PAPER *Special Section on Information Theory and Its Applications*

A Method for Grouping Symbol Nodes of Group Shuffled BP Decoding Algorithm

Yoshiyuki SATO[†a)], ***Nonmember***, **Gou HOSOYA**[††], **Hideki YAGI**[†††], ***Members***, ***and*** **Shigeichi HIRASAWA**[††], ***Fellow***

SUMMARY In this paper, we propose a method for enhancing performance of a sequential version of the belief-propagation (BP) decoding algorithm, the group shuffled BP decoding algorithm for low-density parity-check (LDPC) codes. An improved BP decoding algorithm, called the shuffled BP decoding algorithm, decodes each symbol node in serial at each iteration. To reduce the decoding delay of the shuffled BP decoding algorithm, the group shuffled BP decoding algorithm divides all symbol nodes into several groups. In contrast to the original group shuffled BP, which automatically generates groups according to symbol positions, in this paper we propose a method for grouping symbol nodes which generates groups according to the structure of a Tanner graph of the codes. The proposed method can accelerate the convergence of the group shuffled BP algorithm and obtain a lower error rate in a small number of iterations. We show by simulation results that the decoding performance of the proposed method is improved compared with those of the shuffled BP decoding algorithm and the group shuffled BP decoding algorithm.

key words: *low-density parity-check code, belief propagation decoding, shuffled BP decoding, Tanner graph, symbol nodes grouping method*

1. Introduction

Low-density parity-check (LDPC) codes [1] proposed by R. G. Gallager in 1962 have been forgotten for a long time, and recently their performance is re-focused [2]. The belief-propagation (BP) decoding algorithm is a well-known as iterative decoding algorithm of LDPC codes [1], [2]. In this paper, we call this decoding algorithm the *standard* BP decoding algorithm. The standard BP decoding algorithm attains performance near to the Shannon limit when using the LDPC codes with a large code length [3], [4]. Moreover, its decoding complexity is linear to the code length.

The standard BP decoding algorithm calculates a posterior probability for each received symbol. This algorithm can be regarded as a message passing algorithm on a Tanner graph, and it processes all symbol nodes in parallel at each iteration. The standard BP decoding algorithm requires a large number of iterations to achieve high performance. This fact indicates necessity to accelerate convergence of the decoding algorithm, and an improved version of the standard BP decoding algorithm, the shuffled BP (SBP) decoding algorithm has been discussed [5], [7]. The SBP decoding algorithm processes symbol nodes in serial at each iteration, resulting in its high performance within a less number of iterations. However this decoding algorithm causes a large decoding delay due to updating messages for each symbol node in serial. To reduce the decoding delay of the SBP decoding algorithm, the group SBP decoding algorithm has been devised [5], [7]. This decoding algorithm divides all symbol nodes into several groups. It processes each group in serial, and all symbol nodes in a same group in parallel. Though it automatically generates groups according to the order of symbol positions, no methods for composing each group to accelerate the convergence of the algorithm have been considered.

In this paper, we propose a symbol nodes grouping method, which generates groups by taking the structure of a given Tanner graph into account. The proposed method can accelerate the convergence of the grouping SBP decoding algorithm and improve error rates in a small number of iterations. We show by simulation results that the decoding performance of the proposed method is better than those of the SBP decoding algorithm and the group SBP decoding algorithm within a small number of iterations. Some simulation results indicate that the group SBP decoding algorithm of the proposed method attains almost the same error rates as the conventional group SBP decoding algorithm of a greater number of groups. In general, the group SBP decoding algorithm attains lower error rates but causes larger decoding delay as the number of groups increases. The proposed method can decrease decoding delay due to its high decoding performance with a small number of groups.

This paper is organized as follows. In Sect. 2, we describe LDPC codes and the standard BP decoding algorithm. In Sect. 3, we briefly review the SBP and the group SBP decoding algorithms. In Sect. 4, we propose a grouping method for the group SBP decoding algorithm. An overview of the proposed method is given in Sect. 4.1. In Sect. 4.2, we explain procedures of the proposed algorithm in detail. Some simulation results and discussions are presented in Sect. 5, and conclusion and further works are given in Sect. 6.

Manuscript received January 25, 2008.
Manuscript revised April 21, 2008.

[†]The author is with the Graduate School of Creative Science and Engineering, Waseda University, Tokyo, 169-8555 Japan.
[††]The authors are with the School of Creative Science and Engineering, Waseda University, Tokyo, 169-8555 Japan.
[†††]The author is with The University of Electro-Communications, Chofu-shi, 182-8585 Japan.
a) E-mail: yosiyuki@hirasa.mgmt.waseda.ac.jp

DOI: 10.1093/ietfec/e91–a.10.2745

2. Preliminaries

2.1 LDPC Codes and Channel Model

An LDPC code is defined by a sparse parity-check matrix $H = [H_{mn}]$, where $H_{mn} \in \{0, 1\}$ for $m = 1, 2, \cdots, M, n = 1, 2, \cdots, N$, and M and N denote the number of rows and columns of H, respectively. Throughout this paper, we assume only binary regular LDPC codes, whose parity-check matrices H have a constant number of 1's in each row and that in each column. We call the number of 1's in each row the weight of rows and denote it by w_r. Similarly, we call the number of 1's in each column the weight of columns and denote it by w_c.

Assume that a codeword $\boldsymbol{x}$, $\boldsymbol{x}H^{\mathrm{T}} = \boldsymbol{0}$, is transmitted through the additive white Gaussian noise (AWGN) channel of signal to noise ratio E_b/N_0, where H^{T} denotes the transposed matrix of H. A decoder estimates the transmitted sequence from a received sequence $\boldsymbol{y} \in \mathcal{R}^N$.

2.2 Standard BP Decoding Algorithm

The standard BP decoding algorithm [1], [2] simultaneously processes all code symbols by calculating their a posterior probability. For a received sequence $\boldsymbol{y} = (y_1, y_2, \cdots, y_N)$, let λ_n, $n = 1, 2, \cdots, N$, be the log likelihood ratio given by

$$\lambda_n = \log \frac{\Pr(y_n | x_n = 0)}{\Pr(y_n | x_n = 1)}. \tag{1}$$

For a parity-check matrix H, let $\mathcal{N}(m)$ and $\mathcal{M}(n)$ be index sets of row m and column n such that $H_{mn} = 1$, respectively, i.e.,

$$\mathcal{N}(m) \stackrel{\triangle}{=} \{n : H_{mn} = 1\}, \quad m = 1, 2, \cdots, M, \tag{2}$$

$$\mathcal{M}(n) \stackrel{\triangle}{=} \{m : H_{mn} = 1\}, \quad n = 1, 2, \cdots, N. \tag{3}$$

[Standard BP Decoding Algorithm]

Initialization) For any (m, n) such that $H_{mn} = 1$, set $z_{mn}^{(0)} := \lambda_n$. Set $i := 1$ and the maximum number of iteration $I_{\max}$ to some constant value.

Step 1-a) For $n = 1, 2, \cdots, N$, perform the following horizontal step.

(Horizontal step) For $m \in \mathcal{M}(n)$, calculate the following equations:

$$\tau_{mn}^{(i)} := \prod_{n' \in \mathcal{N}(m) \backslash n} \tanh(z_{mn'}^{(i-1)}/2), \tag{4}$$

$$\varepsilon_{mn}^{(i)} := \log \frac{1 + \tau_{mn}^{(i)}}{1 - \tau_{mn}^{(i)}}. \tag{5}$$

Step 1-b) For $n = 1, 2, \cdots, N$, perform the following vertical step.

(Vertical step) For $m \in \mathcal{M}(n)$, calculate the following equation:

$$z_{mn}^{(i)} := \lambda_n + \sum_{m' \in \mathcal{M}(n) \backslash m} \varepsilon_{m'n}^{(i)}. \tag{6}$$

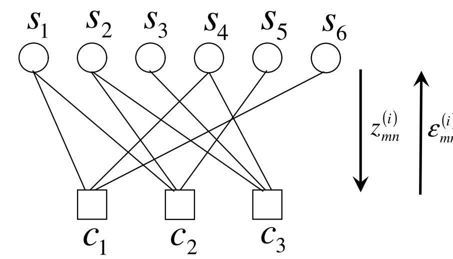

Fig. 1 An example of a Tanner graph and a behavior of the standard BP decoding algorithm when $N = 6$ and $M = 3$. Messages $\varepsilon_{mn}^{(i)}$ and $z_{mn}^{(i)}$ at each symbol node are updated in a parallel manner.

Step2) Perform the following hard decision step and stopping criterion test.

(Hard decision) For $1, 2, \cdots, N$, evaluate the estimated sequence $\hat{\boldsymbol{x}}^{(i)} = (\hat{x}_1^{(i)}, \hat{x}_2^{(i)}, \cdots, \hat{x}_N^{(i)})$ at iteration i, by the following equation:

$$\hat{x}_n^{(i)} := \begin{cases} 0, & \text{if } z_n^{(i)} \geq 0; \\ 1, & \text{otherwise}, \end{cases} \tag{7}$$

where

$$z_n^{(i)} := \lambda_n + \sum_{m \in \mathcal{M}(n)} \varepsilon_{mn}^{(i)}. \tag{8}$$

(Stopping criterion test) If $\hat{\boldsymbol{x}}^{(i)}H^{\mathrm{T}} = \boldsymbol{0}$ or $i = I_{\max}$, then stop the algorithm and output $\hat{\boldsymbol{x}}^{(i)}$ as an estimated sequence. Otherwise, $i := i + 1$ and go to step 1-a. □

The standard BP decoding algorithm can be regarded as a message passing algorithm on a Tanner graph of a given code [2]. The Tanner graph consists of two types of nodes called *check nodes* indexed by position of rows in H, and *symbol nodes* indexed by position of columns in H. A check node c_m and a symbol node s_n are connected with an edge if and only if $H_{mn} = 1$.

Example 1: Figure 1 shows an example of a Tanner graph of $N = 6$ and $M = 3$, where H is given by the following equation:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{9}$$

In Fig. 1, c_1, c_2, c_3 and $s_1, s_2, \cdots, s_6$ represent the check nodes and symbol nodes, respectively. $\varepsilon_{mn}^{(i)}$ and $z_{mn}^{(i)}$ are messages from a check node c_m to a symbol node s_n and from s_n to c_m at iteration i obtained by Eq. (5) and Eq. (6), respectively. The standard BP decoding algorithm updates messages for each symbol node simultaneously. We call this manner a *parallel* manner. □

3. Shuffled BP and Group Shuffled BP Decoding Algorithms

3.1 Shuffled BP Decoding Algorithm

As mentioned in Sect. 2.2, the standard BP decoding algorithm updates messages for each symbol node in a parallel manner, while the SBP decoding algorithm updates them symbol by symbol. We call this manner a *serial* manner [5], [7]. The SBP decoding algorithm is expected to converge faster than that of the standard BP decoding algorithm. As a result, the decoding performance of SBP decoding algorithm is better than the standard BP decoding algorithm at a small number of iterations. If both decoding algorithms sufficiently iterate, the decoding performance is almost the same [5].

Although in the standard BP decoding algorithm, the messages $\varepsilon_{mn}^{(i)}$ are obtained by using $z_{mn'}^{(i-1)}$ from Eqs. (4) and (5), it would be better to calculate the messages $\varepsilon_{mn}^{(i)}$ by using as many $z_{mn'}^{(i)}$ as possible in stead of $z_{mn'}^{(i-1)}$. The SBP decoding algorithm uses $z_{mn'}^{(i)}$ by processing for each symbol node in a serial manner.

[Shuffled BP Decoding Algorithm]
The initialization and the step 2 of the SBP decoding algorithm are the same as those of the standard BP decoding algorithm. The step 1 of the SBP decoding algorithm is modified as follows:

Step1') For $n = 1, 2, \cdots, N$, perform the following horizontal step and vrtical step iteratively.

(Horizontal step) For $m \in \mathcal{M}(n)$, calculate the following equations:

$$\tau_{mn}^{(i)} := \prod_{\substack{n' \in \mathcal{N}(m)\backslash n \\ n' < n}} \tanh(z_{mn'}^{(i)}/2) \times \prod_{\substack{n' \in \mathcal{N}(m)\backslash n \\ n' > n}} \tanh(z_{mn'}^{(i-1)}/2), \tag{10}$$

$$\varepsilon_{mn}^{(i)} := \log \frac{1 + \tau_{mn}^{(i)}}{1 - \tau_{mn}^{(i)}}. \tag{11}$$

(Vertical step) For $m \in \mathcal{M}(n)$, calculate the following equation:

$$z_{mn}^{(i)} := \lambda_n + \sum_{m' \in \mathcal{M}(n)\backslash m} \varepsilon_{m'n}^{(i)}. \tag{12}$$

□

3.2 Group Shuffled BP Decoding Algorithm

The SBP decoding algorithm tends to cause a large decoding delay due to updating messages in a serial manner. Moreover, it is difficult to implement the fully parallel structure of the decoder in VLSI with long LDPC codes. To solve these problems, the group SBP decoding algorithm has been devised [5], [7]. To decrease the decoding delay, the group SBP decoding algorithm divides all symbol nodes into several groups. Updating messages for symbol nodes belonging to the same group is carried out in parallel, but the processing among groups remains in serial. When it divides all symbol nodes into several groups for a long LDPC codes, a parallel structure of the decoder can be feasible for the hardware implementation.

Let G denote the number of groups, and assume that N and G satisfy $G|N$. Let N_G be the number of symbol nodes in each group, namely $N_G = \frac{N}{G}$. Although we assume $G|N$ for simplicity, this condition is not essential.

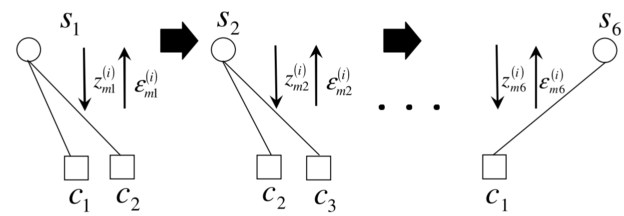

Fig. 2 An example of a behavior of the SBP decoding algorithm ($G = N$) when $N = 6$ and $M = 3$. Messages $\varepsilon_{mn}^{(i)}$ and $z_{mn}^{(i)}$ at each symbol node are updated in a serial manner.

[Group Shuffled BP Decoding]
The initialization and the step 2 of the group SBP decoding algorithm are the same as those of the standard BP decoding algorithm. The step 1 of the group SBP decoding algorithm is modified as follows:

Step1") For $g = 1, 2, \cdots, G$, perform the following horizontal step and vertical step iteratively.

(Horizontal step) For (m, n) such that $n = (g-1)N_G+1, (g-1)N_G+2, \cdots, gN_G$ and $m \in \mathcal{M}(n)$, calculate the following equations:

$$\tau_{mn}^{(i)} := \prod_{\substack{n' \in \mathcal{N}(m)\backslash n \\ n' \le (g-1)N_G}} \tanh(z_{mn'}^{(i)}/2) \times \prod_{\substack{n' \in \mathcal{N}(m)\backslash n \\ n' \ge (g-1)N_G+1}} \tanh(z_{mn'}^{(i-1)}/2), \tag{13}$$

$$\varepsilon_{mn}^{(i)} := \log \frac{1 + \tau_{mn}^{(i)}}{1 - \tau_{mn}^{(i)}}. \tag{14}$$

(Vertical step) For (m, n) such that $n = (g-1)N_G + 1, (g-1)N_G+2, \cdots, gN_G$ and $m \in \mathcal{M}(n)$, calculate the following equation:

$$z_{mn}^{(i)} := \lambda_n + \sum_{m' \in \mathcal{M}(n)\backslash m} \varepsilon_{m'n}^{(i)}. \tag{15}$$

□

When $G = 1$ and $G = N$, the group SBP decoding algorithm is reduced to the standard BP decoding and the SBP decoding algorithms, respectively.

Example 2: Figure 2 shows an updating order of messages in the SBP decoding algorithm with an example of a Tanner graph of $N = 6$ and $M = 3$ where H is given by Eq. (9). Figure 3 shows an example of an updating order of messages in the group SBP decoding algorithm with $G = 3$. In Fig. 2, $\varepsilon_{mn}^{(i)}$ and $z_{mn}^{(i)}$ are messages from a check node c_m to a symbol node s_n and from s_n to c_m at iteration i obtained by Eqs. (11) and (12), respectively.

The SBP decoding algorithm updates messages for each symbol node in a serial manner. In Fig. 3, $\varepsilon_{mn}^{(i)}$ and $z_{mn}^{(i)}$ represent messages at iteration i obtained by Eqs. (14) and (15), respectively.

The group SBP decoding algorithm updates messages belonging to the same group in a parallel manner, and

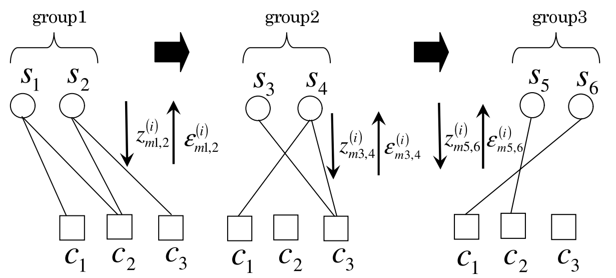

Fig. 3 An example of a behavior of the group SBP decoding algorithm (G = 3) when N = 6 and M = 3. Messages $\varepsilon^{(i)}_{mn}$ and $z^{(i)}_{mn}$ in a group are updated in a parallel manner and groups are updated in a serial manner.

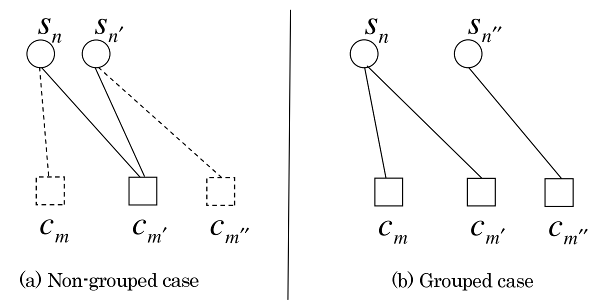

Fig. 4 (a) The case where symbol nodes s_n and $s_{n'}$ are not grouped, and (b) the case where symbol nodes s_n and $s_{n''}$ are grouped.

groups are processed in a serial manner. Although the group SBP decoding algorithm converges slower than the SBP decoding algorithm, it reduces decoding delay due to a parallel processing in respective groups. □

4. Proposed Grouping Method

4.1 Overview of the Method

In this section, we propose a concept of systematic grouping of symbol nodes, which has not been considered before.

In the group SBP decoding algorithm, N_G symbol nodes consisting of one group are automatically determined according to their positions. For example, the first group is constituted by the symbol nodes s_n, $n = 1, 2, \cdots, N_G$, the second one is constituted by the symbol nodes s_n, $n = N_G + 1, N_G + 2, \cdots, 2N_G$, and so on. We call this method the conventional grouping method. In order to reduce degradation in the performance of the group SBP decoding algorithm, we consider an effective grouping method of symbol nodes which utilizes the structure of a Tanner graph.

As mentioned in Sect. 3, it would be better to calculate the messages $\varepsilon^{(i)}_{mn}$ by using as many $z^{(i)}_{mn'}$ at iteration i as possible, in stead of using $z^{(i-1)}_{mn'}$ at iteration $i - 1$. Although the total number of messages from symbol nodes to check nodes are fixed, the numbers of messages $z^{(i-1)}_{mn'}$ and $z^{(i)}_{mn'}$ used for calculation of $\varepsilon^{(i)}_{mn}$ can be altered according to a way of grouping symbol nodes. From this reason, the proposed method divides symbol nodes into groups so that the messages $\varepsilon^{(i)}_{mn}$ are calculated by using as many $z^{(i)}_{mn'}$ as possible. This approach can be realized by making symbol nodes in the same group connect to as many check nodes as possible.

After dividing all symbol nodes into several groups by the proposed grouping method, the group SBP decoding algorithm is performed.

Example 3: An overview of the proposed grouping method is explained with a Tanner graph in Fig. 4. For example, Fig. 4(a) indicates that two symbol nodes s_n and $s_{n'}$ connecting to the same check node $c_{m'}$ are not categorized into the same group. On the other hand, Fig. 4(b) represents that two symbol nodes s_n and $s_{n''}$ not connecting to the same check node are categorized into the same group. Since the group SBP decoding algorithm updates messages for symbol nodes belonging to the same group in a parallel manner, it cannot use the message $z^{(i)}_{m'n}$ when updating the message $\varepsilon^{(i)}_{m'n'}$ in the case of Fig. 4(a). Therefore the strategy of the proposed grouping method is to categorize symbol nodes which do not connect to any common check nodes in a same group. □

4.2 Procedure of the Proposed Grouping Method

We here describe procedures of the proposed grouping method. Recall that N and G denote the code length and the number of groups, respectively. A set of all symbol positions is denoted by $\Phi = \{1, 2, \cdots, N\}$. Let $\mathcal{A}_g, g = 1, 2, \cdots, G$, denote the set of symbol positions belonging to a group g. The proposed grouping algorithm sequentially divides symbol positions into $\mathcal{A}_g$. A set of symbol positions which has not been divided into any groups at some step is denoted by $\overline{\Delta} = \Phi \setminus \bigcup_{g=1}^{G} \mathcal{A}_g$.

[Proposed Algorithm]

1. Set $\mathcal{A}_i := \emptyset$ for $i = 1, 2, \cdots, G$ and $g := 1$.

2. Find a position k, satisfying

$$k = \arg \min_{n \in \overline{\Delta}} \Big| \bigcup_{j \in \mathcal{A}_g} \mathcal{M}(j) \cap \mathcal{M}(n) \Big|, \tag{16}$$

and add k into $\mathcal{A}_g$. If there exist more than one positions satisfying Eq. (16), then choose one symbol position at random and add it into $\mathcal{A}_g$. Set $\overline{\Delta} := \overline{\Delta} \setminus \{k\}$.

3. If $g < G$, then $g := g + 1$ and go to **2.**. Otherwise, go to **4.**.

4. If $|\mathcal{A}_g| < \frac{N}{G}$, then $g := 1$ and go to **2.**. Otherwise, stop the algorithm. □

The step 2 aims at making symbol nodes in the same group connect to as many check nodes as possible. It is ideal that all of the symbol nodes in each group connect to distinct check nodes. In other words, the ideal case is that the inside term of the operation "arg" in Eq. (16) always satisfies the following equation during the execution of the algorithm:

$$\min_{n \in \overline{\Delta}} \Big| \bigcup_{j \in \mathcal{A}_g} \mathcal{M}(j) \cap \mathcal{M}(n) \Big| = 0, \quad \text{for any } g. \tag{17}$$

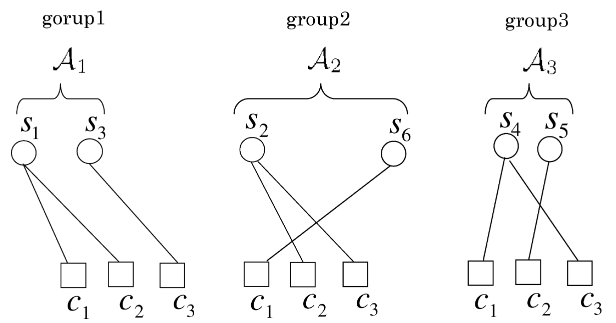

Fig. 5 An example of grouping results by the proposed grouping method ($G = 3$) for a code of $N = 6$ and $M = 3$.

Example 4: An example of grouping the symbol nodes by the proposed grouping method is presented in Fig. 5 by using the parity-check matrix given by Eq. (9). At first, we suppose that the symbol positions 2, 3 and 5 are selected at random and are divided into $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$, respectively. Then we obtain $\mathcal{A}_1 = \{3\}$, $\mathcal{A}_2 = \{2\}$, and $\mathcal{A}_3 = \{5\}$, where $\overline{\Delta} = \{1, 4, 6\}$ and set $g := 1$ at the step 4. When returning to the step 2, $n = 1$ and 6 satisfy Eq. (16). Then we randomly choose either $n = 1$ or 6. In the same way, we obtain $\mathcal{A}_1 = \{1, 3\}$, $\mathcal{A}_2 = \{2, 6\}$, and $\mathcal{A}_3 = \{4, 5\}$.

From Fig. 3, since some groups have symbol nodes connecting to the same check node, these symbol nodes cannot use the message $z_{mn}^{(i)}$ which are updated at iteration i. From Fig. 5, on the other hand, the proposed grouping method divides symbol nodes, where any symbol nodes in a group do not connect to any check nodes in common. When performing the horizontal step for the symbol nodes in $\mathcal{A}_2$, the horizontal and the vertical steps for the symbol nodes s_1 and s_3 in the set of positions $\mathcal{A}_1$ have already been performed. Therefore updating messages for the symbol nodes in the set of positions $\mathcal{A}_2$ can utilize $z_{mn}^{(i)}$. In this way, the proposed grouping method divides symbol nodes connecting to the same check nodes into different groups to use as many $z_{mn}^{(i)}$ as possible. □

Example 5: We show effectiveness of the proposed grouping method by an example with a code of $N = 12$ and $M = 6$. Figures 6(a) and (b) show a result of their Tanner graphs obtained by the conventional and the proposed grouping methods. For both methods, we set $G = 4$. Table 1 shows the number of times that the group SBP decoding algorithm of $G = 4$ uses the messages $z_{mn}^{(i-1)}$ and $z_{mn}^{(i)}$ at one iteration for both Tanner graphs. From this table, the number of times using the messages $z_{mn}^{(i)}$ for the proposed grouping method is larger than that for the conventional one. Therefore this example shows effectiveness of the proposed grouping method for the group SBP decoding algorithm. □

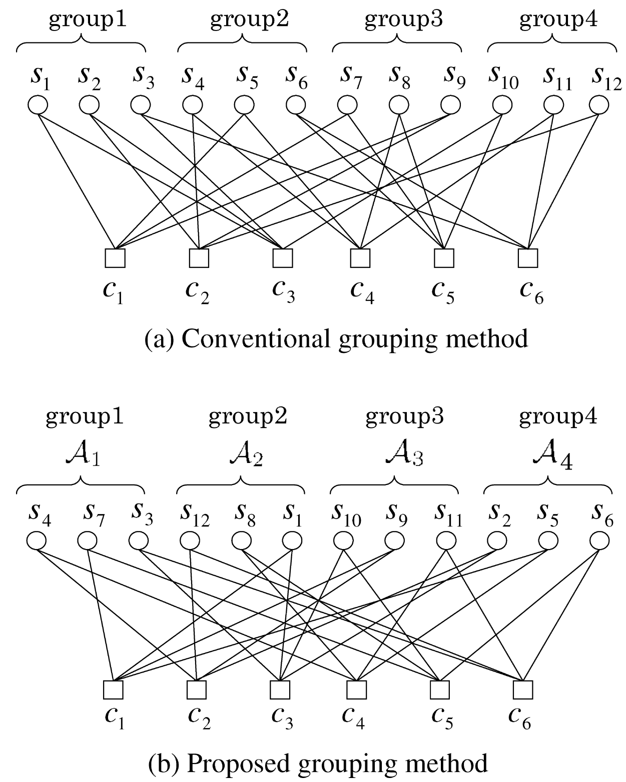

Fig. 6 (a) An example of a grouping result by the conventional grouping method for a code of $N = 12$ and $M = 6$. (b) An example of a grouping result by the proposed grouping method for the same code.

Table 1 The numbers of using $z_{mn}^{(i)}$ and $z_{mn}^{(i-1)}$ for the conventional grouping and the proposed grouping methods for a code of $N = 12$ and $M = 6$.

		group1	group2	group3	group4	total
conventional	$z_{mn}^{(i)}$	0	3	10	16	29
	$z_{mn}^{(i-1)}$	18	15	8	2	43
proposed	$z_{mn}^{(i)}$	0	6	12	18	36
	$z_{mn}^{(i-1)}$	18	12	6	0	36

5. Simulation Results and Discussions

5.1 Conditions for Simulation

In order to show effectiveness of the proposed grouping method, we show some simulation results. We compare four decoding algorithms: the standard BP decoding algorithm [1], [2], the SBP decoding algorithm [5], [7], the group SBP decoding algorithm with the conventional grouping method ($G = 2, 4, 10$) [5], [7], and the group SBP decoding algorithm with the proposed grouping method ($G = 2, 4, 10$). We use two regular LDPC codes of length $N = 4000$ and $N = 500$ with $w_r = 6$ and $w_c = 3$. We transmit at most 10^6 codewords through the AWGN channel until 100 decoding failure occurs.

We compare the decoding performance of the group

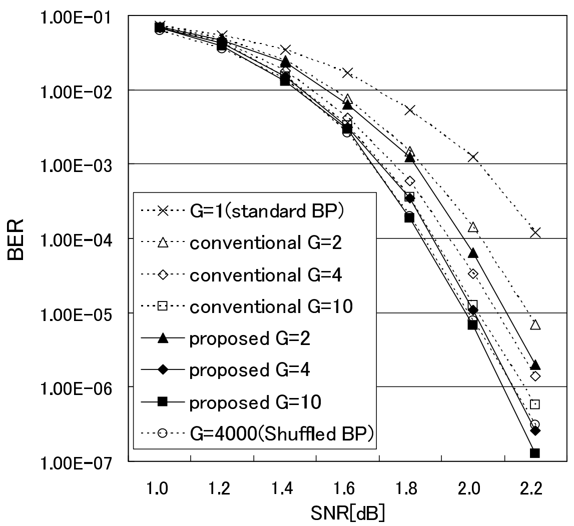

Fig. 7 BER of the code with $N = 4000$ when $I_{\max} = 10$.

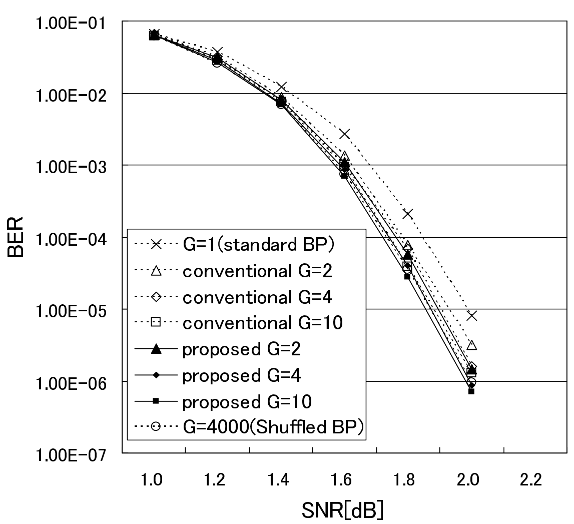

Fig. 8 BER of the code with $N = 4000$ when $I_{\max} = 20$.

SBP decoding algorithm with the conventional grouping method and that with the proposed one. We here compare (a) the decoding performance for various G, and (b) the number of iterations for each decoding algorithm as the decoding complexity. Finally (c), we discuss the number of times using $z_{mn}^{(i)}$ in Eq. (13) which are massages from check nodes at iteration i.

5.2 Simulation Results

Figures 7–9 show the bit error rate (BER) of the code of length $N = 4000$ for each decoding algorithm when $I_{\max} = 10, 20$ and 60, respectively. Figure 10 shows the BER of the code with $N = 500$ for each decoding algorithm when $I_{\max} = 5$. Figure 11 shows the BER of the code with $N = 4000$ for several values of G when $I_{\max} = 10$. Figure 12 shows the average number of iterations for each decoding algorithm of the code with $N = 4000$ when $I_{\max} = 10$.

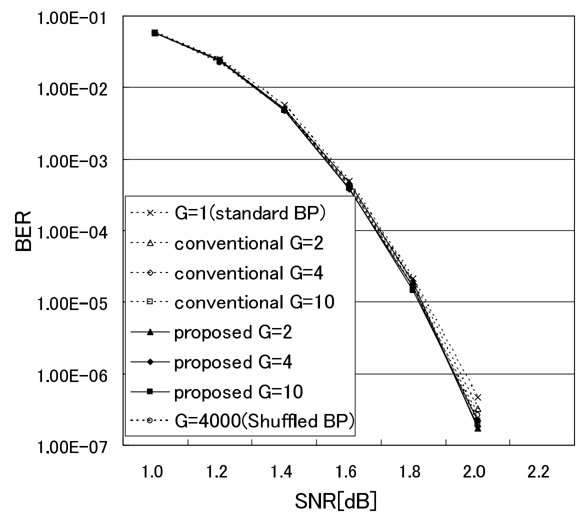

Fig. 9 BER of the code with $N = 4000$ when $I_{\max} = 60$.

Figure 13 shows the frequencies distribution on the number of iterations for each decoding algorithm when transmitting 10^4 codewords with $N = 4000$, $I_{\max} = 10$ and SNR=2.2 [dB].

In Figs. 7–12, the horizontal axis represents the signal to noise ratio (SNR) [dB]. In Figs. 7–11, the vertical axis represents the BER. In Fig. 12, the vertical axis represents the number of iterations. In Fig. 13, the horizontal axis represents the number of iterations, and the vertical axis represents the frequency on the number of iterations. In each figure, "Conventional G" and "Proposed G" denote the group SBP decoding algorithm grouped by the conventional method with the number of groups G and by the proposed method with the number of groups G, respectively.

5.3 Discussions

(a) Decoding Performance

First, from Fig. 7, the better decoding performance is obtained as the number of groups G increases. For a fixed G, decoding performance grouped by the proposed method is better than that grouped by the conventional one. It should be noted that the proposed grouping method for $G = 4$ and 10 achieves lower BERs than even for $G = 4000$ (i.e., the SBP decoding algorithm). The decoding delay becomes small when the value of G decreases. Therefore the proposed grouping method with a small number of groups not only enhances the decoding performance but also reduces the decoding delay.

Second, Figs. 8 and 9 indicate that, although the proposed grouping method attains lower BER than the conventional one for $I_{\max} = 20$ and 60, the performance difference between the proposed grouping method and the conventional one becomes smaller as $I_{\max}$ increases. Remark that when $I_{\max}$ is large and the algorithm sufficiently iterates, all decoding algorithms considered here have almost the same performance independent of the value of G [5], [7].

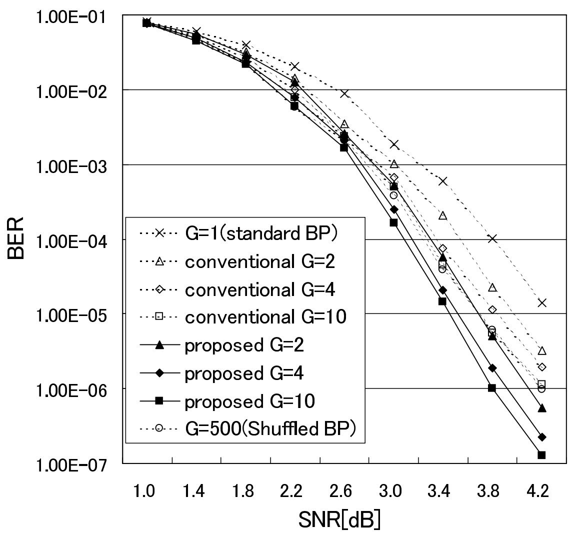

Fig. 10 BER of the code with $N = 500$ when $I_{\max} = 5$.

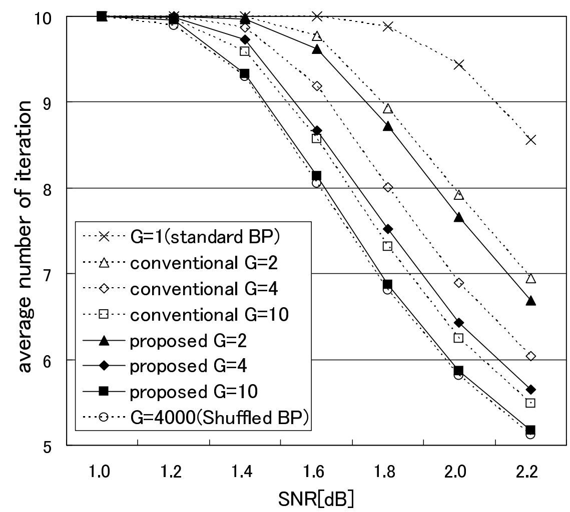

Fig. 12 Average number of iterations for the code with $N = 4000$ when $I_{\max} = 10$.

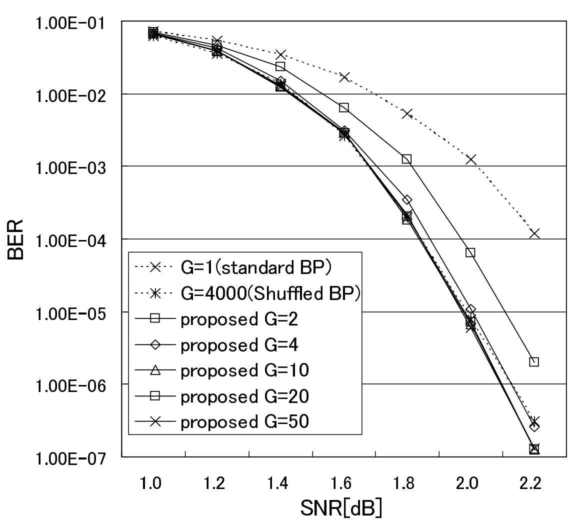

Fig. 11 BERs for several groups G for the code with $N = 4000$ when $I_{\max} = 10$.

Finally, we consider the case of using the codes with a small length. Because the convergence speed of the algorithm becomes faster when the code length is small [2], we here set $I_{\max} = 5$ to examine effects within a small number of iterations. In Fig. 10, the group SBP decoding algorithm with the proposed grouping method ($G = 2, 4, 10$) gives better performance than even for $G = 500$ (i.e., the SBP decoding algorithm) with the code of $N = 500$ when $I_{\max} = 5$. The result indicates that the proposed grouping method also has good performance for the codes with a small length.

So far, we have shown the decoding performance for $G = 2, 4, 10$. In Fig. 11, we discuss the number of groups suitable for the proposed grouping method. The BERs of $G = 20$ and 50 are almost the same. Moreover the performance difference between the proposed methods of $G = 10$ and 20 is quite small. These results indicate that the proposed grouping method of $G = 10$ has appropriate performance without causing large decoding delay.

(b) The Number of Decoding Operations

Since the number of operations per one iteration is the same for all the decoding algorithms, the smaller their average number of iterations of decoding is, the smaller the total number of operations is required. We compare the total number of operations for each decoding algorithm in Fig. 12 by showing the average number of iterations. From Fig. 12, the average number of iterations becomes small as G increases. Moreover this number for the proposed grouping method is smaller than that of the conventional one for a fixed G. This implies that the total number of operations for the proposed grouping method is smaller than that for the conventional one. Note that the gap of the average numbers of iterations between the conventional and the proposed grouping methods for a given G becomes large at high SNRs, which is often the case in practical applications.

If we set $I_{\max}$ to a large value such as $I_{\max} = 60$, all the decoding algorithms almost converge and the performance gaps become small regardless the number of groups at low SNRs. At high SNRs, on the other hand, an overall behavior for $I_{\max} = 60$ is similar to that for $I_{\max} = 10$. The reason to this phenomenon is that, at high SNRs, all of the decoding algorithms except for the conventional method of $G = 1$ tend to converge within ten iterations. This result implies that, at high SNRs, the proposed grouping method also works well even for a large value of $I_{\max}$.

In addition, from Fig. 13, the frequencies for $G = 1$ (the standard BP decoding algorithm) distributes in a wide range of the number of iterations. Comparing the frequencies of the proposed method of $G = 10$ with those of the conventional method of $G = 10$, those of the proposed method come near to smaller values of iterations. Moreover, the frequencies of the proposed method of $G = 10$ and the conventional method of $G = 4000$ (the SBP decoding algorithm) distribute almost equally. Therefore this result implies that the proposed method is more effective than the conventional

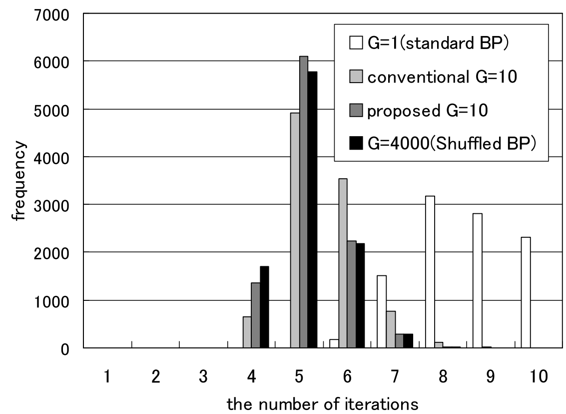

Fig. 13 Frequency on the number of iterations when transmitting 10^4 codewords with $N = 4000$, $I_{\max} = 10$ and SNR = 2.2 [dB].

one for the same number of groups and the proposed method of $G = 10$ has almost the same performance as the conventional method of $G = 4000$.

(c) Effectiveness of Grouping

We compare the number of symbol nodes that connect to the same check node in each group. The group SBP decoding algorithm cannot use many of the messages $z_{mn}^{(i)}$ at iteration i as the number of symbol nodes connecting to the same check node in each group increases. We here consider the number of symbol nodes connecting to same check nodes for the code of length $N = 4000$ in detail. When $G = 2$, the number is approximately 4030–4040 in each group for the conventional grouping method, while it is approximately 3996–3998 for the proposed one. For $G = 4$, the number is approximately 1330–1400 in each group for the conventional grouping method, while it is approximately 1000–1010 for the proposed one. For $G = 10$, the number is approximately 250–290 in each group for the conventional grouping method, while it is approximately 10–20 for the proposed one. From these results, it is better to make the value of G large, and $G = 10$ gives appropriate performance for this code.

6. Conclusion

In this paper, we have proposed an effective grouping method of symbol nodes for the group SBP decoding algorithm. From simulation results, it has been shown that performance of the group SBP decoding algorithm with the proposed grouping method is better than that with the conventional one for given G. The performance of the group SBP decoding algorithm with the proposed grouping method for $G = 4$ and 10 is even better than that of the SBP decoding algorithm. Since the decoding delay is reduced for small values of G, the proposed grouping method can decrease the decoding delay and it has higher performance than the SBP decoding algorithm. Moreover it enables to reduce the total number of decoding operations.

For further works, we need to consider a grouping method that is the optimal in the decoding performance. We should also consider a method for non-uniform number of symbol nodes in each group. There have been proposed other types of the SBP decoding which process each check node in a serial manner [8]. It is possible to apply a similar method of this paper to this type of the SBP decoding algorithm, and thorough investigation in such a case is also considered as a future work. Moreover, it is also further works to consider processing each symbol node in serial for other types of decoding algorithm such as the Min-Sum decoding algorithm.

Acknowledgments

The authors would like to thank Associate Editor, Prof. M. Isaka, and anonymous reviewers for their valuable comments. One of the authors, Y. Sato wishes to thank Dr. M. Kobayashi at Shonan Institute of Technology, Dr. T. Niinomi at Musashi Institute of Technology and Dr. T. Ishida at Waseda University for their fruitful suggestions.

References

[1] R.G. Gallager, Low density parity check codes, MIT Press, 1963.
[2] D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. Inf. Theory, vol.45, no.2, pp.399–431, March 1999.
[3] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," IEEE Trans. Inf. Theory, vol.47, no.2, pp.599–618, Feb. 2001.
[4] T. Richardson 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.
[5] J. Zhang and M.P.C. Fossorier, "Shuffled iterative decoding," IEEE Trans. Commun., vol.53, no.2, pp.209–213, Feb. 2005.
[6] Z. Chuan-Gang, Y. Jin-Sheng, L. Xue-Hong, and L. Jia-Ru, "Improvement of shuffled iterative decoding," Proc. 2006 IEEE Inf. Theory Workshop, pp.114–116, Chengdu, China, Oct. 2006.
[7] J. Zhang, Y. Wang, M.P.C. Fossorier, and J.S. Yedidia, "Iterative decoding with replica," IEEE Trans. Inf. Theory, vol.53, no.5, pp.1644–1663, May 2007.
[8] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," IEEE Trans. Inf. Theory, vol.53, no.11, pp.4076–4091, Nov. 2007.

Yoshiyuki Sato was born in Fukuoka, Japan, on April 15, 1983. He received the B.E. degree in Industrial and Management Systems Engineering from Waseda University, Tokyo, Japan, in 2007. He is currently a master's student in Industrial and Management Systems Engineering at the Graduate School of Waseda University. His research interests are coding theory and information theory.

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

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 at the University of Electro-Communications, Tokyo, Japan. His research interests are coding theory and information security. He is a member of the Society of Information Theory and its Applications 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. Since 1981, he has been a professor of the School of Science and Engineering, Waseda University, Tokyo, Japan. 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, the Information Processing Society of Japan, and the Japan Industrial Management Association.