$. In the case for the BEC, the received word $\boldsymbol{y}$ fulfills $y_{i,j} = x_{i,j}$ if the bit $y_{i,j}$ is received correctly. We assume that codewords are chosen uniformly. Define

$$\begin{aligned}\mathcal{X}_{\mathrm{MAP}}(\boldsymbol{y}) := \{&\boldsymbol{x} \in C \mid H\boldsymbol{x}^T = \boldsymbol{0}^T, x_{i,j} = y_{i,j} \\ &\text{for } (i,j) \in [1; N] \times [1; m] \text{ s.t. } y_{i,j} \in \{0,1\}\}.\end{aligned}$$

The symbol-wise MAP decoding for the i-th symbol is given by the set of the i-th symbol in $\mathcal{X}_{\mathrm{MAP}}(\boldsymbol{y})$, i.e.,

$$D_i^{\mathrm{MAP}}(\boldsymbol{y}) = \{x_i \in \mathbb{F}_{2^m} \mid \boldsymbol{x} \in \mathcal{X}_{\mathrm{MAP}}(\boldsymbol{y})\}.$$

The i-th symbol is correctly decoded if and only if $|D_i^{\mathrm{MAP}}(\boldsymbol{y})| = 1$. Let $\hat{x}_i^{\mathrm{MAP}}$ be the i-th output of the MAP decoding algorithm. If $|D_i^{\mathrm{MAP}}(\boldsymbol{y})| = 1$, $\hat{x}_i^{\mathrm{MAP}}$ is set the unique element of $D_i^{\mathrm{MAP}}(\boldsymbol{y})$. Otherwise, set $\hat{x}_i^{\mathrm{MAP}} = ?$.

It is well known that if symbols are decoded under BP decoding, the symbols also are decoded under MAP decoding. Hence, $\mathcal{X}_{\mathrm{MAP}}$ is rewritten as

$$\begin{aligned}\mathcal{X}_{\mathrm{MAP}} &= \{\boldsymbol{x} \in C \mid H\boldsymbol{x}^T = \boldsymbol{0}^T, x_i \in D_i \text{ for } i \in [1; N]\} \\ &= \{\boldsymbol{x} \in C \mid H_{\mathcal{D}}\boldsymbol{x}_{\mathcal{D}}^T = \boldsymbol{s}^T, x_i \in D_i \text{ for } i \in [1; N]\},\end{aligned}$$

where $H_{\mathcal{D}}$ denotes the submatrix of H, restricted to the columns indexed by $\mathcal{D}$, $x_{\mathcal{D}}$ denotes the corresponding subvector of $\boldsymbol{x}$ and $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$. Thus, the i-th symbol is correctly decoded under symbol-wise MAP decoding *if* $H_{\mathcal{D}}\boldsymbol{x}_{\mathcal{D}}^T = \boldsymbol{s}^T$ has a unique solution.

3. MAP Decoding on Zigzag Cycles over BEC

In this section, we define the zigzag cycles in the non-binary LDPC codes and we show MAP decoding results for the zigzag cycles over the BEC.

3.1 Zigzag Cycle

Definition 1: A zigzag cycle is a simple cycle or a circuit such that all the variable nodes in the zigzag cycle are of degree two. More precisely, a zigzag cycle $(\mathsf{V}_{\mathrm{zc}} \cup \mathsf{C}_{\mathrm{zc}}, \mathsf{E}_{\mathrm{zc}})$ is a *connected* subgraph [10] of the Tanner graph such that all the variable nodes in V_{zc} are of degree two, E_{zc} is the set of all the edges connecting to the variable nodes in V_{zc} and the check nodes in C_{zc} connect to the edges in E_{zc} at *exactly* twice.

Since the zigzag cycles cause small weight errors, the decoding error rates in the error floors are mainly caused by zigzag cycles. All the zigzag cycles cause decoding erasures under BP decoding for the BEC if all the bits in the zigzag cycles are received erasures, i.e., $\forall i \in \mathsf{V}_{\mathrm{zc}} \forall j \in [1; m]\ y_{i,j} = ?$ [11, Appendix A.2]. However, most of the zigzag cycles are decodable under MAP decoding even if all the bits in the zigzag cycles are received erasures. In the next section, we consider the zigzag cycles under MAP decoding.

3.2 MAP Decoding on Zigzag Cycles

In this section, we assume that the residual graph after BP decoding forms a zigzag cycle. To simplify the notation, we consider the zigzag cycle in the residual graph as in Fig. 1. In words, we consider a zigzag cycle of weight w labeled with $h_{1,1}, h_{1,2}, h_{2,2}, \ldots, h_{w,w}, h_{w,1}$.

The submatrix H_{zc} corresponding to the zigzag cycle is

$$H_{\mathrm{zc}} = \begin{pmatrix} h_{1,1} & h_{1,2} & 0 & \cdots & 0 \\ 0 & h_{2,2} & h_{2,3} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{w,1} & 0 & 0 & \cdots & h_{w,w} \end{pmatrix}.$$

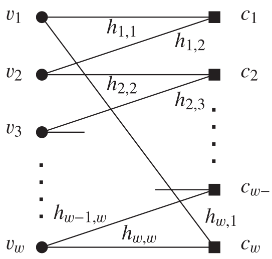


Fig. 1 A zigzag cycle of weight w.

To simplify the notation, we define for $i \in [1; w]$, $n \in \mathbb{Z}$

$$h_{nw+i,nw+i} := h_{i,i}, \quad h_{nw+i,nw+i+1} := h_{i,i+1},$$
$$\zeta_i := h_{i-1,i} h_{i,i}^{-1}, \quad \beta := \textstyle\prod_{i=1}^{w} \zeta_i. \tag{1}$$

Note that an *empty product* is equal to 1, i.e., $\prod_{i\in\emptyset} a_i = 1$ for any $a_1, a_2, \dots, a_k \in \mathbb{F}$. To simplify the notation, we denote $\hat{x}_i^{\mathrm{MAP}}$ as $\hat{x}_i$ in this section. Recall that s_i represents the syndrome corresponding to the i-th check node. For a given $s_1, s_2, \dots, s_w$, the MAP decoding result $(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_w)$ fulfills the following equation

$$(s_1, s_2, \dots, s_w)^T = H_{\mathrm{zc}} (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_w)^T .$$

If the matrix H_{zc} is non-singular, i.e., $\beta \neq 1$, then the (i, j)-th entry of H_{zc}^{-1} is given as

$$(H_{\mathrm{zc}}^{-1})_{i,j} = \begin{cases} (1+\beta)^{-1} h_{i,i}^{-1} \prod_{k\in[i+1;j]} \zeta_k, & i \leq j, \\ (1+\beta)^{-1} h_{i,i}^{-1} \prod_{k=i+1}^{w+j} \zeta_k, & i > j. \end{cases} \tag{2}$$

From Eq. (2), the MAP decoding result $\hat{x}_i$ is written as

$$\hat{x}_i = \textstyle\sum_{j=1}^{w} (H_{\mathrm{zc}}^{-1})_{i,j} s_j = A_{w,i}(1+\beta)^{-1}, \tag{3}$$

for $i \in [1; w]$, where

$$A_{w,i} := h_{i,i}^{-1} \textstyle\sum_{j=0}^{w-1} s_{i+j} \prod_{k\in[1;j]} \zeta_{i+k}.$$

In the case for $A_{w,i} = 0$ and $\beta \neq 1$, we obtain from (3)

$$\hat{x}_i = 0.$$

Define

$$B_{w,i} := \beta^{-1} A_{w,i} = h_{i-1,i}^{-1} \textstyle\sum_{j=0}^{w-1} s_{i+j} \prod_{k\in[j+1;w-1]} \zeta_{i+k}^{-1}.$$

Then, the submatrix H_{zc} is non-singular if and only if

$$\beta = A_{w,i} B_{w,i}^{-1} \neq 1 \iff A_{w,i} \neq B_{w,i}$$

for $A_{w,i}, B_{w,i} \neq 0$. For $A_{w,i}, B_{w,i} \neq 0$, the substitution $\beta = A_{w,i} B_{w,i}^{-1}$ into Eq. (3) yields

$$\hat{x}_i = A_{w,i} B_{w,i} (A_{w,i} + B_{w,i})^{-1}.$$

Hence, if $A_{w,i}$ and $B_{w,i}$ are derived by an MP algorithm, we obtain the MAP decoding results on the zigzag cycles.

Now, we consider two recurrence relations

$$a_{w,i}^{(\ell)} = h_{i-\ell,i-\ell}^{-1} \Big(h_{i-\ell,i-\ell+1} a_{w,i}^{(\ell-1)} + s_{i-\ell} \Big), \tag{4}$$

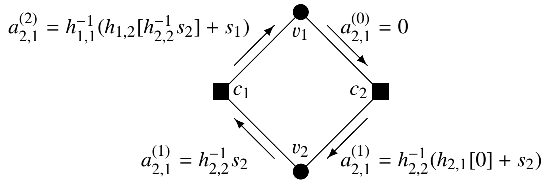

$a_{2,1}^{(2)} = h_{1,1}^{-1}(h_{1,2}[h_{2,2}^{-1}s_2] + s_1)$
v_1
$a_{2,1}^{(0)} = 0$
c_1
c_2
$a_{2,1}^{(1)} = h_{2,2}^{-1}s_2$
v_2
$a_{2,1}^{(1)} = h_{2,2}^{-1}(h_{2,1}[0] + s_2)$

Fig. 2 Calculation of $A_{2,1}$.

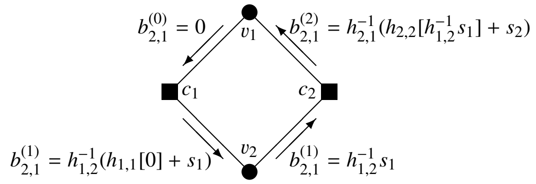

$b_{2,1}^{(0)} = 0$
v_1
$b_{2,1}^{(2)} = h_{2,1}^{-1}(h_{2,2}[h_{1,2}^{-1}s_1] + s_2)$
c_1
c_2
$b_{2,1}^{(1)} = h_{1,2}^{-1}(h_{1,1}[0] + s_1)$
v_2
$b_{2,1}^{(1)} = h_{1,2}^{-1}s_1$

Fig. 3 Calculation of $B_{2,1}$.

$$b_{w,i}^{(\ell)} = h_{i+\ell-1,i+\ell}^{-1} \Big(h_{i+\ell-1,i+\ell-1} b_{w,i}^{(\ell-1)} + s_{i+\ell-1} \Big), \tag{5}$$

with the initial terms $a_{w,i}^{(0)} = b_{w,i}^{(0)} = 0$. Then, we get $A_{w,i} = a_{w,i}^{(w)}$ and $B_{w,i} = b_{w,i}^{(w)}$. Thus, we are able to obtain the MAP decoding result $\hat{x}_i$ by solving Eqs. (4), (5).

The calculations of the recurrence Eqs. (4) and (5) are carried out in the check nodes as the following way:

1. Multiply the incoming message by the label of the edge which transfers the incoming message to the check node.
2. Add the syndrome of the check node and the computation result of Step 1.
3. Multiply the computed result of Step 2 by the inverse of the label of the edge which transfers the outgoing message from the check node.

Section 4.2 realizes this calculation by an MP algorithm for any zigzag cycles with decoding erasures.

Example 1: Consider a zigzag cycle of weight 2 with labels $h_{1,1}, h_{2,1}, h_{2,2}, h_{1,2}$. Figures 2 and 3 illustrate calculation of $A_{2,1}$ and $B_{2,1}$ in the Tanner graph, respectively. The second check node c_2 in Fig. 2 carries out the following calculations:

1. Multiply the incoming message $a_{2,1}^{(0)}$ by the label $h_{2,1}$ in the edge conveying the incoming message.
2. Add the calculation result $h_{2,1}a_{2,1}^{(0)}$ and the syndrome s_2 of the second check node.
3. Multiply the calculation result $h_{2,1}a_{2,1}^{(0)} + s_2$ by the inverse of label $h_{2,2}^{-1}$ transferring the outgoing message.

Similarly, the first check node in Fig. 2 outputs $a_{2,1}^{(2)} = A_{2,1}$ from the incoming message $a_{2,1}^{(1)}$. Finally, v_1 gets the message $A_{2,1}$. Conversely, the message flow in Fig. 3 computes $B_{2,1}$. By using $A_{2,1}$ and $B_{2,1}$, the first variable node v_1 obtains the MAP decoding result $\hat{x}_1^{\mathrm{MAP}}$.

4. MP Decoder with Decoding on Zigzag Cycles for BEC

In this section, we propose an MP decoding algorithm which reduces the decoding erasure rates in the error floors for non-binary LDPC codes over the BEC. If there are variable nodes with erasures after BP decoding, the proposed decoding algorithm decodes the erasures in the zigzag cycles by MAP decoding algorithm. Hence, the proposed decoder needs to search the zigzag cycles with BP decoding erasures before MAP decoding on the zigzag cycles. Thus, the proposed decoding algorithm is divided into 3 steps: (i) BP decoding, (ii) zigzag cycle detection and (iii) MAP decoding on zigzag cycles. All the steps are realized by means of an MP algorithm. Steps (i), (ii) and (iii) are given in Sects. 2.3, 4.1 and 4.2, respectively.

4.1 Zigzag Cycle Detection

If there are variable nodes with erasures after BP decoding, the proposed decoder searches the zigzag cycles with BP decoding erasures. In other words, the proposed decoder removes the stopping sets except the zigzag cycles from the residual graph after BP decoding.

A residual graph after BP decoding is divided into some connected subgraphs. From Definition 1, a connected graph in a residual graph is not zigzag cycle if the connected subgraph contains the node of degree more than two in the residual graph. Hence, if we remove such connected subgraphs, we are able to detect the zigzag cycles with decoding erasures.

The detection of zigzag cycles is detailed in Algorithm 1. In this algorithm, there are three messages "0", "1" and "−1". The variable nodes sending the message "0" are successfully decoded under BP decoding. The variable nodes sending the message "−1" are not in zigzag cycles with BP decoding erasures. If variable nodes can be in zigzag cycles with BP decoding erasures, the variable nodes send the message "1".

Steps 2–11 in Algorithm 1 carry out the initialization of the variable nodes. If a variable node with BP decoding erasure is of degree more than two, the connected subgraph which contains the variable node is not zigzag cycle. Hence such variable node sends the message "−1". If a variable node with BP decoding erasure is of degree two, the connected subgraph which contains the variable node can be a zigzag cycle. Hence such variable node sends the message "1".

Steps 12–16 in Algorithm 1 carry out the initialization of the check nodes. If a check node receiving the message "−1" or receiving more than two messages "1", the check node sends the message "−1" since the connected subgraph which contains the check node is not zigzag cycle.

Steps 17–29 are the iteration step of Algorithm 1. If a node receives the message "−1", the node sends the message "−1" since the connected subgraph which contains the check node is not zigzag cycle. The set $\mathcal{A}^{(\ell)}$ represents the set of indices of the variable nodes which can be in the zigzag cycles with BP decoding erasures. If $\mathcal{A}^{(\ell)} = \mathcal{A}^{(\ell-1)}$, $\mathcal{A}^{(\ell)}$ gives the set of indices of variable nodes in zigzag cycles with BP decoding erasures. Hence, the algorithm sets $\mathcal{A} \leftarrow \mathcal{A}^{(\ell)}$ and outputs $\mathcal{A}$.

Algorithm 1 Zigzag cycle detection

```
Input: The set $\mathcal{D}$ of indices of the variable node with BP decoding erasures
Output: The set $\mathcal{A}$ of indices of the variable nodes in the zigzag cycles with BP decoding erasures
 1: $\ell \leftarrow 0, \mathcal{A}^{(0)} \leftarrow \mathcal{D}$
 2: for $i \in \mathcal{D}$ do
 3:     if the degree of the $i$-th variable node is more than two then
 4:         the $i$-th variable node sends "$-1$", $\mathcal{A}^{(0)} \leftarrow \mathcal{A}^{(0)} \setminus \{i\}$
 5:     else
 6:         the $i$-th variable node sends "1"
 7:     end if
 8: end for
 9: for $i \in [1;N] \setminus \mathcal{D}$ do
10:     the $i$-th variable node sends "0"
11: end for
12: for $j \in [1;M]$ do
13:     if the $j$-th check node receives "$-1$" or more than two "1" then
14:         the $j$-th check node sends "$-1$"
15:     end if
16: end for
17: repeat
18:     $\ell \leftarrow \ell + 1, \mathcal{A}^{(\ell)} \leftarrow \mathcal{A}^{(\ell-1)}$
19:     for $i \in \mathcal{A}^{(\ell)}$ do
20:         if the $i$-th variable node receives "$-1$" then
21:             the $i$-th variable node sends "$-1$", $\mathcal{A}^{(\ell)} \leftarrow \mathcal{A}^{(\ell)} \setminus \{i\}$
22:         end if
23:     end for
24:     for $j \in [1;M]$ do
25:         if the $j$-th check node receives "$-1$" then
26:             the $j$-th check node sends "$-1$"
27:         end if
28:     end for
29: until $\mathcal{A}^{(\ell)} = \mathcal{A}^{(\ell-1)}$
30: $\mathcal{A} \leftarrow \mathcal{A}^{(\ell)}$
```

4.2 MP Decoding Algorithm on Zigzag Cycles

In the previous section, we get the zigzag cycles in the residual graph. In this section, we realize MAP decoding on zigzag cycles by an MP algorithm for the BEC. In other words, we propose an MP algorithm which gives $A_{w,i}$ and $B_{w,i}$ by using Eqs. (4), (5). In this section, we assume that the codes contain no singular submatrices corresponding to zigzag cycles of weight in $[1; w_{\mathrm{c}}]$. In other words, we employ the codes designed by cycle cancellation [5] or improved cycle cancellation [6] since those codes exhibit good decoding performance in error floors and the proposed algorithm works well.

Let $(\psi_{c,v}^{(\ell)}, p_{c,v}^{(\ell)}) \in \mathbb{F}_{2^m} \times [1;N]$ (resp. $(\phi_{c,v}^{(\ell)}, q_{c,v}^{(\ell)}) \in \mathbb{F}_{2^m} \times [1;N]$) be the message from the v-th variable node (resp. c-th check node) to the c-th check node (resp. v-th variable node) at the ℓ-th iteration. Denote the set of indices of the variable nodes in the zigzag cycles with BP decoding erasures, by $\mathcal{A}$. The decoding algorithm on the zigzag cycles is in Algorithm 2.

Steps 1–3 in Algorithm 2 perform the initialization of this algorithm. In those steps, all the variable nodes in the zigzag cycles send the initial messages. The first component $\phi_{j,i}^{(0)}$ of the initial message represents the initial terms

Algorithm 2 Decoding algorithm on zigzag cycles

```
Input: The set $\mathcal{A}$ of indices of the variable nodes in the zigzag cycle with
    BP decoding erasures and syndrome $\boldsymbol{s}$
Output: The decoding result $\{\hat{x}_i\}_{i=1}^{N}$
 1: for $i \in \mathcal{A}$ do
 2:     $(\psi^{(0)}_{j,i}, p^{(0)}_{j,i}) \leftarrow (0, i)\ \forall j \in \mathcal{N}_{\mathrm{v}}(i)$
 3: end for
 4: $\ell \leftarrow 1$
 5: while $\mathcal{A} \neq \emptyset$ do
 6:     for $j \in \mathcal{N}_{\mathrm{v}}(\mathcal{A})$ do
 7:         for $i \in \mathcal{N}_{\mathrm{c}}(j) \cap \mathcal{A}$ do
 8:             Set $i'$ as the unique element of $(\mathcal{N}_{\mathrm{c}}(j) \cap \mathcal{A}) \setminus \{i\}$
 9:             $(\phi^{(\ell)}_{j,i}, q^{(\ell)}_{j,i}) \leftarrow (h^{-1}_{j,i}(h_{j,i'}\psi^{(\ell-1)}_{j,i'} + s_j), p^{(\ell-1)}_{j,i'})$
10:         end for
11:     end for
12:     for $i \in \mathcal{A}$ do
13:         if $q^{(\ell)}_{j,i} \neq i$ for $j \in \mathcal{N}_{\mathrm{v}}(i)$ then
14:             for $j \in \mathcal{N}_{\mathrm{v}}(i)$ do
15:                 Set $j'$ as the unique element of $\mathcal{N}_{\mathrm{v}}(i) \setminus \{j\}$
16:                 $(\psi^{(\ell)}_{j,i}, p^{(\ell)}_{j,i}) \leftarrow (\phi^{(\ell)}_{j',i}, q^{(\ell)}_{j',i})$
17:             end for
18:         else if $\phi^{(\ell)}_{j,i} \neq \phi^{(\ell)}_{j',i}$ for $j \in \mathcal{N}_{\mathrm{v}}(i)$, $j' \in \mathcal{N}_{\mathrm{v}}(i) \setminus \{j\}$ then
19:             $\hat{x}_i \leftarrow (\phi^{(\ell)}_{j,i} + \phi^{(\ell)}_{j',i})^{-1}\phi^{(\ell)}_{j,i}\phi^{(\ell)}_{j',i}$, $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i\}$
20:         else if $\phi^{(\ell)}_{j,i} = 0$ and $\ell \leq w_{\mathrm{c}}$ then
21:             $\hat{x}_i \leftarrow 0$, $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i\}$
22:         else
23:             $\hat{x}_i \leftarrow ?$, $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i\}$
24:         end if
25:     end for
26:     $\ell \leftarrow \ell + 1$
27: end while
```

of Eqs. (4), (5), namely 0. The second component $p^{(0)}_{j,i}$ of the initial message represents the index of the variable node which sends the initial message.

Steps 6–11 in Algorithm 2 execute the check node calculation at the ℓ-th iteration. Let $\mathcal{N}_{\mathrm{v}}(\mathcal{A})$ be the set of the indices of the check nodes adjacent to the variable nodes in $\mathcal{A}$, i.e., $\mathcal{N}_{\mathrm{v}}(\mathcal{A}) := \cup_{i \in \mathcal{A}} \mathcal{N}_{\mathrm{v}}(i)$. Note that the calculation of $\phi^{(\ell)}_{j,i}$ is carried out as the recurrence Eqs. (4), (5).

Steps 12–25 in Algorithm 2 execute the variable node calculation at the ℓ-th iteration. If $q^{(\ell)}_{j,i} = i$ for $j \in \mathcal{N}_{\mathrm{v}}(i)$, the first components of the incoming messages $\phi^{(\ell)}_{j,i}, \phi^{(\ell)}_{j',i}$ to the i-th variable node give $A_{w,i}$ and $B_{w,i}$. If the condition of Step 18 is true[†], i.e., the submatrix corresponding to the zigzag cycle is non-singular, the MAP decoding result is given as Step 19. Recall that there are no zigzag cycles of weight in $[1, w_{\mathrm{c}}]$ with $\beta \neq 1$. If Step 20 is true, i.e., $\beta \neq 1$ and $\phi^{(\ell)}_{j,i} = 0$, then we set $\hat{x}_i = 0$. Otherwise, MAP decoding has been failures since the submatrix corresponding to the zigzag cycle is singular. Hence, in Step 23, set the decoding result $\hat{x}_i$ as ? and remove the i-th variable node from the set of indices of the variable nodes in $\mathcal{A}$. If $\mathcal{A} = \emptyset$, then the algorithm stops.

Remark 1: Consider a zigzag cycle of weight w labeled with $h_{1,1}, h_{1,2}, \ldots, h_{w,w}, h_{w,1}$. The first components of the messages in Algorithm 2 satisfy for $i, \ell \in [1; w]$

$$a^{(\ell)}_{w,i} = \phi^{(\ell)}_{i-\ell,i-\ell} = \psi^{(\ell-1)}_{i-\ell,i-\ell+1}, \quad b^{(\ell)}_{w,i} = \phi^{(\ell)}_{i+\ell-1,i+\ell} = \psi^{(\ell-1)}_{i+\ell-1,i+\ell-1}.$$

4.3 Complexity of Algorithms 1 and 2

In this section, we evaluate the decoding complexity of the proposed decoding algorithm.

4.3.1 Time Complexity

At first, we consider the time complexity of Algorithm 1. In the iteration steps (Steps 17–29), Algorithm 1 calculates the messages in at most $|\mathcal{D}|$ variable nodes and at most $|\mathcal{D}|$ check nodes. The number of iterations of Algorithm 1 is determined from the topology of the residual graph after BP decoding. In the worst case, the number of iterations is $|\mathcal{D}|/2$. Hence, we conclude that an upper bound on the complexity of Algorithm 1 is $|\mathcal{D}|^2$.

Secondly, we consider the complexity of Algorithm 2. Let z be the number of variable nodes in the zigzag cycles with BP decoding erasures. It requires at most $4z$ finite field multiplication and $2z$ finite field addition to calculate the messages in Step 9. It also requires at most $2z$ finite field multiplication and z finite field addition to determine $\hat{x}_i$ in Step 19. The number of iterations of Algorithm 2 is equal to the maximum weight of the zigzag cycles constructed by the variable nodes with BP decoding erasures. Hence, the number of iterations is at most z. Thus, we conclude that Algorithm 2 requires at most $4z^2 + 2z$ finite field multiplication and $2z^2 + z$ finite field addition.

4.3.2 Space Complexity

The space complexity of Algorithm 1 is $2\Lambda'(1) + N$ since Algorithm 1 requires N memories to express the set $\mathcal{A}$ and $2\Lambda'(1)$ memories to express the messages, where $\Lambda'(1)$ gives the total number of edges in the Tanner graph. Similarly, the space complexity of Algorithm 2 is $4(m + \lceil \log_2 N \rceil)z + N$, where $\lceil \cdot \rceil$ is the ceiling function.

4.4 Comparison with MAP Decoder

For the non-binary LDPC codes over the BECs, some stopping sets except the zigzag cycles are also recoverable under MAP decoding. Hence the proposed decoding algorithm does not achieve the MAP decoding performance.

The generalized tree-structured expectation propagation (GTEP) [12] is an efficient maximum likelihood decoding algorithm[††] for the binary LDPC codes. The GTEP carries out the Gaussian elimination if there are no check nodes of degree one in a residual graph. The GTEP is easily extended to the non-binary LDPC codes.

The time complexity of the GTEP is roughly approximated by $O(|\mathcal{D}|^3)$ [12]. The space complexity of the

[†] Notice that j' in Step 18 is the unique element of $\mathcal{N}_{\mathrm{v}}(i) \setminus \{j\}$

[††] Since we assume that codewords are chosen uniformly, the maximum likelihood decoding achieves the MAP decoding performance.

GTEP is $O(m|\mathcal{D}|^2 + N)$ to carry out the Gaussian eliminations and to store the indices of variable nodes in residual graph. On the other hand, the time and space complexity of the proposed algorithm is upper bounded by $O(|\mathcal{D}|^2)$ and $O((m + \lceil \log_2 N \rceil)|\mathcal{D}| + N)$, respectively.

5. MP Decoder with Decoding on Zigzag Cycles for MBIOS Channel

In this section, we extend the decoding algorithm proposed in the previous section to the MBIOS channels. If we are able to detect the symbols with decoding failures for the MBIOS channels, the decoding algorithm proposed in the previous section is easily extended to the MBIOS channels. Hence, firstly, this section proposes a method to detect the symbols with decoding failures for the MBIOS channels. Secondly, we extend the decoding algorithm proposed in the previous section to the MBIOS channels.

5.1 Detection of the Symbols with Decoding Failures

To detect the symbols with decoding failures for the irregular non-binary LDPC codes over the MBIOS channels, we define the stationarity of decoding result.

Definition 2: The i-th decoding result is *stationary* if there exist $\gamma \in \mathbb{F}_{2^m}$ and L such that $D_i^{(\ell)} = \{\gamma\}$ for all $\ell > L$.

Note that the i-th symbol is *eventually correct* [13] if there exists L such that $D_i^{(\ell)} = \{x_i\}$ for all $\ell > L$, where x_i is the i-th transmitted symbol. Hence, if the i-th symbol is eventually correct, the i-th decoding result is stationary. Hence, we get the following fact.

Fact 1: If the i-th decoding result is not stationary, then the i-th symbol is not eventually correct.

Note that there is a possibility that the i-th symbol is not eventually correct if the i-th decoding result is stationary. However, in the case for the zigzag cycles, the i-th symbol is eventually correct if the i-th decoding result is stationary. To prove the above, we consider *zigzag cycle codes*. A code is a zigzag cycle code if the Tanner graph is represented by a single zigzag cycle. The proof of the following theorem is given in Appendix.

Theorem 1: We assume a zigzag cycle code of symbol code length w such that the corresponding matrix is non-singular. If the i-th decoding result is stationary, then the i-th decoding result is eventually correct.

From Fact 1 and Theorem 1, the following algorithm correctly detects the set of indices $\hat{\mathcal{D}}$ of symbols with decoding errors for zigzag cycles in Tanner graphs.

1. If the decoding results $(D_i^{(\ell)})_{i=1}^N$ satisfy the parity check constraints in Step 3 of the BP decoding algorithm, set $\hat{\mathcal{D}} = \emptyset$.
2. If the decoding round ℓ reaches the maximum number of iterations $L_{\max}$ in Step 3 of the BP decoding algorithm, determine $\hat{\mathcal{D}}$ from the decoding results $(D_i^{(\ell)})_{i=1}^N, (D_i^{(\ell-1)})_{i=1}^N, \dots, (D_i^{(\ell-s)})_{i=1}^N$ as follows:
$$\hat{\mathcal{D}} = \left\{ i \in [1; N] \mid \exists j \in [1; s] \text{ s.t. } D_i^{(\ell)} \neq D_i^{(\ell-j)} \right\}.$$

Thus, the detection of symbols with BP decoding errors is realized by storing the decoding results $(D_i^{(\ell)})_{i=1}^N$, $(D_i^{(\ell-1)})_{i=1}^N, \dots, (D_i^{(\ell-s)})_{i=1}^N$. We refer to the parameter s as the *storing size*. Note that there are cases where the proposed error detection algorithm does not detect symbols with BP decoding errors if the symbols with decoding errors do not form of a zigzag cycle.

5.2 Symbol-Wise Peeling Algorithm

A trapping set $\mathcal{T}$ [13] is a set of variable nodes which are not eventually correct. For the BEC case, a trapping set forms a stopping set. For the MBIOS channel case, it is known that there exist neighbors of $\mathcal{T}$ adjacent to $\mathcal{T}$ only *once*.

Hence, we see that there exist neighbors of $\hat{\mathcal{D}}$ adjacent to $\hat{\mathcal{D}}$ only once. We denote the set of indices of the check nodes adjacent to $\hat{\mathcal{D}}$ only once, by $\mathsf{C}_1(\hat{\mathcal{D}})$. Since we regard the variable nodes in $[1; N] \setminus \hat{\mathcal{D}}$ as correct, we are able to decode the symbols in the variable nodes adjacent to the check node in $\mathsf{C}_1(\hat{\mathcal{D}})$.

To decode such variable nodes, we employ the *symbol-wise peeling decoder*. The symbol-wise peeling decoder for the non-binary LDPC codes is a straightforward extension of the peeling decoder [9] for the binary LDPC codes. The symbol-wise peeling decoding algorithm with inputs $\hat{\mathcal{D}}$ and $(\hat{x}_i^{\mathrm{BP}})_{i=1}^N$ is described in the following:

1. Set $\hat{x}_i^{\mathrm{PA}} \leftarrow \hat{x}_i^{\mathrm{BP}}$ and $\hat{\mathcal{D}}^{\mathrm{PA}} \leftarrow \hat{\mathcal{D}}$.
2. If $\mathsf{C}_1(\hat{\mathcal{D}}^{\mathrm{PA}}) = \emptyset$, the decoder terminates and outputs $(\hat{x}_i^{\mathrm{PA}})_{i=1}^N$ and $\hat{\mathcal{D}}^{\mathrm{PA}}$.
3. Choose an index j of a check node from $\mathrm{C}_1(\hat{\mathcal{D}}^{\mathrm{PA}})$ uniformly. Let i be the unique element in $\hat{\mathcal{D}}^{\mathrm{PA}} \cap \mathcal{N}_{\mathrm{c}}(j)$. Set $\hat{x}_i^{\mathrm{PA}} \leftarrow h_{j,i}^{-1} \sum_{i' \in \mathcal{N}_{\mathrm{c}}(j) \setminus \{v\}} h_{j,i'} \hat{x}_{i'}^{\mathrm{PA}}$ and $\hat{\mathcal{D}}^{\mathrm{PA}} \leftarrow \hat{\mathcal{D}}^{\mathrm{PA}} \setminus \{i\}$.
4. Go to Step 2.

Since this algorithm is a straightforward extension of the binary peeling decoder, the algorithm is realized by an MP algorithm.

5.3 Decoding Algorithm for MBIOS Channels

In this section, we propose an MP decoding algorithm with decoding on zigzag cycles for the MBIOS channels. The MP decoding algorithm with decoding on zigzag cycles is divided into the following 4 steps:

1. BP decoder (Sect. 2.3) with error detection (Sect. 5.1),
2. symbol-wise peeling algorithm (Sect. 5.2),
3. zigzag cycle detection (Sect. 4.1),
4. decoding on zigzag cycles (Sect. 4.2).

Figure 4 illustrates the data flow of the MP decoder with decoding on zigzag cycles for the MBIOS channels.

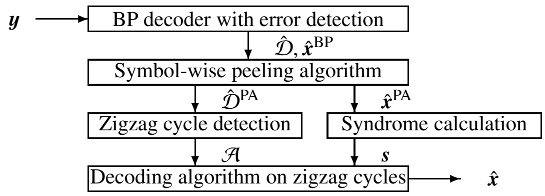

$\boldsymbol{y}$
BP decoder with error detection
$\hat{\mathcal{D}}, \hat{\boldsymbol{x}}^{\mathrm{BP}}$
Symbol-wise peeling algorithm
$\hat{\mathcal{D}}^{\mathrm{PA}}$
$\hat{\boldsymbol{x}}^{\mathrm{PA}}$
Zigzag cycle detection
Syndrome calculation
$\mathcal{A}$
$\boldsymbol{s}$
Decoding algorithm on zigzag cycles
$\hat{\boldsymbol{x}}$

Fig. 4 Data flow of the proposed decoding algorithm.

The BP decoder with error detection gives the set $\hat{\mathcal{D}}$ of indices of variable nodes with BP decoding errors and BP decoding outputs $\hat{\boldsymbol{x}}^{\mathrm{BP}}$ from the channel outputs $\boldsymbol{y}$. By using $\hat{\mathcal{D}}$ and $\hat{\boldsymbol{x}}^{\mathrm{BP}}$, the symbol-wise peeling algorithm decides the set $\hat{\mathcal{D}}^{\mathrm{PA}}$ of indices of variable nodes with decoding errors and the decoding outputs $\hat{\boldsymbol{x}}^{\mathrm{PA}}$. The zigzag cycle detection determines the set $\mathcal{A}$ of indices of variable nodes in the zigzag cycles with decoding errors from $\hat{\mathcal{D}}^{\mathrm{PA}}$. The syndrome calculation derives the syndrome $\boldsymbol{s}$ from $\hat{\boldsymbol{x}}^{\mathrm{PA}}$. From $\mathcal{A}$ and syndrome $\boldsymbol{s}$, the decoding algorithm on zigzag cycles gives the decoding output $\hat{\boldsymbol{x}}$ of proposed decoder.

5.4 Decoding Complexity

For the error detection, the proposed decoding algorithm requires msN memories and $(s-1)N$ equality testing. The symbol-wise peeling algorithm needs $mN + n$ memories to express $\{\hat{x}_i^{\mathrm{PA}}\}_{i=1}^{N}$ and $\hat{\mathcal{D}}^{\mathrm{PA}}$. Moreover, in Step 3, the symbol-wise peeling algorithm executes $\deg(\mathtt{c}_j)$ finite field multiplications and $\deg(\mathtt{c}_j) - 2$ finite field addition, where $\deg(\mathtt{c}_j)$ is the degree of the check node $\mathtt{c}_j$. Since the number of iterations of the symbol-wise peeling algorithm is upper bounded by $|\hat{\mathcal{D}}|$, in total, the symbol-wise peeling algorithm execute at most $|\hat{\mathcal{D}}|\mathrm{Deg}(P(x))$ finite field multiplications, where $\mathrm{Deg}(P(x))$ is the degree of the polynomial $P(x)$, i.e., the maximum degree of the check nodes.

6. Simulation Result

This section compares the symbol error rates in the error floors for non-binary LDPC codes over the BECs and the BAWGN channels by the BP decoding algorithm with by the proposed decoding algorithm. In this section, as a regular LDPC code, we employ the (2,4)-regular non-binary LDPC codes over $\mathbb{F}_{2^6}$ with symbol code length 168. As an example of irregular LDPC codes, we employ the irregular non-binary LDPC codes with the degree distribution pair $\Lambda(x) = 195x^2 + 26x^3 + 29x^4 + 2x^5$, $P(x) = 36x^4 + 90x^5$ over $\mathbb{F}_{2^4}$ [4].

Figures 5(a) and 5(b) demonstrate the symbol erasure rates for the regular and irregular non-binary LDPC codes over the BEC, respectively. The red curves and the green curves give the symbol erasure rates by the BP algorithm and the proposed decoding algorithm, respectively. The dashed lines and the solid lines give the symbol erasure rates for the non-binary LDPC codes constructed in the cycle cancellation [5] and the improved cycle cancellation [11], respectively. From Figs. 5(a) and 5(b), we see that the proposed decoding algorithm improves the decoding erasure rates in the error floor. Moreover, we see that the non-binary LDPC codes constructed in the improved cycle cancellation outperform those constructed in the cycle cancellation for the proposed decoding algorithm.

Figure 6(a) (resp. Fig. 6(b)) compares the symbol erasure rates for the regular (resp. irregular) non-binary LDPC codes constructed by the algorithm proposed in [6] over the BAWGN channels under the BP algorithm and the proposed decoding algorithm with the storing size $s = 1, 2, 3$. From Figs. 6(a) and 6(b), we see that the proposed decoding algorithm improves the decoding error rates in the error floors. In those cases, there are not much difference in the symbol error rates among $s \in [2; 10]$.

7. Conclusion

In this paper, we have proposed a decoding algorithm which lower decoding error rates in the error floors for non-binary LDPC codes over the MBIOS channel. We have shown that the proposed decoding algorithm is realized by means of an MP algorithm.

Acknowledgment

This work was partially supported by a Grant-in-Aid for JSPS Fellows from Japan Society for the Promotion of Science. The work of K. Kasai was supported by the grant from the Storage Research Consortium.

References

[1] R.G. Gallager, Low Density Parity Check Codes, in Research Monograph series, MIT Press, Cambridge, 1963.

[2] T. Richardson, M.A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," IEEE Trans. Inf. Theory, vol.47, no.2, pp.619–637, Feb. 2001.

[3] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," IEEE Commun. Lett., vol.2, no.6, pp.165–167, June 1998.

[4] X.Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," IEEE Trans. Inf. Theory, vol.51, no.1, pp.386–398, Jan. 2005.

[5] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular $(2,d_c)$-LDPC codes over GF(q) using their binary images," IEEE Trans. Commun., vol.56, no.10, pp.1626–1635, Oct. 2008.

[6] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analysis of error floors of non-binary LDPC codes over MBIOS channel," IEICE Trans. Fundamentals, vol.E94-A, no.11, pp.2144–2152, Nov. 2011.

[7] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error-Correcting Codes, Elsevier, Amsterdam, 1977.

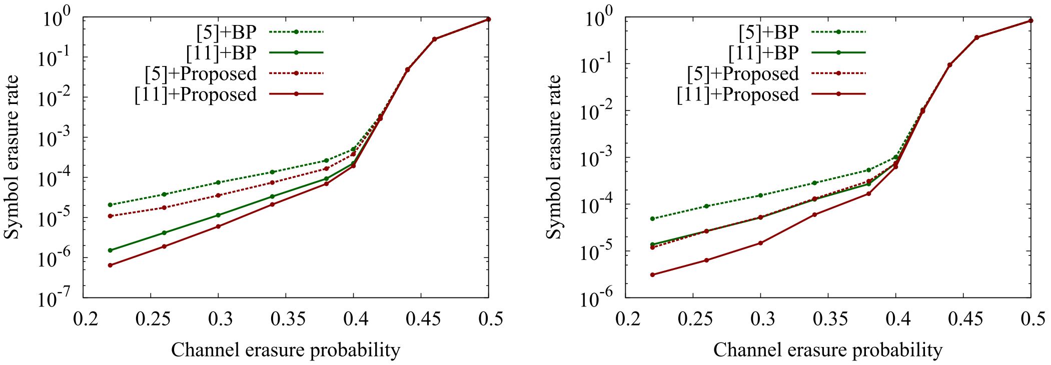
[8] V. Rathi, "Non-binary LDPC codes and EXIT like functions," Ph.D. Thesis, EPFL, Lausanne, 2008.

[9] M. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, and V. Stemann, "Practical loss-resilient codes," Proc. 29th annual ACM Symposium on Theory of Computing, pp.150–159, 1997.

[10] W. Mayeda, Graph Theory, Wiley-Interscience, 1972.

[11] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analysis of error floors of non-binary LDPC codes over BEC," IEICE Trans. Fundamentals, vol.E95-A, no.1, pp.381–390, Jan. 2012.

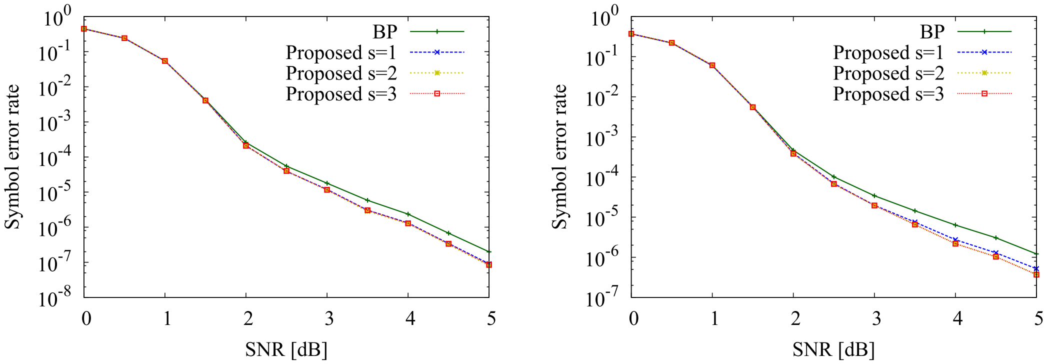
[12] L. Salamanca, P. Olmos, J. Murillo-Fuentes, and F. Perez-Cruz, "Tree expectation propagation for ML decoding of LDPC codes over the BEC," vol.61, no.2, pp.465–473, 2013.



(a) Regular case

(b) Irregular case

Fig. 5 The symbol erasure rates for non-binary LDPC codes constructed in the cycle cancellation [5] and the improved cycle cancellation [11] under the BP algorithm and the proposed decoding algorithm through the BEC.



(a) Regular case

(b) Irregular case

Fig. 6 The symbol error rates for non-binary LDPC codes constructed by the improved cycle cancellation [6] under the BP algorithm and the proposed decoding algorithm with the storing size $s = 1, 2, 3$ through the BAWGN channel.

[13] T.J. Richardson, “Error floors of LDPC codes,” Proc. 41th Annual Allerton Conf. on Commun., Control and Computing, pp.1426–1435, Oct. 2003.

[14] V. Rathi, “Conditional entropy of non-binary LDPC codes over the BEC,” Proc. 2008 IEEE Int. Symp. Inf. Theory (ISIT), pp.945–949, July 2008.

Appendix: Proof of Theorem 1

Without loss of generality, we assume the zigzag cycle code of weight w labeled with $h_{1,1}, h_{1,2}, h_{2,2}, \ldots, h_{w,1}$, as described in Fig. 1. We assume that the all-zero codewords are sent without loss of generality for analyzing the decoding error rate [14]. Since the matrix corresponding to the zigzag cycle code is non-singular, the parameter β, defined in Eq. (1), is not equal to 1. Let the order of β be denoted by σ. Note that $\sigma > 1$ since $\beta \neq 1$.

Let $\tilde{\Psi}^{(\ell)}_{v,c}$ be the *unnormalized* message from the v-th variable node to the c-th check node at the ℓ-th iteration. For all $x \in \mathbb{F}_{2^m}$, $\ell \geq 1$ and $i \in [1; w]$, the unnormalized message in the zigzag cycle code of weight w is written as follows:

$$\tilde{\Psi}^{(0)}_{i,i-1}(x) := C_i(x), \quad \tilde{\Psi}^{(\ell)}_{i,i-1}(x) := C_i(x)\tilde{\Psi}^{(\ell-1)}_{i+1,i}(\gamma_i^{-1}x),$$

$$\tilde{\Psi}^{(0)}_{i,i}(x) := C_i(x), \qquad \tilde{\Psi}^{(\ell)}_{i,i}(x) := C_i(x)\tilde{\Psi}^{(\ell-1)}_{i-1,i-1}(\gamma_{i-1}x),$$
$$\tilde{D}^{(\ell)}_i(x) := C_i(x)\Psi^{(\ell-1)}_{i-1,i-1}(\gamma_{i-1}x)\tilde{\Psi}^{(\ell-1)}_{i+1,i}(\gamma_i^{-1}x),$$

where $\gamma_i := h^{-1}_{i,i}h_{i,i+1}$, $\gamma_0 = \gamma_w$ and $\tilde{\Psi}^{(\ell)}_{w+i,w+j} = \tilde{\Psi}^{(\ell)}_{i,j}$ for $i, j \in [1, w]$. These notations give $D^{(\ell)}_i = \operatorname{argmax}_{x\in\mathbb{F}_{2^m}} \tilde{D}^{(\ell)}_i(x)$.

In the same way to the proof of Theorem 1 in [6], we have for all $t \geq 1$, $x \in \mathbb{F}_{2^m}$ and $i \in [1; w]$

$$\tilde{D}^{(\sigma t w)}_i(x) = \{B(\chi_i x)\}^{2t} C_i(x),$$
$$\tilde{D}^{(\sigma t w-1)}_i(x) = \{B(\chi_i x)\}^{2t}\{C_i(x)\}^{-1}.$$

where $\chi_i = \prod_{j=1}^{i-1}\gamma_j$ and $B(x) := \prod_{s=0}^{\sigma-1}\prod_{k=1}^{w} C_k(\beta^s x \prod_{j=k}^{w} \gamma_j)$. Note that $B(x) = B(\beta x)$ for all $x \in \mathbb{F}_{2^m}$. From these equations, we have for all $t \geq 1$, $x \in \mathbb{F}_{2^m}$ and $i \in [1; w]$

$$\begin{aligned}\tilde{D}^{(\sigma t w)}_i(x)\tilde{D}^{(\sigma t w-1)}_i(x) &= \{B(\chi_i x)\}^{4t}\\ &= \{B(\beta\chi_i x)\}^{4t}\\ &= \tilde{D}^{(\sigma t w)}_i(\beta x)\tilde{D}^{(\sigma t w-1)}_i(\beta x). \qquad (\mathrm{A}\cdot 1)\end{aligned}$$

We will show that the i-th symbol is eventually correct if the i-th decoding result is stationary. Recall that the i-th decoding result is stationary if there exist L and $x \in \mathbb{F}_{2^m}$ such that $\tilde{D}_i^{(\ell)}(x) > \tilde{D}_i^{(\ell)}(y)$ for all $\ell > L$ and $y \in \mathbb{F}_{2^m} \setminus \{x\}$. From Eq. (A· 1), for all $t \geq 1$, if $\tilde{D}^{(\sigma tw)}(x) > \tilde{D}^{(\sigma tw)}(y)\ \ \forall y \in \mathbb{F}_{2^m} \setminus \{x\}$, namely $\tilde{D}^{(\sigma tw)}(x) > \tilde{D}^{(\sigma tw)}(\beta x)$, then $\tilde{D}_i^{(\sigma tw-1)}(\beta x) > \tilde{D}_i^{(\sigma tw-1)}(x)$. Hence, the i-th decoding result is not stationary if the i-th symbol is not correct. □

Takayuki Nozaki received B.E., M.E. and D.E. degrees from Tokyo Institute of Technology in 2008, 2010 and 2012, respectively. From April 2010 to March 2013, he is a Research Fellow of the Japan Society for the Promotion of Science. He has been a Research Associate with Faculty of Engineering, Kanagawa University since April 2013. His research interests are codes on graph and iterative decoding algorithm. He is a member of IEEE.

Kenta Kasai received B.E., M.E. and Ph.D. degrees from Tokyo Institute of Technology in 2001, 2003 and 2006, respectively. Since April 2012, he has been an associate professor in the Department of Communications and Computer Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests include codes on graphs and iterative decoding algorithms.

Kohichi Sakaniwa received B.E., M.E. and Ph.D. degrees all in electronic engineering from Tokyo Institute of Technology, Tokyo Japan, in 1972, 1974 and 1977, respectively. He joined Tokyo Institute of Technology in 1977 as a research associate and served as an associate professor from 1983 to 1991. Since 1991 he has been a professor in the Department of Electrical and Electronic Engineering, and since 2000 in the Department of Communications and Computer Engineering, Graduate School of Science and Engineering, both in Tokyo Inst. of Tech. From November 1987 to July 1988, he stayed at the University of Southwestern Louisiana as a Visiting Professor. He received the Excellent Paper Award from the IEICE of Japan in 1982, 1990, 1992 and 1994. His research area includes Communication Theory, Error Correcting Coding, (Adaptive) Digital Signal Processing and so on. Dr. Sakaniwa is a member of IEEE, Information Processing Society of Japan, and Institute of Image Information and Television Engineers of Japan.