Message Passing Algorithm with MAP Decoding on Zigzag Cycles for Non-binary LDPC Codes

Takayuki Nozaki
Kanagawa University, JAPAN
Email: nozaki@kanagawa-u.ac.jp

Kenta Kasai, and Kohichi Sakaniwa
Tokyo Institute of Technology, JAPAN
Email: {kenta, sakaniwa}@comm.ce.titech.ac.jp

***Abstract*—In this paper, we propose a decoding algorithm which lowers decoding erasure rates in the error floor regions for non-binary low-density parity-check codes transmitted over the binary erasure channels. This decoding algorithm is a combination with belief propagation (BP) decoding and maximum a posteriori (MAP) decoding on zigzag cycles, which cause decoding erasures in the error floor region. We show that MAP decoding on the zigzag cycles is realized by means of a message passing algorithm. A simulation result shows that the decoding erasure rates in the error floor regions by the proposed decoding algorithm are lower than those by the BP decoder.**

I. 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 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 binary erasure channel (BEC), 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. It is known that each LDPC code is represented by Tanner graphs.

The curve of the decoding erasure 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 region, the decoding erasure rate drops off steeply as the function of channel erasure probability. The waterfall region is mainly caused by the large weight erasures. In the error floor region, the decoding erasure rate has a gentle slope. The error floor region is mainly caused by the small weight erasures. This paper investigates the decoding erasure rates in the error floor regions for non-binary LDPC codes.

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. Hence, the stopping sets are important to characterize the decoding erasures for the LDPC codes transmitted over the BEC.

A *zigzag cycle* is a *simple cycle* such that the degrees of variable nodes are two in the Tanner graph. Since the stopping sets of small weight is caused by zigzag cycles for the LDPC codes [4], the decoding erasure rates in the error floors are mainly caused by zigzag cycles. Hence, in this paper, we focus on the decoding erasures in the zigzag cycles to lower the decoding erasure rates in the error floor regions.

There are two approaches to lowering the decoding erasure rates in the error floors: by optimizing the code and the decoder. By optimizing labels in the zigzag cycles in non-binary LDPC codes, the decoding erasure rates in the error floors were lowered by [5], [6]. On the other hand, there are no methods to lower the decoding erasure rates in the error floors by optimizing the decoders which lower the decoding erasures in the zigzag cycles for non-binary LDPC codes.

In this paper, we propose a message passing decoding algorithm which reduces the decoding erasures in the zigzag cycles. The proposed decoding algorithm is combined maximum a posteriori (MAP) decoding on zigzag cycles and BP decoding. Moreover, we show that MAP decoding on the zigzag cycles is realized by means of a message passing algorithm. A simulation result shows that the decoding erasure rates in the error floor regions by the proposed decoding algorithm is lower than those by the BP decoder.

The remainder of this paper is organized as follows: Section II defines non-binary LDPC codes and introduces BP decoding and MAP decoding for the non-binary LDPC codes over the BEC. Section III reviews a MAP decoding algorithm on the zigzag cycles and realizes MAP decoding on the zigzag cycles by message passing algorithm. Section IV proposes a decoding algorithm which reduces the decoding erasure rates in the error floors for non-binary LDPC codes over the BEC. Section V shows by a simulation result that the proposed decoding algorithm lowers the decoding erasure rates in the error floors.

II. Preliminaries

In this section, we define non-binary LDPC codes and recall BP decoding and MAP decoding for the non-binary LDPC codes over the BEC. This section introduces some notations throughout used in this paper.

A. Non-binary LDPC Codes

A non-binary LDPC code C over $\mathbb{F}_{2^m}$ is defined by the null space of a sparse parity-check matrix $H = (h_{i,j}) \in \mathbb{F}_{2^m}^{M\times N}$. 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 variable nodes, check nodes and labeled edges, respectively. The v-th variable node and the c-th check node are connecting to an edge labeled $h_{c,v} \in \mathbb{F}_{2^m} \setminus \{0\}$ if $h_{c,v} \neq 0$.

An LDPC code defined by a Tanner graph with the variable nodes of degree d_{v} and the check nodes of degree d_{c} is called a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC code. It is empirically known that $(2, d_{\mathrm{c}})$-regular non-binary LDPC codes exhibit good decoding performance among other LDPC codes for $2^m \geq 64$ [7]. In this paper, we consider the irregular non-binary LDPC codes containing variable nodes of degree two for the generality of code.

B. 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 [8, 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 $[a; b]$. More precisely, we define $[a; b] := \{n \in \mathbb{N} \mid a \leq n \leq b\}$. 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 $\mathbb{F}_{2^m}$ can be transmitted over binary channels. We denote the received word as $\boldsymbol{y} = (y_{1,1}, y_{1,2}, \dots, y_{N,m})$.

Let X and Y be the channel input and channel output, respectively. For the BEC, the channel input and channel output take value in the alphabet $X \in \{0, 1\}$ and $Y \in \{0, 1, ?\}$, respectively, where ? indicates an erasure. Each channel input is either erased with probability ϵ or received correctly with probability $1 - \epsilon$, where ϵ is referred to as *channel erasure probability*.

C. BP Decoding Algorithm

In this section, we recall the BP decoding algorithm for the non-binary LDPC codes over the BECs. The BP decoding algorithm is a message passing algorithm on the Tanner graph. In the case for the BEC, each message in the BP decoding algorithm for the BEC is represented as a set of candidate symbols for the decoding result [9, Lemma 2]. Let $\Psi_{c,v}^{(\ell)} \subseteq \mathbb{F}_{2^m}$ (resp. $\Phi_{c,v}^{(\ell)} \subseteq \mathbb{F}_{2^m}$) 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.

a) Initialization: Set $\ell = 0$. For $v \in [1; N]$, the initial message of the v-th variable node, denoted by $E_v \subseteq \mathbb{F}_{2^m}$, is given by the channel outputs $(y_{v,1}, y_{v,2}, \dots, y_{v,m})$ as

$$E_v = \{\gamma \mid b_i(\gamma) = y_{v,i} \quad (\text{for } i \text{ s.t. } y_{v,i} \in \{0, 1\})\}.$$

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. j-th check node). For $v \in [1; N]$ and $c \in \mathcal{N}_{\mathrm{v}}(v)$, set $\Phi_{c,v}^{(0)} = \mathbb{F}_{2^m}$.

b) Iteration and Decision: Define $\gamma S := \{\gamma s \mid s \in S\}$ for $\gamma \in \mathbb{F}_{2^m}$ and $S \subseteq \mathbb{F}_{2^m}$. For $S_1, S_2, \dots, S_k \subseteq \mathbb{F}_{2^m}$, we define $\sum_{i=1}^{k} S_k := \{\sum_{i=1}^{k} s_k \mid s_j \in S_j \ (\text{for } j \in [1; k])\}$.

1) In the v-th variable node, the message $\Psi_{c,v}^{(\ell)}$ is given by
$$\Psi_{c,v}^{(\ell)} = E_v \cap \left(\bigcap_{c' \in \mathcal{N}_{\mathrm{v}}(v) \setminus \{c\}} \Phi_{c',v}^{(\ell)}\right).$$
2) In the c-th check node, the message $\Phi_{c,v}^{(\ell+1)}$ is given by
$$\Phi_{c,v}^{(\ell+1)} = h_{c,v}^{-1} \left(\sum_{v' \in \mathcal{N}_{\mathrm{c}}(c) \setminus \{v\}} h_{c,v'} \Psi_{c,v'}^{(\ell)}\right).$$
3) In the v-th variable node, the decoding result $D_v^{(\ell)} \subseteq \mathbb{F}_{2^m}$ is calculated as
$$D_v^{(\ell+1)} = E_v \cap \left(\bigcap_{c' \in \mathcal{N}_{\mathrm{v}}(v)} \Phi_{c',v}^{(\ell+1)}\right).$$
If $D_v^{(\ell+1)} = D_v^{(\ell)}$ for all $v \in [1; N]$, the decoding output D_v is set as $D_v \leftarrow D_v^{(\ell+1)}$ for all $v \in [1; N]$. Otherwise, set $\ell \leftarrow \ell + 1$ and repeat Step 1.

If the cardinality of D_v is equal to 1, i.e., $|D_v| = 1$, the v-th variable node is correctly decoded. Otherwise, the v-th variable node has decoding erasure.

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

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

where d_i represents the unique element in D_i for $i \in \bar{\mathcal{D}}$.

We refer the subgraph constructed by the variable nodes with decoding erasures as *residual graph* [10] 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 a 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. Hence, the stopping sets are important to characterize the decoding erasures for the LDPC codes transmitted over the BECs.

D. MAP Decoding Algorithm

In this paper, we consider the *symbol-wise* MAP decoding using the decoding rule $\hat{x}_i(\boldsymbol{y}) = \operatorname{argmax}_{\gamma \in \mathbb{F}_{2^m}} p_{X_i|\boldsymbol{Y}}(\gamma \mid \boldsymbol{y})$, for a given received word $\boldsymbol{y} \in \boldsymbol{Y} = Y^{mN}$. In the case for the BEC, the received word $\boldsymbol{y}$ fulfills a condition such that $y_{i,j} = x_{i,j}$ if the bit $y_{i,j}$ is received correctly. Define

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

The symbol-wise MAP decoding for the i-th symbol is

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

The i-th symbol is correctly decoded if $|\hat{x}_i(\boldsymbol{y})| = 1$.

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

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

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 $\boldsymbol{s} = (s_1, s_2, \dots, s_m)$. Thus, the i-th symbol is correctly decoded by the symbol-wise MAP decoding *if* $H_{\mathcal{D}} \boldsymbol{x}_{\mathcal{D}}^T = \boldsymbol{s}^T$ has a unique solution.

III. MAP Decoding on Zigzag Cycles

This section defines the zigzag cycles and shows MAP decoding results for the zigzag cycles in non-binary LDPC codes.

A. Zigzag Cycle

Definition 1: A zigzag cycle is a simple cycle or a circuit such that the degrees of variable nodes are two in a Tanner graph. More precisely, a zigzag cycle $(\mathsf{V}_{\mathrm{zc}} \cup \mathsf{C}_{\mathrm{zc}}, \mathsf{E}_{\mathrm{zc}})$ is a *connected* subgraph [11] of a Tanner graph such that 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 stopping sets of small weight are caused by zigzag cycles for the LDPC codes [4], the decoding erasure rates in the error floors are mainly caused by zigzag cycles. Hence, we consider the decoding erasures on the zigzag cycles.

All the zigzag cycles cause decoding erasures under BP decoding for the BECs if all the bits in the zigzag cycles are received erasures, i.e., $\forall i \in \mathsf{V}_{\mathrm{zc}} \forall j \in [1; m]\ y_{i,j} = ?$ [6, 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.

B. 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 a zigzag cycle of weight w labeled with $h_{1,1}, h_{1,2}, h_{2,2}, \dots, h_{w,w}, h_{w,1}$.

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

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

Let s_i be the syndrome corresponding to the i-th check node. To simplify the notation, for $i \in [1; w]$ and $n \in \mathbb{Z}$, we define $h_{nw+i,nw+i} := h_{i,i}, h_{nw+i,nw+i+1} := h_{i,i+1}, \zeta_i := h_{i-1,i} h_{i,i}^{-1}$ and $\beta := \prod_{i=1}^{w} \zeta_i$. Note that an *empty product* is equal to 1, i.e., $\prod_{i \in \emptyset} a_i = 1$ for any $a_1, a_2, \dots, a_k \in \mathbb{F}$. For a given s_i for $i \in [1; w]$, 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., $\det H_{\mathrm{zc}} \neq 0 \iff \beta \neq 1$, then the (i,j)-th entry of H_{zc}^{-1} is given as

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

From (1), the MAP decoding result $\hat{x}_i$ is written as follows:

$$\hat{x}_i = \textstyle\sum_{j=1}^{w} \left(H_{\mathrm{zc}}^{-1}\right)_{i,j} s_j = (1+\beta)^{-1} A_{w,i}, \tag{2}$$

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

$$\beta = A_{w,i} B_{w,i}^{-1} \neq 1 \iff A_{w,i} \neq B_{w,i}. \tag{3}$$

By using (2) and $\beta = A_{w,i} B_{w,i}^{-1}$, the MAP decoding result $\hat{x}_i$ is rewritten as

$$\hat{x}_i = (A_{w,i} + B_{w,i})^{-1} A_{w,i} B_{w,i}. \tag{4}$$

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

Now, we consider two recurrence relations

$$a_{w,i}^{(\ell)} = h_{i-\ell,i-\ell}^{-1} \left(h_{i-\ell,i-\ell+1} a_{w,i}^{(\ell-1)} + s_{i-\ell}\right), \tag{5}$$

$$b_{w,i}^{(\ell)} = h_{i+\ell-1,i+\ell}^{-1} \left(h_{i+\ell-1,i+\ell-1} b_{w,i}^{(\ell-1)} + s_{i+\ell-1}\right), \tag{6}$$

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 (5), (6).

IV. Message Passing Decoder with MAP Decoding on Zigzag Cycle

In this section, we propose a decoding algorithm which reduces the decoding erasure rates in the error floors for non-binary LDPC codes over the BECs. 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 in 3 steps: (i) BP decoding (ii) zigzag cycle detection and (iii) MAP decoding on zigzag cycles. All the steps are realized by means of message passing algorithms. The algorithms of steps (i), (ii) and (iii) are given in Section II-C, IV-A and IV-B, respectively.

A. Zigzag Cycle Detection

If there are variable nodes with erasures after BP decoding, the proposed decoder searches the zigzag cycles with BP decoding erasures. More precisely, 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 a zigzag cycle if the connected subgraph contains some nodes 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 cycle is detailed in Algorithm 1. In this algorithm, there are three messages "0", "1" and "-1". The variable nodes which are successfully decoded in BP decoding send the messages "0". The variable nodes which are not in zigzag cycles with BP decoding erasures send the messages "-1". If variable nodes can be in zigzag cycles with BP decoding erasures, the variable nodes send the messages "1". Let $\mathcal{A}^{(\ell)}$ represent the set of indices of the variable nodes which can be in the zigzag cycles with BP decoding erasures at the ℓ-th iteration.

Steps 2-11 in Algorithm 1 give 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" and the position of the variable node is removed from $\mathcal{A}^{(0)}$. 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 give the initialization of the check nodes. In the case for a check node receiving the message "-1" or for a check node 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 in Algorithm 1 give the iteration 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. Moreover, the variable nodes which sends the messages "-1" are removed from $\mathcal{A}^{(\ell)}$. If $\mathcal{A}^{(\ell)} = \mathcal{A}^{(\ell-1)}$, the variable nodes in $\mathcal{A}^{(\ell)}$ are 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

Require: $\mathcal{D}$ is the set of indices of the variable node with BP decoding erasures
Ensure: $\mathcal{A}$ is the set of indices of the variable nodes in the zigzag cycle 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 i-th variable node is more than two **then**
4: i-th variable node sends "-1", $\mathcal{A}^{(0)} \leftarrow \mathcal{A}^{(0)} \setminus \{i\}$
5: **else**
6: i-th variable node sends "1"
7: **end if**
8: **end for**
9: **for** $i \in [1; N] \setminus \mathcal{D}$ **do**
10: i-th variable node sends "0"
11: **end for**
12: **for** $j \in [1; M]$ **do**
13: **if** j-th check node receives "-1" or more than two "1" **then**
14: 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** i-th variable node receives "-1" **then**
21: 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** j-th check node receives "-1" **then**
26: j-th check node sends "-1"
27: **end if**
28: **end for**
29: **until** $\mathcal{A}^{(\ell)} = \mathcal{A}^{(\ell-1)}$
30: $\mathcal{A} \leftarrow \mathcal{A}^{(\ell)}$

B. MAP decoding on Zigzag Cycle by Message Passing

In this section, we realize MAP decoding on the zigzag cycles by a message passing algorithm. In other words, we propose a message passing algorithm which gives (5), (6).

Let $\left(\psi_{c,v}^{(\ell)}, p_{c,v}^{(\ell)}\right) \in \mathbb{F}_{2^m} \times [1; N]$ (resp. $\left(\phi_{c,v}^{(\ell)}, q_{c,v}^{(\ell)}\right) \in \mathbb{F}_{2^m} \times [1; N]$) 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 variable nodes which are in the zigzag cycles with BP decoding erasures, by $\mathcal{A}$. MAP decoding on the zigzag cycles is detailed in Algorithm 2.

Steps 1-3 in Algorithm 2 give initialization of this algorithm. In those steps, all the variable nodes in the zigzag cycles send the initial messages. The first entry of the initial message $\phi_{j,i}^{(0)}$ represents the initial terms of (5), (6). The second entry of the initial message $p_{j,i}^{(0)}$ represents the index of the variable node which sends the initial message.

Steps 6-11 in Algorithm 2 give the check node calculation at the ℓ-th iteration. Let $\mathcal{N}_{\mathrm{v}}(\mathcal{A})$ be the set of indexes of the check nodes adjacent to the variable nodes in $\mathcal{A}$, i.e., $\mathcal{N}_{\mathrm{v}}(\mathcal{A}) := \cup_{i \in \mathcal{A}} \mathcal{N}_{\mathrm{v}}(i)$. Steps 12-20 in Algorithm 2 give the variable node calculation at the ℓ-th iteration. The first entries of the messages $\psi_{j,i}^{(\ell)}, \phi_{j,i}^{(\ell)}$ derive (5), (6). The second entries of the messages $p_{j,i}^{(\ell)}, q_{j,i}^{(\ell)}$ represent the indices of the variable nodes which send the initial message.

If $q_{j,i}^{(\ell)} = q_{j',i}^{(\ell)} = i$ for $j \in \mathcal{N}_{\mathrm{v}}(i)$ and $j' \in \mathcal{N}_{\mathrm{v}}(i) \setminus \{j\}$, the first entries of the incoming messages $\phi_{j,i}^{(\ell)}, \phi_{j',i}^{(\ell)}$ to the i-th variable node give $A_{w,i}$ and $B_{w,i}$. If the condition of Step 15 is true, i.e., the submatrix corresponding to the zigzag cycle is singular, the variable nodes in the zigzag cycle are not recovered under MAP decoding. Hence, in Step 16, set the decoding result $\hat{x}_i$ as ? and remove the i-th variable node from the set of indices of variable nodes in the zigzag cycle $\mathcal{A}$. If the condition of Step 15 is false, i.e., the submatrix corresponding to the zigzag cycle is non-singular, the MAP decoding result is given as Step 18. If $\mathcal{A} = \emptyset$, then the algorithm stops.

Remark 1: Consider a zigzag cycle of weight w labeled with $h_{1,1}, h_{1,2}, \dots, h_{w,w}, h_{w,1}$. From (5), (6), we see that the first entries of the messages in Algorithm 2 satisfy for $i, \ell \in [1; w]$

$$a_{w,i}^{(\ell)} = \phi_{i-\ell,i-\ell}^{(\ell)} = \psi_{i-\ell,i-\ell+1}^{(\ell-1)},$$
$$b_{w,i}^{(\ell)} = \phi_{i+\ell-1,i+\ell}^{(\ell)} = \psi_{i+\ell-1,i+\ell-1}^{(\ell-1)}.$$

C. Decoding Complexity

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

At first, we consider the 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

Algorithm 2 MAP decoding on zigzag cycles

Require: $\mathcal{A}$ is the set of indices of the variable nodes in the zigzag cycle with BP decoding erasures
1: **for** $i \in \mathcal{A}$ **do**
2: $\quad(\psi^{(0)}_{j,i}, p^{(0)}_{j,i}) \leftarrow (0, i)\ \forall j \in \mathcal{N}_{\mathrm{v}}(i)$
3: **end for**
4: $\ell \leftarrow 1$
5: **while** $\mathcal{A} \neq \emptyset$ **do**
6: $\quad$**for** $j \in \mathcal{N}_{\mathrm{v}}(\mathcal{A})$ **do**
7: $\quad\quad$**for** $i \in \mathcal{N}_{\mathrm{c}}(j) \cap \mathcal{A}$ **do**
8: $\quad\quad\quad$Set i' as the unique element of $(\mathcal{N}_{\mathrm{c}}(j) \cap \mathcal{A}) \setminus \{i\}$
9: $\quad\quad\quad(\phi^{(\ell)}_{j,i}, q^{(\ell)}_{j,i}) \leftarrow (h^{-1}_{j,i}(h_{j,i'}\psi^{(\ell-1)}_{j,i'} + s_j), p^{(\ell-1)}_{j,i'})$
10: $\quad\quad$**end for**
11: $\quad$**end for**
12: $\quad$**for** $i \in \mathcal{A}$ **do**
13: $\quad\quad$**if** $q^{(\ell)}_{j,i} \neq i$ **or** $q^{(\ell)}_{j',i} \neq i$ for $j \in \mathcal{N}_{\mathrm{v}}(i)$, $j' \in \mathcal{N}_{\mathrm{v}}(i) \setminus \{j\}$ **then**
14: $\quad\quad\quad(\psi^{(\ell)}_{j,i}, p^{(\ell)}_{j,i}) \leftarrow (\phi^{(\ell)}_{j',i}, q^{(\ell)}_{j',i})\quad \forall j \in \mathcal{N}_{\mathrm{v}}(i)$, $j' \in \mathcal{N}_{\mathrm{v}}(i) \setminus \{j\}$
15: $\quad\quad$**else if** $\phi^{(\ell)}_{j,i} = \phi^{(\ell)}_{j',i}$ for $j \in \mathcal{N}_{\mathrm{v}}(i)$, $j' \in \mathcal{N}_{\mathrm{v}}(i) \setminus \{j\}$ **then**
16: $\quad\quad\quad\hat{x}_j \leftarrow ?$, $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i\}$
17: $\quad\quad$**else**
18: $\quad\quad\quad\hat{x}_i \leftarrow (\phi^{(\ell)}_{j,i} + \phi^{(\ell)}_{j',i})^{-1}\phi^{(\ell)}_{j,i}\phi^{(\ell)}_{j',i}$, $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i\}$
19: $\quad\quad$**end if**
20: $\quad$**end for**
21: $\quad\ell \leftarrow \ell + 1$
22: **end while**

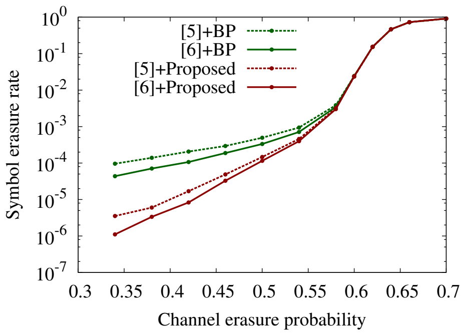

Fig. 1. The symbol erasure rates for the (2,3)-regular non-binary LDPC codes with symbol code length 315 over $\mathbb{F}_{2^4}$. The green and red lines give the symbol erasure rates by the BP algorithm and the proposed decoding algorithm, respectively. The dashed and solid lines give the symbol erasure rates for the non-binary LDPC codes constructed in the cycle cancellation [5] and the improved cycle cancellation [6], respectively.

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 concludes that an upper bound of 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 $4z$ finite field multiplication and $2z$ finite field addition to determine $\hat{x}_i$ in Step 18. 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 concludes that Algorithm 2 requires at most $4z^2+4z$ finite field multiplication and $4z^2+4z$ finite field addition.

V. Simulation Result

In this section, we compare the symbol erasure rates in the error floors for non-binary LDPC codes over the BECs by the BP algorithm with by the proposed decoding algorithm.

Figure 1 shows the symbol erasure rates for the (2,3)-regular non-binary LDPC codes with symbol code length 315 over $\mathbb{F}_{2^4}$. The green and red lines give the symbol erasure rates by the BP algorithm and the proposed decoding algorithm, respectively. The dashed and solid lines give the symbol erasure rates for the non-binary LDPC codes constructed in the cycle cancellation [5] and the improved cycle cancellation [6], respectively. From Fig. 1, we see that the proposed decoding algorithm improves the decoding erasure rates in the error floor region. 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.

VI. Conclusion

In this paper, we have proposed a decoding algorithm which lower decoding erasure rates in the error floor regions for non-binary LDPC codes over the BECs. The proposed decoding algorithm is a combination with the BP decoding algorithm and the MAP decoding algorithm on the zigzag cycles. We have shown that the proposed decoding algorithm is realized by means of a message passing algorithm. The proposed algorithm is extended to the memoryless binary-input output-symmetric channels in [12].

Acknowledgment

This work was partially supported by Grant-in-Aid for JSPS Fellows.

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, Jun. 1998.

[4] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.

[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 BEC,” *IEICE Transactions*, vol. 95-A, no. 1, pp. 381–390, 2012.

[7] 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.

[8] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[9] V. Rathi, “Non-binary LDPC codes and EXIT like functions,” Ph.D. dissertation, EPFL, Lausanne, 2008. [Online]. Available: http://library.epfl.ch/theses/?nr=4111

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

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

[12] T. Nozaki, K. Kasai, and K. Sakaniwa, “Message passing decoder with decoding on zigzag cycles for non-binary LDPC codes,” *IEICE Transactions*, submitted to.