PAPER *Special Section on Information Theory and Its Applications*

A Modification Method for Constructing Low-Density Parity-Check Codes for Burst Erasures

Gou HOSOYA[†a)], ***Student Member***, **Hideki YAGI**[††], **Toshiyasu MATSUSHIMA**[†], ***Members***, ***and*** **Shigeichi HIRASAWA**[†], ***Fellow***

SUMMARY We study a modification method for constructing low-density parity-check (LDPC) codes for solid burst erasures. Our proposed modification method is based on a column permutation technique for a parity-check matrix of the original LDPC codes. It can change the burst erasure correction capabilities without degradation in the performance over random erasure channels. We show by simulation results that the performance of codes permuted by our method are better than that of the original codes, especially with two or more solid burst erasures.

key words: *low-density parity-check (LDPC) code, burst erasure, sum-product decoding, stopping set*

1. Introduction

The combination of LDPC codes with the sum-product (SP) decoding algorithm can achieve high performance with low decoding complexity [1]. Most of studies of LDPC codes assume random errors or random erasures. When we consider situations using LDPC codes more practically, we must take into account correction capabilities of not only random errors or erasures but also burst ones. In order to adapt the code to burst channels, two approaches have been considered. The first approach is to improve decoding methods for the burst channels, and the second one is to construct or modify the codes suitable for the burst channels. The first approach has been taken by J. Garcia-Frias [9] and A.W. Eckford et al. [10]. The second has been taken by the present authors [6], M. Yang and W.E. Ryan [7], and T. Wadayama [8]. The present authors have proposed a column permutation algorithm for a parity-check matrix of LDPC codes over burst error channels without degradation in decoding performance over random error channels. Yang et al. have proposed on $L_{\max}$ algorithm which can evaluate a reasonable maximum burst erasure length for a given parity-check matrix of LDPC codes by an exhaustive search method. Wadayama has also proposed a column permutation algorithm which can increase $L_{\max}$ for given LDPC codes. The works by Yang et al. and by Wadayama have considered only one burst erasure[*]. When we consider the case of two or more solid burst erasures, we need to devise other methods.

In this paper, we propose a new modification method for constructing LDPC codes for burst erasure channels of two or more burst erasures. The modification method is also based on a column permutation technique for a parity-check matrix of the LDPC codes. Our proposed method permutes the columns based on the distance between elements (DBE), which is defined as the number of symbol positions between adjacent elements 1 at each row of a parity-check matrix of the code, and can change the burst erasure correction capabilities maintaining the original performance for random erasure correction. We show by simulation results that the performance of codes permuted by our method are better than those of the original codes and the codes obtained by Wadayama's method when two or more solid burst erasures have been occurred.

This paper is organized as follows. In Sect. 2, we describe LDPC codes and the SP decoding algorithm. In Sect. 3.1, we describe a correction capability of the LDPC codes for one burst erasure. A column permutation algorithm based on increasing correction capability for one burst erasure is presented in Sect. 3.2 and we propose a column permutation algorithm based on increasing DBE in Sect. 3.3. Finally, some simulation results are presented in Sect. 4 and concluding remarks are given in Sect. 5.

2. LDPC Codes and a Decoding Algorithm

2.1 LDPC Codes

We assume a codeword of the LDPC code $\boldsymbol{c} = (c_1, c_2, \ldots, c_N) \in \{0, 1\}^N$ of length N is transmitted through an erasure channel. $\boldsymbol{c}$ is disturbed by the sequence from the channel $\boldsymbol{e} = (e_1, e_2, \ldots, e_N) \in \{0, \epsilon\}^N$ where ϵ denotes an erasure, and the decoder receives a sequence $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$. The addition of a binary symbol and the erasure symbol are defined as $0 + \epsilon = \epsilon$ and $1 + \epsilon = \epsilon$. The decoder estimates the transmitted codeword from the received sequence.

Let $H = [H_{mn}]$, $m \in [1, M]$, $n \in [1, N]$, $\boldsymbol{c}H^{\mathrm{T}} = \mathbf{0}$, be a parity-check matrix whose row and column lengths are M and N, respectively. In this paper, we consider binary regular LDPC codes for simplify the discussion. Let w_r and w_c

Manuscript received January 24, 2006.
Manuscript revised April 14, 2006.
Final manuscript received June 6, 2006.

[†]The authors are with the Department of Industrial and Management Systems Engineering, School of Science and Engineering, Waseda University, Tokyo, 169-8555 Japan.
[††]The author is with the Media Network Center, Waseda University, Tokyo, 169-8050 Japan.
a) E-mail: hosoya@hirasa.mgmt.waseda.ac.jp

DOI: 10.1093/ietfec/e89–a.10.2501

[*]A "solid burst erasure" of length t stands for consecutive t erasures.

be row and column weight of H, respectively. The number of rows M is given by $M = Nw_c/w_r$ and the designed rate of the (N, w_r, w_c) LDPC codes is $R' = 1 - \frac{M}{N}$. The rate of the codes R satisfies $R \leq R'$ since H is not guaranteed to be a full rank matrix.

A parity-check matrix of a LDPC code is constructed from a seed of random generator. We use a construction method based on the guidelines given by D.J.C. MacKay [2].

2.2 The SP Decoding Algorithm

The SP decoding algorithm on the binary erasure channel fails in decoding when a subset of erased symbols have a *stopping set*.

Definition 1: [Stopping set [5]] Choose some columns of H to make a submatrix. A stopping set is a subset of symbol positions such that weights of all rows of this submatrix are at lease two. A union of stopping sets is also a stopping set, so any submatrix of a parity-check matrix has a unique maximal stopping set. □

3. Modification Method of LDPC Codes

3.1 Reasonable Maximum Burst Erasure Length

In [7], a measure of burst erasure correction capability denoted by $L_{\max}$ has been presented.

Definition 2: [$L_{\max}$ [7]] $L_{\max}$ is a *reasonable maximum burst erasure length* which can be decoded by the SP decoding algorithm when only one burst erasure with length equal to or smaller than $L_{\max}$ occurres. □

Note that $L_{\max}$ is a correctable capability for only one solid burst erasure. Yang et al. have proposed the $L_{\max}$ algorithm which can evaluate a burst erasure correctable length for a given parity-check matrix of LDPC codes by an exhaustive search. See [7] for details.

3.2 Column Permutation

A column permutation of a parity-check matrix will easily change the performance of the code for burst erasure channels. While column permuted parity-check matrices are equivalent for memoryless channels in the sense that parameters of the codes, such as the code length or the weight distribution of the codewords, are equal [6]. In addition, since a distribution of lengths of loops in the bipartite graph is not changed by column permutation, the performance on the random erasure channels decoded by the SP decoding algorithm would also be unchanged.

Remark 1: Random erasure correction capability by the SP decoding algorithm is invariant by column permutation to a parity-check matrix of the code. □

Wadayama has proposed a column permutation algorithm which permutes columns of a parity-check matrix to increase $L_{\max}$. When some column prevents a large $L_{\max}$, elimination of bad columns is performed [8]. The code given by this method is the almost optimal in performance for correcting one solid burst erasure. However, it causes change in the random erasure correction capability in general by deleting some columns of a parity-check matrix.

3.3 Proposed Column Permutation Method

The optimal column permutation method is to minimize the probability of decoding error for a given LDPC code over burst erasure channels. It requires an exhaustive search since it generates $N!$ patterns of column permuted parity-check matrices and needs to evaluate its probability of decoding error for each of permutation patterns.

We propose a column permutation algorithm based on a different measure from $L_{\max}$. The reasons are that (1) a column permutation algorithm based on the value of $L_{\max}$ needs much time, since it requires to perform the SP decoding algorithm for evaluating correctable capabilities of all burst erasure patterns of length equal to or less than $L_{\max}$, and (2) as mentioned in Sect. 3.1, that increasing the value of $L_{\max}$ is not enough to correct two or more burst erasures.

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

$$\mathcal{A}(m) \triangleq \{n \mid H_{mn} = 1\} = \{n_{m,1}, n_{m,2}, \dots, n_{m,w_r}\},$$
$$\mathcal{B}(n) \triangleq \{m \mid H_{mn} = 1\} = \{m_{n,1}, m_{n,2}, \dots, m_{n,w_c}\},$$

where $n_{m,1} < n_{m,2} < \dots < n_{m,w_r}$ and $m_{n,1} < m_{n,2} < \dots < m_{n,w_c}$, respectively. We define the *distance between elements* (DBE) as the number of symbol positions between adjacent elements 1 at each row of the parity-check matrix.

Definition 3: [DBE] The DBEs $d_{m\gamma}$, $m \in [1, M]$, $\gamma \in [1, w_r - 1]$, and the minimum value of DBEs $D_{\min}$ are defined by the following equations, respectively:

$$d_{m\gamma} \triangleq n_{m,\gamma+1} - n_{m,\gamma}, \quad (1)$$
$$D_{\min} \triangleq \min_{m,\gamma}\{d_{m\gamma}\}. \quad (2)$$

□

For example, assume that the row of the parity-check matrix has the form (100001001). The elements 1 at this row are in positions 1, 6, and 9, so DBEs are $d_{11} = 5$ and $d_{12} = 3$.

Remark 2: $D_{\min}$ is the maximum value of a burst erasure correctable length by the SP decoding algorithm at the first iteration [7]. □

From Definition 1, the weight of each row of a submatrix that consists of a stopping set is at least two. Therefore, in order to avoid two or more nonzero columns in one row being contained in a stopping set of burst erasures, it is better to make the minimum value of DBEs large.

For a parity-check matrix, the DBEs are changed by column permutation. Let D_{ave} be an arithmetic average value of DBEs defined as

$$D_{\text{ave}} \triangleq \frac{1}{M(w_r-1)}\sum_{m=1}^{M}\sum_{\gamma=1}^{w_r-1} d_{m\gamma}. \tag{3}$$

To increase DBEs, we consider the following two conditions: (i) D_{ave} has a large value, (ii) $D_{\min} > \delta$ where δ, is some positive constant. From Eqs. (1) and (3)

$$\sum_{\gamma=1}^{w_r-1} d_{m\gamma} = n_{m,w_r} - n_{m,1}. \tag{4}$$

We can easily see that D_{ave} depends on a difference of column positions between the leftmost element 1 and the rightmost element 1 at each row of a parity-check matrix. Therefore the Condition (i) implies that the sum of DBEs has a large value.

We now let the leftmost (rightmost) symbol position at each row of H be small (large) value as possible. Let $r' \triangleq M \bmod w_c$ and $\rho' \triangleq \frac{M-r'}{w_c} = \frac{N}{w_r} - \frac{r'}{w_c}$. Assume that H has a following form:

$$H \triangleq [H_{\text{lef}}, H_{\text{mid}}, H_{\text{rig}}]. \tag{5}$$

H_{lef} and H_{rig} are $M \times \rho'$ matrices such that the weights of r' rows and $M - r'$ rows of these matrices are 0 and 1, respectively, and the weights of columns of those are w_c. Fig. 1 shows an example of H that has the form of Eq. (5). H_{lef} constitutes the leftmost ρ' columns of H. Assume that column positions of element 1 at those columns are $n_{m,1}$, the leftmost element 1 at each row. Similarly, H_{rig} constitutes the rightmost ρ' columns of H. Assume again that column positions of element 1 at those columns are n_{m,w_r}, the rightmost element 1 at each row. Then the following lemma holds.

Lemma 1: The parity-check matrix H that has the form of Eq. (5) maximizes the right-hand side of Eq. (3). □

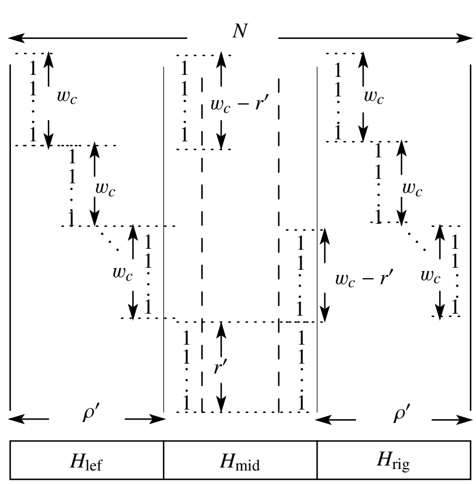

Fig. 1 An example of H that has a form of Eq. (5).

The following theorem on the relation of D_{ave} and (N, w_r, w_c) LDPC codes holds.

Theorem 1: The parity-check matrix of (N, w_r, w_c) LDPC codes satisfies the following equation:

$$D_{\text{ave}} \le \rho \tag{6}$$

where $\rho \triangleq \frac{N}{w_r}$. □

Proof: See Appendix. □

Note that the average value of DBEs D_{ave} is an upper bound of the minimum value of DBEs $D_{\min}$. To modify the parity-check matrix of LDPC codes suitable for burst erasures, we permute the columns of parity-check matrix of LDPC codes to have the form Eq. (5) as nearly same as possilble.

Definition 4: Let $\tilde{H}$ be a parity-check matrix of the LDPC codes which is generated from $H = [\boldsymbol{h}_1^{\mathrm{T}}, \boldsymbol{h}_2^{\mathrm{T}}, \ldots, \boldsymbol{h}_N^{\mathrm{T}}]$ by a column permutation. $\boldsymbol{h}_n^{\mathrm{T}}$, $n = 1, 2, \cdots, N$, denotes the column vector at a position n.

$$\tilde{H} = [\tilde{\boldsymbol{h}}_1^{\mathrm{T}}, \tilde{\boldsymbol{h}}_2^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{h}}_N^{\mathrm{T}}] \triangleq [\boldsymbol{h}_{\theta(1)}^{\mathrm{T}}, \boldsymbol{h}_{\theta(2)}^{\mathrm{T}}, \ldots, \boldsymbol{h}_{\theta(N)}^{\mathrm{T}}].$$

where $\theta(n)$ denotes a permutation function. We define $\tilde{\mathcal{A}}(m)$, $m = 1, 2, \cdots, M$, and $\tilde{\mathcal{B}}(n)$, $n = 1, 2, \cdots, N$, as follows:

$$\tilde{\mathcal{A}}(m) \triangleq \{n \mid \tilde{H}_{mn} = 1\},\ \tilde{\mathcal{B}}(n) \triangleq \{m \mid \tilde{H}_{mn} = 1\}.$$

□

Next, we present the proposed column permutation algorithm to obtain a parity-check matrix $\tilde{H}$ of the form Eq. (5). The algorithm produces a new parity-check matrix $\tilde{H} \triangleq [\tilde{H}_{\text{lef}}, \tilde{H}_{\text{mid}}, \tilde{H}_{\text{rig}}]$ from original one H by column permutation. The algorithm constitutes four steps (step (A)–(D)). The overview of the algorithm is as follow:

[Overview of the Algorithm]

At the step (A), we choose the columns of $\tilde{H}_{\text{lef}}$ from the columns of H and we fix the column position of $\tilde{H}_{\text{lef}}$. At the step (B), we choose the columns of $\tilde{H}_{\text{rig}}$ from the columns of H. At the step (C), we choose the columns of $\tilde{H}_{\text{mid}}$ from the columns of H with restrictions to have DBEs large. At the step (D), we swap columns of $\tilde{H}_{\text{rig}}$ with restrictions to have DBEs large. □

[Proposed Column Permutation Algorithm]

Step (A) : Choosing columns of $\tilde{H}_{\text{lef}}$

(A1) Set $i := 1$ and set counters at all the symbol positions to 0.

(A2) Choose a column of H at a symbol position n which is not chosen before and whose counter is 0. If we choose a column, then set $\theta(i) = n$, set the counter at position n to 1, $i := i + 1$, and go to (A3). If there are no columns whose counter is 0, stop the step (A).

(A3) For any $m \in \mathcal{B}(n)$, set the counter at symbol positions $n' \in \mathcal{A}(m)$ to 1 and go to (A2).

Step (B) : Choosing columns of $\tilde{H}_{\rm rig}$

(B1) Set $j := 1$ and set counters at all the symbol positions to 0.

(B2) Choose a column of H at a symbol position n which is not chosen before (which includes at the step (A)) and whose counter is 0. If we choose a column, then set $\theta(N - j + 1) = n$ and set the counter at the position n to 1, $j := j + 1$, and go to (B3). If there are no columns whose counter is 0, stop the step (B).

(B3) For any $m \in \mathcal{B}(n)$, set the counter at symbol positions $n' \in \mathcal{A}(m)$ to 1 and go to (B2).

Step (C) : Choosing columns of $\tilde{H}_{\rm mid}$

(C1) Set $k := i$ and set counters at all the symbol positions to 0. Set δ for some value that satisfies $\rho \geq \delta$. For $t = k - \delta + 1, k - \delta + 2, \ldots, k - 1$, set the counter at symbol positions $n' \in \mathcal{A}(m)$, $m \in \tilde{\mathcal{B}}(t)$, to 1.

(C2) Choose a column of H at a symbol position n which is not chosen before (which includes at the steps (A) and (B)) and whose counter is 0. If we choose a column and $k < N - j + 1$, then set $\theta(k) = n$ and set the counter at the position n to 1 and go to (C3). If we choose a column and $k = N - j + 1$, then stop the step (C). If there are no columns whose counter is 0, stop the algorithm (in this case, the algorithm fails).

(C3) Set counters at all the symbol positions to 0. For $t = k-\delta+2, k-\delta+3, \ldots, k$, set the counter at symbol positions $n' \in \mathcal{A}(m)$, $m \in \tilde{\mathcal{B}}(t)$, to 1, $k := k + 1$, and go to (C2).

Step (D) : Permuting columns of $\tilde{H}_{\rm rig}$

(D1) Set $z := N - j + 2$ and set counters at all the symbol positions to 0. For $t = z - \delta + 1, z - \delta + 2, \ldots, z - 1$, set the counter at symbol positions $n \in \tilde{\mathcal{A}}(m)$, $m \in \tilde{\mathcal{B}}(t)$, to 1.

(D2) If the counter at a symbol position z equals 0, then a column at a symbol position z is reasonable and go to (D3). If there are no columns whose counter is 0, stop the algorithm (in this case, the algorithm fails). Otherwise, go to (D4).

(D3) If $z = N$, then the algorithm succeeds. Otherwise, set counters at all the symbol positions to 0. For $t = z - \delta + 2, z - \delta + 3, \ldots, z$, set the counter at symbol positions $n' \in \tilde{\mathcal{A}}(m)$, $m \in \tilde{\mathcal{B}}(t)$, to 1, $z := z + 1$ and go to (D2).

(D4) Swap a column at a symbol position z with a column at a symbol position z', $z' > z$ and its counter is 0. i.e., $q = \theta(z')$, $\theta(z') = \theta(z)$, $\theta(z) = q$ where q is some value. Go to (D3). □

4. Simulation Results and Discussion

In order to demonstrate decoding performance of codes obtained by our proposed modification method, we show some simulation results.

Table 1 The parameters of codes.

	N	w_r	w_c	R'
Code A	500	6	3	0.5
Code B	2000	6	3	0.5
Code C	500	10	3	0.7

4.1 Conditions for Simulations

We use LDPC codes that have three different parameters (denoted by "Code A," "Code B," and "Code C") in simulations. The parameters of these codes are shown in Table 1. We construct three codes (denoted by "Code 1," "Code 2," and "Code 3") by different seeds of a random generator (these original codes are denoted by "Original") for each code with different parameters. For each code, we permute columns of a parity-check matrix of the original codes. The permutation methods are 1) based on DBEs (denoted by "DBE"), 2) based on increasing $L_{\max}$ (denoted by "$L_{\max}$")[†], and 3) to have values of many DBEs are 1 (denoted by "Small"). We decode until at least 1×10^7 codewords are transmitted or 100 codewords have failed to decode by the SP decoding algorithm.

4.2 The Values of $L_{\max}$ and DBEs

We show the values $L_{\max}$, $D_{\rm ave}$, and $D_{\min}$ for Code A–C in Tables 2–7, respectively.

From Tables 2, 4, and 6, the values of $L_{\max}$ of the code "DBE" and the code "$L_{\max}$" are increased by column permutation from the code "Original"[††]. Note that the code "$L_{\max}$" is the optimal one since it is constructed to have a large value of $L_{\max}$. The value of $L_{\max}$ of the code "DBE" is close to that of the code "$L_{\max}$," compared with the other codes. The code "Small" has the smallest values of $L_{\max}$. So a column permutation having many DBEs 1s produces bad codes for one burst erasure.

From Tables 3, 5, and 7, the values of $D_{\rm ave}$ of the code "DBE" are the largest. Note that an upper bound of $D_{\rm ave}$ of the Codes A, B, and C are $83.33\cdots$, $333.33\cdots$, and 50, respectively. The values of $D_{\rm ave}$ of the codes "Original" and "$L_{\max}$" are almost same, although the values $L_{\max}$ of those codes differ greatly, as in Tables 3, 5, and 7. The code "DBE" is constructed to have a large value of DBEs; values of $D_{\min}$ of the code "DBE" are 53 for Code A[†††], and those of the other codes are all 1.

[†]Note that this column permutation method may delete some columns for increasing $L_{\max}$. However, we do not delete any columns for a fair comparison.

[††]As for the Code A where $L_{\max} \geq 202$ of a column permutation algorithm based on increasing $L_{\max}$ in Sect. 3.2, we could not succeed via a column permutation in constructing the code "$L_{\max}$" from the code "Original" without deleting any columns.

[†††]As for the Code A where $\delta \geq 54$ by the our proposed algorithm in Sect. 3.3, we could not succeed via a column permutation in constructing the code "DBE" from the code "Original."

Table 2 The values of $L_{\max}$ of the Code A.

	Original	DBE	$L_{\max}$	Small
Code 1	163	190	201	137
Code 2	157	192	201	137
Code 3	147	198	201	129

Table 3 The values of D_{ave} ($D_{\min}$) of the Code A.

	Original	DBE	$L_{\max}$	Small
Code 1	62.42 (1)	82.41 (53)	63.47 (1)	54.14 (1)
Code 2	62.83 (1)	82.35 (53)	65.12 (1)	53.99 (1)
Code 3	61.20 (1)	82.33 (53)	62.47 (1)	53.46 (1)

Table 4 The values of $L_{\max}$ of the Code B.

	Original	DBE	$L_{\max}$	Small
Code1	607	771	786	543
Code2	604	766	786	537
Code3	600	769	786	537

Table 5 The values of D_{ave} ($D_{\min}$) of the Code B.

	Original	DBE	$L_{\max}$	Small
Code1	241.12 (1)	330.62 (193)	244.02 (1)	215.21 (1)
Code2	241.84 (1)	330.30 (193)	246.23 (1)	212.80 (1)
Code3	241.33 (1)	330.22 (193)	246.78 (1)	214.83 (1)

Table 6 The values of $L_{\max}$ of the Code C.

	Original	DBE	$L_{\max}$	Small
Code1	90	110	121	60
Code2	89	109	120	53
Code3	91	111	120	60

Table 7 The values of D_{ave} ($D_{\min}$) of the Code C.

	Original	DBE	$L_{\max}$	Small
Code1	42.99 (1)	49.74 (32)	44.01 (1)	36.28 (1)
Code2	42.29 (1)	49.79 (32)	44.25 (1)	37.08 (1)
Code3	41.88 (1)	49.77 (32)	44.15 (1)	36.79 (1)

4.3 Decoding for Several Burst Erasures

4.3.1 Decoding Peformance for L Solid Burst Erasures

We denote the number of solid burst erasures by L and the total length of erasures by T. We assume L solid burst erasures have occurred by fixing $T = 190$, 200, and 210. Lengths for each burst are chosen from a seed of random generator. Note that the total length of burst erasures of $T = 190$ and 200 is equal to or smaller than the values of $L_{\max}$ of the codes "$L_{\max}$" and is greater than those of the other codes (except "DBE" of the code 3). Figures 2–4 show decoding performance for total burst length $T = 190$, 200, and 210, respectively. The horizontal axis shows the number of solid burst erasures L, and the vertical axis shows the word error rate (WER).

From these figures, the performance of the code "DBE" is better than that of the other codes when $L \geq 2$. In Fig. 2, WER of the code "DBE" is approximately 10^5 times smaller

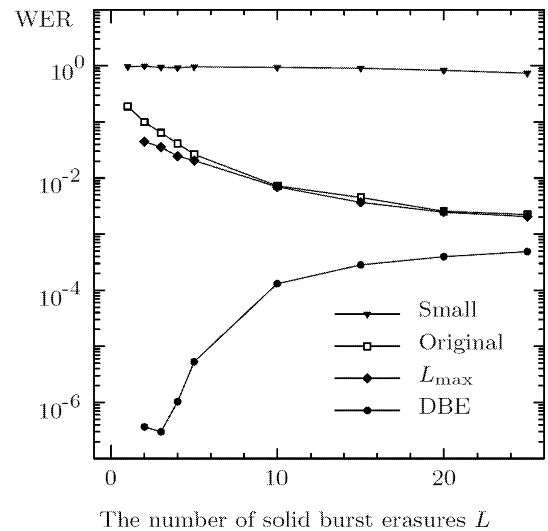

Fig. 2 Decoding performance for L solid burst erasures whose sum T is 190 of the Code A. Note that when $L = 1$, the code "$L_{\max}$" produces zero WER, since the lengths of a solid burst erasure 190 and 200 are smaller than its $L_{\max}$ (≥ 201) from Table 2.

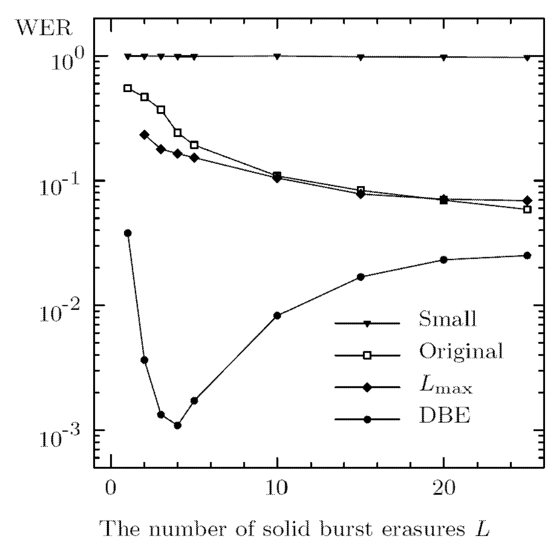

Fig. 3 Decoding performance for L solid burst erasures whose sum T is 200 of the Code A. Note that when $L = 1$, the code "DBE" also produces zero WER, since the length of solid burst erasure 190 is smaller than or equal to its $L_{\max}$ (≥ 190) of the code "DBE" from Table 2.

than that of the code "$L_{\max}$" when $L = 3$. The difference of the performance between the code "DBE" and the other codes is large when L is small (but it is not the case when $L = 1$) and becomes smaller as L tends to have a large value. The performance of the code "Small" is the worst for all values of L.

Note 1: When $L = 1$ in Figs. 2 and 3, the code "$L_{\max}$" produces zero WER, since the lengths of a solid burst erasure 190 and 200 are smaller than its $L_{\max}$ (≥ 201) from Table 2. When $L = 1$ in Fig. 2, the code "DBE" also produces zero WER, since the length of solid burst erasure 190 is smaller than or equal to its $L_{\max}$ (≥ 190) of the code "DBE." □

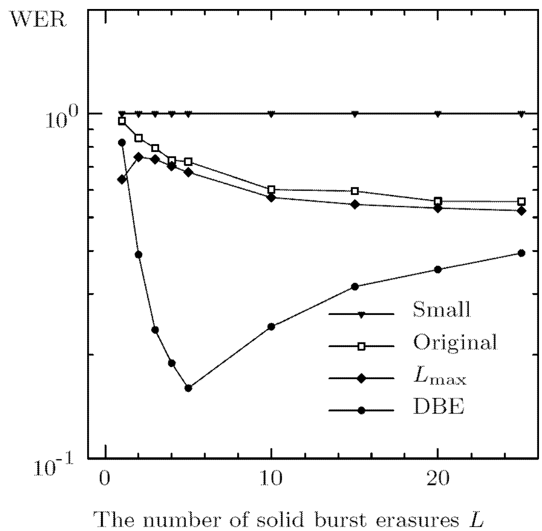

Fig. 4 Decoding performance for L solid burst erasures whose sum T is 210.

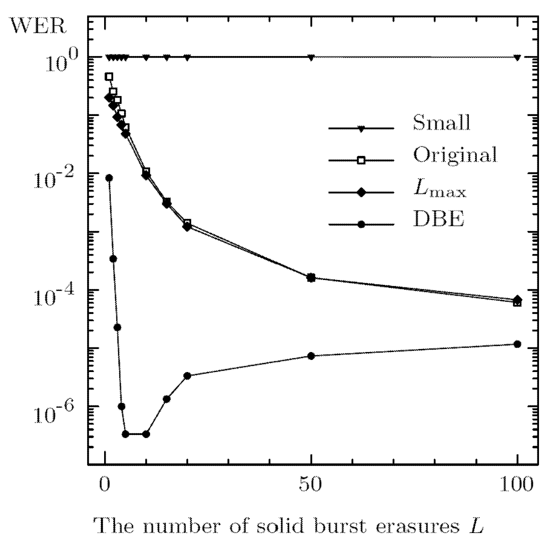

Fig. 5 Decoding performance for L solid burst erasures whose sum T = 800 of the Code B.

From Figs. 3 and 4, although the difference between the performance of the code "DBE" and the other codes is smaller than the one in Fig. 2, the code "DBE" is better than the other codes. The case of $T = 210$ equals to the binary erasure channel with erasure probability 0.41, and the iterative threshold of erasure probability of LDPC codes, with $w_r = 6$ and $w_c = 3$, decoded by the SP decoding algorithm is approximately 0.429 [4], so all of the codes in Fig. 4 cannot correct many erasures. Nevertheless, WER of the code "DBE" is approximately 7 times smaller than that of "$L_{\max}$" when $L = 5$. From Figs. 2–4, the performance of the code "DBE" seems better as the value of L becomes smallrt when T is a fixed value.

We show decoding performance for Code B and Code C in Figs. 5 and 6, respectively. Figure 5 shows $T = 800$ for the Code B and Fig. 6 shows $T = 110$ for the Code C.

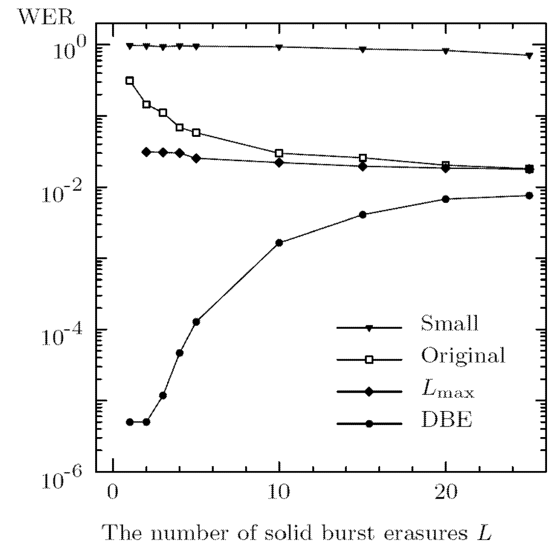

Fig. 6 Decoding performance for L solid burst erasures whose sum $T = 110$ of the Code C. Note that when $L = 1$ in Fig. 6, the code "$L_{\max}$" produces zero WER, since the length of a burst erasure 110 is smaller than the values $L_{\max}$ (≥ 120) of the code "$L_{\max}$" from Table 6.

From these figures, the performance of the code "DBE" is better than the other code even when the codes have different parameters.

Note 2: Note that when $L = 1$ in Fig. 6, the code "$L_{\max}$" produces zero WER, since the length of a burst erasure 110 is smaller than the values $L_{\max}$ (≥ 120) of the code "$L_{\max}$" from Table 6. □

4.3.2 Decoding Performance for Solid Burst Erasures of Length T

We assume that L = 5, 10, and 25 solid burst erasures for a fixed sum of length T have occurred. Lengths for each burst are chosen from a seed of a random generator. Figures 7–9 show decoding performance for the number of solid bursts L = 5, 10, and 25, respectively. The horizontal axis shows the total length of solid bursts T.

From these figures, the performance of the code "DBE" is better than that of the other codes. The difference of the performance between the code "DBE" and the other codes is large when T is small and becomes smaller as T tends to have a large value. BER of the code "DBE" is approximately 10^4 times smaller than that of the code "$L_{\max}$" when $T = 185$ in Fig. 7.

We show decoding performance for the Code B and the Code C at $L = 5$ in Figs. 10 and 11, respectively. Figures 10 and 11 show the decoding results of $L = 5$ for the Code B and the Code C, respectively. From these figures, the performance of the code "DBE" is better than those of the other code. From Figs. 7 and 10, the behavior of performance that of the Code A and that of the Code B under the condition that the designed rate R' is constant and for different code length N, are almost the same. From Figs. 7 and 11, the difference of the performance of the code "DBE" and those of

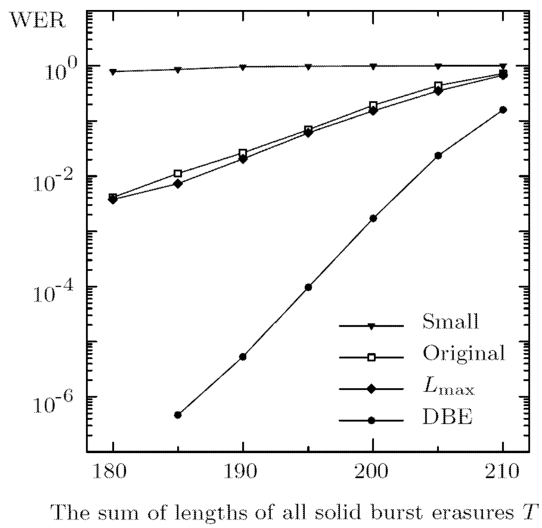

Fig. 7 Decoding performance for $L = 5$ solid burst erasures whose sum is T of the Code A.

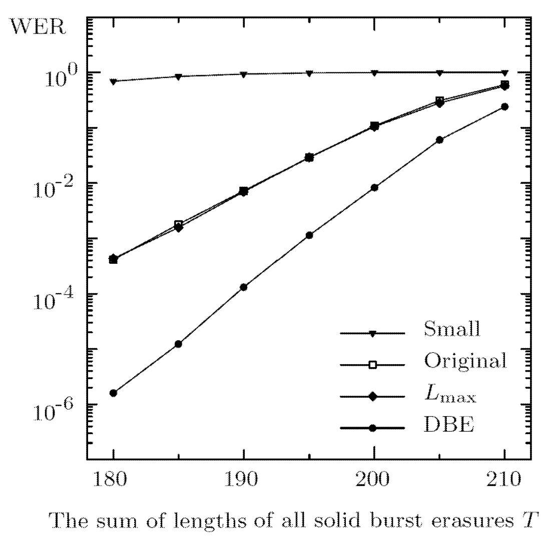

Fig. 8 Decoding performance for $L = 10$ solid burst erasures whose sum is T of the Code A.

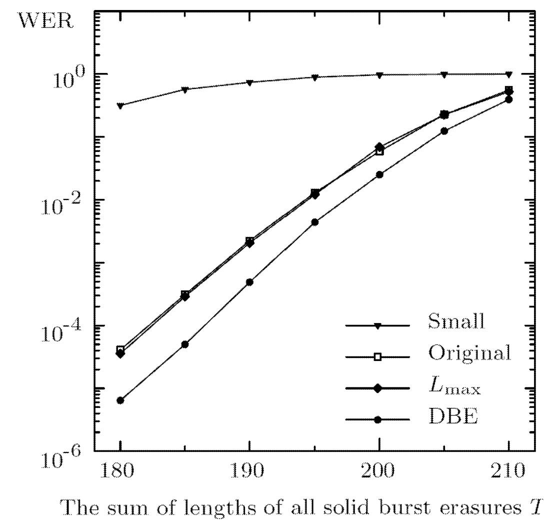

Fig. 9 Decoding performance for $L = 25$ solid burst erasures whose sum is T of the Code A.

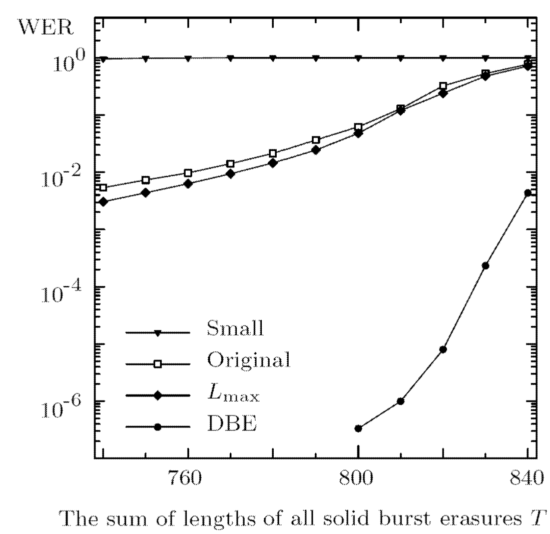

Fig. 10 Decoding performance for $L = 5$ solid burst erasures whose sum is T of the Code B.

the other codes under the condition that the code length N is constant, becomes smaller as the designed rate R' have a large value.

4.4 Decoding Performance for AWGN Channel

Figure 12 shows decoding performance on the AWGN channel of the Code A. The horizontal axis shows Eb/N_0 and the vertical axis shows bit error rate (BER). From Fig. 12, all of the codes have almost the same performance. Since the codes "DBE," "$L_{\max}$," and "Small" are constructed from the parity-check matrix of the code "Original" by column permutation, the performance for random error correction capabilities of those codes are the same.

4.5 Discussion

From the simulation results in Sect. 4.3, the performance of our proposed codes (the code "DBE") depends on the total number of erasures T. When $T \leq L_{\max}$ and $L \geq 2$, the difference of the performance between proposed codes and the other codes is large and becomes larger as T tends to have a small value.

The key idea of a column permutation method for a parity-check matrix of LDPC codes decoded by the SP decoding algorithm is to make consecutive positions of stopping sets separate. From Definition 1, the submatrix of a parity-check matrix whose column's positions are contained a stopping set always has equal to or more than 2 elements of 1 in all of those rows. Therefore, having large values of DBEs leads to making consecutive positions of stopping

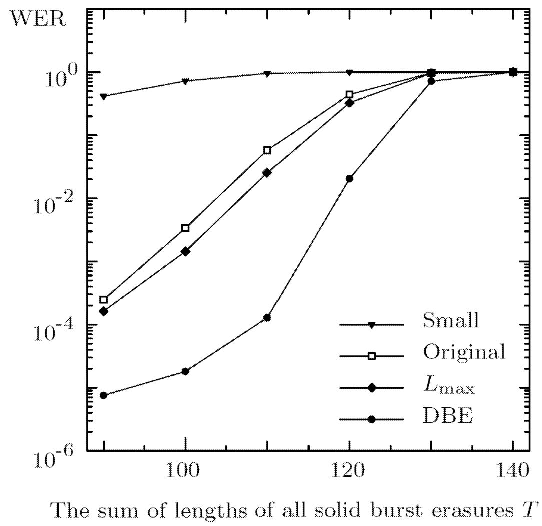

Fig. 11 Decoding performance for L = 5 solid burst erasures whose sum is T of the Code C.

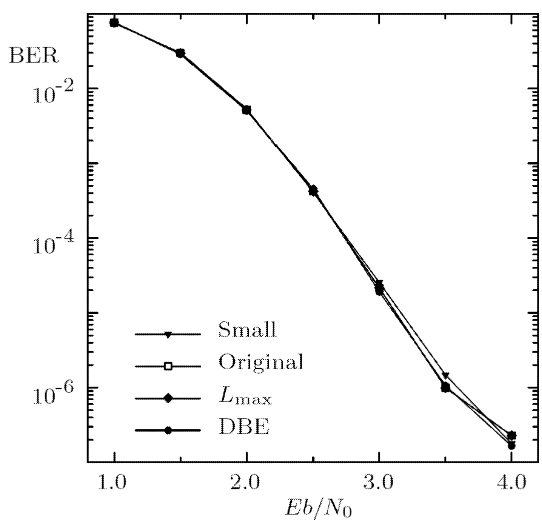

Fig. 12 Decoding performance for AWGN channels of the Code A.

sets separate. From a theoretical point of view, the minimum value of DBE $D_{\min}$ only indicates the correction capability at the first iteration by SP decoding algorithm for a solid burst.

5. Concluding Remarks

In this paper, we have proposed a column permutation method for a parity-check matrix of the LDPC codes for L ($\geq$ 2) burst erasures. From simulation results, our proposed method performs well when L ($\geq$ 2) is large. We also show that the performance of the code having small values of DBEs is bad.

Theoretical performance analyses waits for further investigation. Simulations for LDPC codes with other parameters are also for future work.

Acknowledgments

The authors wish to thank the reviewers for their thorough and conscientious reviewing for this paper. One of the authors, G. Hosoya, would like to thank Dr. M. Kobayashi at Shonan Institute of Technology for his valuable comments. This work is partially supported by Waseda University Grant for Special Research Project No. 2005A-952 and No. 2005B-189.

References

[1] R.G. Gallager, "Low density parity check codes," IRE Trans. Inf. Theory, vol.8, pp.21–28, Jan. 1962.
[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] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, "Improved low-denstiy parity-check codes using irregular graphs," IEEE Trans. Inf. Theory, vol.47, no.2, pp.585–598, Feb. 2001.
[4] T. Richardson and R.L. Urbanke, "Efficient encoding of low-density parity-check codes," IEEE Trans. Inf. Theory, vol.47, no.2, pp.638–656, Feb. 2001.
[5] C. Di, D. Proietti, I.E. Telatar, T.J. Richardson, and R.L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," IEEE Trans. Inf. Theory, vol.48, no.6, pp.1570–1579, June 2002.
[6] G. Hosoya, H. Yagi, M. Kobayashi, and S. Hirasawa, "Construction methods of low-density parity-check codes for burst error channels," IEICE Technical Report, IT2003-20, July 2003.
[7] M. Yang and W.E. Ryan, "Performance of efficiently encordable low-density parity-check codes in noise bursts on the EPR4 channel," IEEE Trans. Magn., vol.40, no.2, pp.507–512, March 2004.
[8] T. Wadayama, "Greedy construction of LDPC codes for burst erasure channel," IEICE Technical Report, IT2004-44, Sept. 2004.
[9] J. Garcia-Frias, "Decoding of low-density parity check codes over finite-state binary Markov channel," IEEE Trans. Commun., vol.52, no.11, pp.1840–1843, Nov. 2004.
[10] A.W. Eckford, F.R. Kschischang, and S. Pasupathy, "Analysis of low-density parity-check codes for the Gilbert-Elliott channel," IEEE Trans. Inf. Theory, vol.51, no.11, pp.3872–3889, Nov. 2005.

Appendix: Proof of Theorem 1

Assume that H has a form of Eq. (5) that can maximize the value of D_{ave}. For $v \in [1, \rho']$,

$$\#\{m : n_{m,1} = v\} = w_c, \tag{A·1}$$

where $\#\{\cdot\}$ denotes the number of elements that constitute the set $\{\cdot\}$. Since the column position $\rho' + 1$ of H has r' rows that have $n_{m,1} = \rho' + 1$,

$$\#\{m \mid n_{m,1} = \rho' + 1\} = r', \tag{A·2}$$

holds. The above discussion is similar to the case of rightmost $\rho' + 1$ columns of H that constitute H_{rig}. For $v \in [N - \rho' + 1, N]$,

$$\#\{m \mid n_{m,w_r} = V\} = w_c, \tag{A·3}$$

$$\#\{m \mid n_{m,w_r} = N - \rho'\} = r', \tag{A·4}$$

hold. Substituing Eq. (4) to Eq. (3),

$$
\begin{aligned}
M(w_r-1)D_{\text{ave}} &= \sum_{m=1}^{M}\sum_{\gamma=1}^{w_r-1} D_{m\gamma} = \sum_{m=1}^{M}(n_{m,w_r}-n_{m,1}) \\
&= \sum_{m=1}^{M} n_{m,w_r} - \sum_{m=1}^{M} n_{m,1}.
\end{aligned}
$$

From Eqs. (A·1)–(A·4), and $\rho = \frac{N}{w_r}$, $\rho' = \frac{N}{w_r} - \frac{r'}{w_c}$, we have

$$
\begin{aligned}
&\sum_{m=1}^{M} n_{m,w_r} - \sum_{m=1}^{M} n_{m,1} \\
&\quad = w_c\Big(\sum_{n\in[N-\rho'+1,N]} n - \sum_{n\in[1,\rho']} n\Big) \\
&\qquad + r'\{(N-\rho')-(\rho'+1)\} \\
&\quad = w_c\rho'(N-\rho') + r'(N-2\rho'-1) \\
&\quad = M(w_r-1)\times\rho + r'(\rho-\rho'-1) \\
&\quad = M(w_r-1)\times\rho + r'\Big(\frac{r'}{w_c}-1\Big). \qquad \text{(A·5)}
\end{aligned}
$$

If M mod $w_c = 0$, we have $r' = 0$ and

$$D_{\text{ave}} = \rho. \qquad \text{(A·6)}$$

If M mod $w_c \neq 0$, since $w_c > r' > 0$, the second term of Eq. (A·5) takes a negative value. i.e.,

$$D_{\text{ave}} < \rho. \qquad \text{(A·7)}$$

If the parity-check matrix H does not have the form of Eq. (5), Eq. (A·7) holds from Lemma 1. Therefore from Eqs. (A·6) and (A·7), Eq. (6) holds. □

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. He is currently a doctoral student in Industrial and Management Systems Engineering at the Graduate School of Waseda University. His research interests are coding theory and information theory.

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. He is currently a research associate at the Media Network Center, Waseda University. His research interests are coding and information security. He is a member of the Society of Information Theory and Its Applications.

Toshiyasu Matsushima was born in Tokyo, Japan, on Nov. 26, 1955. 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 1978, 1980, and 1991, respectively. From 1980 to 1986, he was with Nippon Electric Corporation, Kanagawa, Japan. From 1986 to 1992, he was a lecturer at Department of Management Information, Yokohama College of Commerce. From 1993, he was an associate professor and since 1996 has been a professor of School of Science and Engineering, Waseda University, Tokyo, Japan. From 2001 to 2002, he was a Visiting Professor of the University of Hawaii at Manoa, U.S. From 2005, he has been the Chairperson of the Technical Group on Information Theory of IEICE. His research interests are information theory and its application, statistics, and artificial intelligence. He is a member of the IEEE, the Society of Information Theory and its Applications, the Japan Society for Quality Control, the Japan Industrial Management Association, and the Japan Society of Artificial Intelligence.

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 Fellow, and a member of Soc. of ITA, the Information Processing Society of Japan, and the Japan Industrial Management Association.