PAPER

On the Minimum Weight of Simple Full-Length Array LDPC Codes

Kenji SUGIYAMA†a), *Nonmember and* **Yuichi KAJI**†b), *Member*

SUMMARY We investigate the minimum weights of simple full-length array LDPC codes (SFA-LDPC codes). The SFA-LDPC codes are a subclass of LDPC codes, and constructed algebraically according to two integer parameters p and j. Mittelholzer and Yang et al. have studied the minimum weights of SFA-LDPC codes, but the exact minimum weights of the codes are not known except for some small p and j. In this paper, we show that the minimum weights of the SFA-LDPC codes with $j = 4$ and $j = 5$ are upper-bounded by 10 and 12, respectively, independent from the prime number p. By combining the results with Yang's lower-bound limits, we can conclude that the minimum weights of the SFA-LDPC codes with $j = 4$ and $p > 7$ are exactly 10 and those of the SFA-LDPC codes with $j = 5$ are 10 or 12.

key words: *LDPC code, simple full-length array LDPC code, minimum weight, minimum distance*

1. Introduction

This paper discusses the minimum weights of codes in a certain subclass of *low-density parity check codes* (*LDPC codes*). The *minimum weight* (the *minimum distance*) is one of the most fundamental and the most significant parameters to evaluate the performance of linear block codes. However, unfortunately, it is difficult in general to obtain the exact minimum weight of a long practical LDPC code. Tanner discussed in [11] a certain bound on the minimum weights of LDPC codes by using the Tanner Graph. Hu and Fossorier proposed a probabilistic procedure to compute the minimum weight of LDPC codes by using a decoding algorithm for LDPC codes [6], and Hirotomo et al. proposed another probabilistic procedure [5] that uses the Stern's algorithm.

If we restrict ourselves to LDPC codes with certain structures, then more analytical approach is possible. MacKay showed that the minimum weights of regular-Quasi-Cyclic LDPC codes with column weight j are less than or equal to $(j + 1)!$ [7], and Fossorier discusses the minimum distance of Quasi-Cyclic LDPC codes from circulant permutation matrices [4]. Mittelholzer has derived in [8] some upper-bound limits of the minimum weights of certain *array LDPC codes* [2]. Array LDPC codes are a class of LDPC codes that are algebraically constructed from a family of array codes [1], [3]. Mittelholzer assumes some more additional conditions on the array LDPC codes, and thus it will be better to distinguish the investigated class from the class of general array LDPC codes. In this paper, we name *simple full-length array LDPC codes* (*SFA-LDPC codes*) for the class of LDPC codes that Mittelholzer investigates. A SFA-LDPC code is defined according to two integer parameters p and j. Let $C_A(p, j)$ denote the SFA-LDPC code defined by p and j. Mittelholzer showed that the minimum weight of the SFA-LDPC code $C_A(p, 4)$ is 12 or less [8], which significantly improves the upper bound limit $(4 + 1)! = 120$ given by MacKay [7] for general regular LDPC codes. Mittelholzer also showed that the minimum weight of the SFA-LDPC code $C_A(p, 5)$ is 20 or less.

The study of Mittelholzer is followed by Yang et al. in [12]. Mittelholzer discussed the upper-bounds of the minimum weights of $C_A(p, j)$, but Yang discussed their lower-bounds. With very careful analysis, Yang showed that the minimum weight of $C_A(p, 4)$ is 10 or more if p is a prime number greater than 7. Together with Mittelholzer's upper-bound, this result implies that the minimum weight of $C_A(p, 4)$ is either 10 or 12, because $C_A(p, j)$ does not have odd-weight codewords.

In this study, we partly make use of Yang's analytical approach to discuss the upper-bound limits of SFA-LDPC codes. Through the analytical discussion, we can consider minimum weight codewords in a certain "normal form." We generated the minimum weight codewords in the normal form for several code parameters, and found that the generated codewords have common structures that are independent from the parameter p choices. By carefully analyzing the structure, we are able to show that $C_A(p, 4)$ contains a codeword with weight 10, and that $C_A(p, 5)$ contains a codeword with weight 12. Together with the previously known results, this implies that the minimum weight of $C_A(p, 4)$ is exactly 10 for $p > 7$, and that the minimum weight of $C_A(p, 5)$ is either 10 or 12.

2. Preliminaries and Known Results

Let p be a prime number, and k and j be integers satisfying $k, j \le p$. The binary *simple array LDPC code* $C_A(p, j, k)$ is the null space of the $pj \times pk$ binary matrix

Manuscript received June 13, 2007.
Manuscript revised January 19, 2008.

[†] The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-0101 Japan.
a) E-mail: kenji-su@is.naist.jp
b) E-mail: kaji@is.naist.jp

DOI: 10.1093/ietfec/e91–a.6.1502

Table 1 The minimum weights of $C_A(p, j)$.

p	$j = 4$	$j = 5$	$j = 6$
5	8	–	–
7	8	12	12
11	10	10	16
13	10	12	14
17	10	12	N/A
19	10	12	N/A
23	10	N/A	N/A
⋮	⋮	⋮	⋮
79	10	N/A	N/A

"N/A" denotes that the results are not available due to large computational time.

$$H_A(p, j, k) = \begin{bmatrix} I & I & \cdots & I \\ I & P & \cdots & P^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ I & P^{j-1} & \cdots & P^{(j-1)(k-1)} \end{bmatrix}$$

where I is the $p \times p$ identity matrix and P is a cyclic shift matrix defined by

$$P = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Now we consider a special case such that $k = p$, and call this special array LDPC code a *simple full-length array LDPC code* (*SFA-LDPC code* for short). The "array LDPC codes" discussed in [8] and [12] are indeed this SFA-LDPC codes. For simplicity, $C_A(p, j, p)$ and $H_A(p, j, p)$ are respectively written as $C_A(p, j)$ and $H_A(p, j)$. We denote by $d(p, j)$ the *minimum distance* of $C_A(p, j)$.

In the rest of this section, we summarize known results on the minimum weights of SFA-LDPC codes.

For the case $j = 2$ We can easily show that $d(p, 2) = 4$ for any prime number $p \geq 3$.

For the case $j = 3$ Yang et al. has shown that $d(p, 3) = 6$ [12]

For the case $j = 4$ Mittelholzer showed that $d(p, 4)$ is 12 or less [8]. Yang showed that $d(p, 4) = 8$ for $p = 5$ and $p = 7$, and that $d(p, 4)$ is 10 or more if $p > 7$ [12]. Therefore $d(p, 4)$ with $p > 7$ is either 10 or 12.

For the case $j = 5$ Mittelholzer showed that $d(p, 5)$ is 24 or less [8]. The lower-bound of $d(p, 5)$ with $p > 7$ is 10 or more, which is obtained from Yang's result for $j = 4$, and a simple observation that $d(p, j_1) \geq d(p, j_2)$ if $j_1 \geq j_2$.

For the case $j = 6$ Mittelholzer showed that $d(p, 6)$ is 32 or less [8]. The lower-bound of $d(p, 6)$ with $p > 7$ is 10 or more, as in the case $j = 5$.

The authors have investigated the minimum weights of SFA-LDPC codes from an experimental approach [9]. We developed an algorithm that efficiently generates some of minimum weight codewords, and have computed exact minimum weights of some SFA-LDPC codes. The results are summarized in Table 1. We can see from the table that $d(p, 4)$ is 10 for all prime numbers p from 11 to 79. From this result, it comes quite natural to conjecture that $d(p, 4)$ is always 10 for an arbitrary p greater than seven. Similarly, for the case $j = 5$, $d(p, 5)$ can be upper-bounded by 12 for an arbitrary p greater than seven. The following sections are to give analytical and constructive proofs to these conjectures.

3. Structures and Properties of $H_A(p, j)$

This section is to discuss the structural property of the check matrices of SFA-LDPC codes.

3.1 The ϕ Notation

It can be easily shown that column vectors in $H_A(p, j)$ are all different, and that every codeword in $C_A(p, j)$ has even weight. It is also obvious from the definition that a column vector of $H_A(p, j)$ can be decomposed into j subsequences that have length p and weight one. That is, if we write $H_A(p, j) = [h_{l,i}]$ $(1 \leq i \leq p^2$ and $1 \leq l \leq pj)$, then the weight of the subsequence

$$\boldsymbol{h}_i^r = (h_{p(r-1)+1,i}, h_{p(r-1)+2,i}, \ldots, h_{p(r-1)+p,i})^T$$

is exactly one for any $1 \leq i \leq p^2$ and $1 \leq r \leq j$.

For a vector $\boldsymbol{v}$ with weight one, let $\phi(\boldsymbol{v})$ denote the position of the nonzero component in $\boldsymbol{v}$ where the first component of $\boldsymbol{v}$ is indexed as zero. For example, $\phi((0, 1, 0, 0)^T) = 1$ and $\phi((0, 0, 0, 1)^T) = 3$. The value of ϕ is not defined for a vector whose weight is not one. Extend this ϕ notation to a column vector $\boldsymbol{h}_i = (\boldsymbol{h}_i^{1T}, \boldsymbol{h}_i^{2T}, \ldots, \boldsymbol{h}_i^{jT})^T = (h_{1,i}, h_{2,i}, \ldots, h_{pj,i})^T$ of $H_A(p, j)$ in such a way that

$$\phi(\boldsymbol{h}_i) = (\phi(\boldsymbol{h}_i^1), \ldots, \phi(\boldsymbol{h}_i^j))^T,$$

and also extend the notation to the matrix $H_A(p, j)$ as

$$\phi(H_A(p, j)) = [\phi(\boldsymbol{h}_1), \ldots, \phi(\boldsymbol{h}_{p^2})].$$

For example, the check matrix of $C_A(3, 2)$

$$H_A(3, 2) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

is written by the ϕ notation as

$$\phi(H_A(3, 2)) = \begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \end{bmatrix}. \tag{1}$$

The ϕ notation contributes to represent vectors and matrices in a compact manner. Furthermore, it is helpful to understand the structure of check matrices of SFA-LDPC codes, as illustrated in the following lemmas.

Lemma 3.1: For integers k and k' with $0 \le k, k' < p$, the $(pk + k' + 1)$-th column vector of $\phi(H_A(p, j))$ is

$$(k', k' + k, \ldots, k' + (j-1)k)^T \quad (\text{mod } p).$$

Proof. Obvious from the construction of $H_A(p, j)$. □

Lemma 3.1 means that components in a column vector in $\phi(H_A(p, j))$ constitute an arithmetic sequence. The next lemma implies that two different column vectors in $\phi(H_A(p, j))$ cannot have two or more common components in common positions.

Lemma 3.2: Consider two different columns $\phi(\boldsymbol{h}_{i_1})$ and $\phi(\boldsymbol{h}_{i_2})$ in $\phi(H_A(p, j))$ (thus $i_1 \ne i_2$ and $1 \le i_1, i_2 \le p^2$). If the j_1-th components of $\phi(\boldsymbol{h}_{i_1})$ and $\phi(\boldsymbol{h}_{i_2})$ are the same, then, for any other j_2 with $j_1 \ne j_2$ and $1 \le j_2 \le j$, the j_2-th components of $\phi(\boldsymbol{h}_{i_1})$ and $\phi(\boldsymbol{h}_{i_2})$ cannot be the same.

Proof.

This lemma is easily shown since the construction of SFA-LDPC codes satisfies the condition of the Theorem2.2 of [4].

□

Note that $\phi(H_A(p, j))$ has p^2 column vectors. Hence the next corollary follows from Lemma 3.2.

Corollary 3.3: For any j_1 and j_2 with $1 \le j_1, j_2 \le j$ and $j_1 \ne j_2$, and for any a and b with $0 \le a, b \le p - 1$, $\phi(H_A(p, j))$ contains exactly one column vector whose j_1-th and j_2-th components are a and b, respectively.

3.2 Supports and Cancel-Out Condition

For a binary linear block code C, a vector $\boldsymbol{v}$ is a codeword of C if and only if $H\boldsymbol{v}^T = 0 \bmod 2$ where H is the check matrix of C. This basic property can be stated in terms of the ϕ notation.

Definition 3.4: A collection of integers $q_1, q_2, \ldots, q_n$ is said to satisfy the *cancel-out condition* if no integer appears odd times in $q_1, q_2, \ldots, q_n$.

Lemma 3.5: Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be binary vectors with length p and weight one. We have $\boldsymbol{v}_1 + \cdots + \boldsymbol{v}_n = \boldsymbol{0} \bmod 2$ if and only if the collection $\phi(\boldsymbol{v}_1), \ldots, \phi(\boldsymbol{v}_n)$ satisfies the cancel-out condition.

Proof. Obvious since the sum is taken under the modulus of two. □

Let $\boldsymbol{v} = (v_1, \ldots, v_{p^2})$ be a binary vector of length p^2, and let $\boldsymbol{h}_i$ denote the i-th column vector of $H_A(p, j)$. The vector $\boldsymbol{v}$ is a codeword of $C_A(p, j)$ if and only if $H_A(p, j)\boldsymbol{v}^T = \boldsymbol{0}$ where the operations are taken under the modulus of two. Thus, $\boldsymbol{v} \in C_A(p, j)$ if and only if column vectors in

$$\{\boldsymbol{h}_i | 1 \le i \le p^2, v_i = 1\}$$

sum to zero under the modulus of two. Now define

$$\text{supp}(\boldsymbol{v}) = \{\phi(\boldsymbol{h}_i) | 1 \le i \le p^2, v_i = 1\},$$

and call it the *support*[†] of $\boldsymbol{v}$. The support contains the column vectors of $\phi(H_A(p, j))$ that correspond to nonzero components of $\boldsymbol{v}$. When there is no fear of confusion, we write a support (a set of column vectors) as a matrix, and call it a *support matrix* of the vector $\boldsymbol{v}$. For example, a support

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\}$$

can be written as

$$\begin{bmatrix} 0 & 0 & 2 \\ 0 & 1 & 0 \end{bmatrix}.$$

We note that the order of column vectors in a support matrix has no significance.

If a vector $\boldsymbol{v}$ is a correct codeword, then the binary representation of column vectors in the support $\text{supp}(\boldsymbol{v})$ sum to zero. By applying Lemma 3.5 to components of column vectors in the support, we have the following corollaries.

Corollary 3.6: A vector $\boldsymbol{v}$ is a codeword of $C_A(p, j)$ if and only if the cancel-out condition holds for all rows of the support matrix of $\boldsymbol{v}$. In this case we say that the support $\text{supp}(\boldsymbol{v})$ (and its corresponding support matrix) satisfies the cancel-out condition.

Corollary 3.7: The SFA-LDPC code $C_A(p, j)$ contains a codeword of weight w if and only if $\phi(H_A(p, j))$ has a submatrix that has w columns and satisfies the cancel-out condition.

Consider for example the check matrix $\phi(H_A(3, 2))$ given in (1). If we choose from $\phi(H_A(3, 2))$ four column vectors that are represented as

$$\begin{bmatrix} 0 & 0 & 2 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix},$$

then the support matrix satisfies the cancel-out condition. Note that the four column vectors constitute the support $\text{supp}(\boldsymbol{v})$ for $\boldsymbol{v} = 100101001$, and that $\boldsymbol{v}$ is a correct codeword of $C_A(3, 2)$. Any combination of three or less column vectors cannot satisfy the cancel-out condition for this matrix, and we can conclude that the minimum weight of $C_A(3, 2)$ is four.

3.3 Support Matrices in a Normal Form

To discuss the minimum weights of SFA-LDPC codes, we investigate supports that are in a special form. Assume that an SFA-LDPC code $C_A(p, j)$ has a codeword $\boldsymbol{v}$ which has a nonzero component at the first symbol position. Since the first column vector of $\phi(H_A(p, j))$ is an all-zero vector of length j, $\text{supp}(\boldsymbol{v})$ must contain an all-zero vector of length j. On the other hand, Corollary 3.6 implies that j zeros in the

[†] The word "support" is often used to denote the set of positions of nonzero components in a vector, but we use the word in slightly different manner.

$$\begin{bmatrix} 0 & 0 & s_{1,3} & s_{1,4} & \cdots & s_{1,j+1} & s_{1,j+2} & \cdots & s_{1,w} \\ 0 & s_{2,2} & 0 & s_{2,4} & \cdots & s_{2,j+1} & s_{2,j+2} & \cdots & s_{2,w} \\ 0 & s_{3,2} & s_{3,3} & 0 & \cdots & s_{3,j+1} & s_{3,j+2} & \cdots & s_{3,w} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & s_{j,2} & s_{j,3} & s_{j,4} & \cdots & 0 & s_{j,j+2} & \cdots & s_{j,w} \end{bmatrix}$$

Fig. 1 A Support matrix of a codeword whose weight is w.

all-zero vector must be canceled-out by other column vectors in the support. Since a non-zero vector cancels at most one zero in the all-zero vector (due to Lemma 3.2), supp(v) must contain another j column vectors each of which cancels one of zeros in the all-zero vector. Therefore, without loss of generality, the support matrix of v can be written as in Fig. 1. (Note again that the order of column vectors has no significance in a support matrix, and we have arranged the columns so that the positions of zeros are clearly illustrated as in Fig. 1). In this paper, a support matrix that can be written as in Fig. 1 is referred as a support matrix *in a normal form*.

An interesting property of SFA-LDPC code is that, if $C_A(p, j)$ has a codeword with weight w, then the code also has a codeword that has the same weight w and whose first component is non-zero (Lemma 2 of [12]). Combining this result with Corollary 3.7, we obtain the relation; $C_A(p, j)$ contains a codeword with weight w if and only if $\phi(H_A(p, j))$ has a submatrix "in the normal form" that has w column vectors and satisfies the cancel-out condition.

Support matrices in the normal form have some more useful properties. For example, any of components $s_{i,k}$ with $1 \le i \le j$, $2 \le k \le j+1$ and $i \ne k+1$ in Fig. 1 cannot be zero because of Lemma 3.2. We also remind that components in a column vector of Fig. 1 is an arithmetic sequence, and hence a column vector of Fig. 1 is uniquely determined if two components in the vector are specified. These properties will be further investigated in the following sections to discuss the minimum weights of individual SFA-LDPC codes.

4. New Bounds on the Minimum Weights of SFA-LDPC Codes

4.1 The Upper Bound of $d(p, 4)$

4.1.1 Overview of the Discussion

In this section, we extend our preliminary results in [9] to general SFA-LDPC codes $C_A(p, 4)$, and show that $d(p, 4)$ is 10 or less for any $p > 7$. The discussion in this section is developed according to the following scenario: First, we briefly introduce our experimental results concerning the minimum weight codewords of $C_A(p, 4)$ for several choices of p. By investigating the experimental results analytically, we can discover a universal structure that can be found independent from the prime number p. The structure is further studied, and we derive conditions for $H_A(p, 4)$ to have a support matrix that has 10 columns and satisfies the cancel-out condition. The discussion completes by showing that the derived conditions always hold for $H_A(p, 4)$ with an arbitrary $p > 7$.

$$\begin{bmatrix} 0 & 0 & s_{1,3} & s_{1,4} & s_{1,5} & s_{1,6} & s_{1,7} & s_{1,8} & s_{1,9} & s_{1,10} \\ 0 & s_{2,1} & 0 & s_{2,4} & s_{2,5} & s_{2,6} & s_{2,7} & s_{2,8} & s_{2,9} & s_{2,10} \\ 0 & s_{3,1} & s_{3,2} & 0 & s_{3,5} & s_{3,6} & s_{3,7} & s_{3,8} & s_{3,9} & s_{3,10} \\ 0 & s_{4,1} & s_{4,2} & s_{4,3} & 0 & s_{4,6} & s_{4,7} & s_{4,8} & s_{4,9} & s_{4,10} \end{bmatrix} \quad \text{(A)}$$

$$\begin{bmatrix} 0 & 0 & s_{1,3} & s_{1,4} & s_{1,5} & s_{1,5} & s_{1,4} & s_{1,3} & s_{1,9} & s_{1,9} \\ 0 & s_{2,1} & 0 & s_{2,4} & s_{2,5} & s_{2,1} & s_{2,7} & s_{2,5} & s_{2,7} & s_{2,4} \\ 0 & s_{3,1} & 1 & 0 & s_{3,5} & 1 & s_{3,5} & s_{3,8} & s_{3,8} & s_{3,1} \\ 0 & s_{4,1} & 2 & 1 & 0 & s_{4,6} & s_{4,1} & 1 & s_{4,6} & 2 \end{bmatrix} \quad \text{(B)}$$

$$\begin{bmatrix} 0 & 0 & -1 & -2 & 3s_{3,5} & 3s_{3,5} & -2 & -1 & s_{1,9} & s_{1,9} \\ 0 & s_{2,1} & 0 & -1 & 2s_{3,5} & s_{2,1} & s_{2,7} & 2s_{3,5} & s_{2,7} & -1 \\ 0 & 2s_{2,1} & 1 & 0 & s_{3,5} & 1 & s_{3,5} & s_{3,8} & s_{3,8} & 2s_{2,1} \\ 0 & 3s_{2,1} & 2 & 1 & 0 & s_{4,6} & 3s_{2,1} & 1 & s_{4,6} & 2 \end{bmatrix} \quad \text{(C)}$$

Fig. 2 A normal form support matrix with weight 10.

4.1.2 Support Matrices and Their Common Structure

By using a computer program, the authors have generated minimum weight codewords of $C_A(p, 4)$ for some prime numbers p between 11 and 79. The algorithm for generating the codewords is not the subject of this paper, but it can be found in [9], [10]. Among minimum weight codewords, we restrict ourselves to those that have nonzero symbols at the first positions, compute their support matrices, and reorder the column vectors of the matrices so that the matrices become the normal form. Since the minimum weights of $C_A(p, 4)$ is ten for all primes between 11 and 79, the support matrices in the normal form are thus written as in Fig. 2(A).

In general, one code $C_A(p, 4)$ contains several minimum weight codewords with nonzero symbols at the first positions, and hence we have several support matrices in the normal form for one code. We classified the support matrices according to the cancel-out patterns of components in the matrices, and found that an identical cancel-out pattern appears in different choices of the prime number p. For example, $\phi(H_A(11, 4))$, $\phi(H_A(13, 4))$ and $\phi(H_A(17, 4))$ respectively have the following support matrices;

$$\begin{bmatrix} 0 & 0 & 10 & 9 & 5 & 5 & 9 & 10 & 3 & 3 \\ 0 & 3 & 0 & 10 & 7 & 3 & 9 & 7 & 9 & 10 \\ 0 & 6 & 1 & 0 & 9 & 1 & 9 & 4 & 4 & 6 \\ 0 & 9 & 2 & 1 & 0 & 10 & 9 & 1 & 10 & 2 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 0 & 12 & 11 & 6 & 6 & 11 & 12 & 4 & 4 \\ 0 & 10 & 0 & 12 & 4 & 10 & 0 & 4 & 0 & 12 \\ 0 & 7 & 1 & 0 & 2 & 1 & 2 & 9 & 9 & 7 \\ 0 & 4 & 2 & 1 & 0 & 5 & 4 & 1 & 5 & 2 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 0 & 16 & 15 & 8 & 8 & 15 & 16 & 6 & 6 \\ 0 & 13 & 0 & 16 & 11 & 13 & 6 & 11 & 6 & 16 \\ 0 & 9 & 1 & 0 & 14 & 1 & 14 & 6 & 6 & 9 \\ 0 & 5 & 2 & 1 & 0 & 6 & 5 & 1 & 6 & 2 \end{bmatrix}.$$

The components in the matrices vary in the above three examples, but we can see that the matrices satisfy the cancel-out pattern depicted in Fig. 2(B). Remind the property that the components in a column vector of $\phi(H_A(p, j))$ constitute arithmetic sequences, and we can easily understood that the common differences of the second, third, fourth and the fifth columns of the matrix in Fig. 2(B) are $s_{2,1}$, 1, 1 and $-s_{3,5}$ (mod p), respectively. By using this property, the matrix in Fig. 2(B) can be expressed as in Fig. 2(C), where the components are under the modulus of p.

4.1.3 Conditions on the Matrix Components

Let d_k with $k = 6, \dots, 10$ be common differences of the sequences in the k-th column vector of the matrix in Fig. 2(C). From the relation of components in the the rightmost (the tenth) column of the matrix, we have $2 \equiv -1 + 2k_{10} \bmod p$. This implies that d_{10} must satisfy

$$2d_{10} \equiv 3 \bmod p. \tag{2}$$

The values of $s_{1,9}$ and $2s_{2,1}$ are also represented by using d_{10} as $s_{1,9} \equiv -1 - d_{10}$ and $2s_{2,1} \equiv -1 + d_{10}$. Multiply both sides of the equations by two and we have

$$2s_{1,9} \equiv -2 - 2k_{10} \equiv -5 \bmod p, \tag{3}$$

$$4s_{2,1} \equiv -2 + 2k_{10} \equiv 1 \bmod p. \tag{4}$$

Now let us turn our attention to the sixth column of the matrix. Because $1 - d_6 \equiv s_{2,1}$, we have $4 - 4d_6 \equiv 4s_{2,1} \equiv 1$ and hence

$$4d_6 \equiv 4 - 1 \equiv 3 \bmod p. \tag{5}$$

As for the eighth column, $1 \equiv -1 + 3d_8$ and thus

$$3d_8 \equiv 1 + 1 \equiv 2 \bmod p. \tag{6}$$

We also have the relation $2s_{3,5} \equiv -1 + d_8$ from the sequence in the eighth column. Multiply both sides of the relation by three,

$$6s_{3,5} \equiv -3 + 3d_8 \equiv -1 \bmod p. \tag{7}$$

In the seventh column, we have $s_{3,5} \equiv -2 + 2d_7$. Multiply the relation by six, and

$$12d_7 \equiv 12 + 6s_{3,5} \equiv 11 \bmod p. \tag{8}$$

Finally, consider the relation $s_{2,7} \equiv s_{1,9} + d_9$ in the ninth column, $s_{2,7} \equiv -2 + d_7$ in the seventh column and the conditions (3) and (8), and we can show that

$$12d_9 \equiv 17 \bmod p. \tag{9}$$

Other unknown variables $s_{2,7}$, $s_{3,8}$ and $s_{4,6}$ are uniquely determined once the above variables are determined, and hence relations (2) through (9) completely state the condition with which the matrix in Fig. 2(C) satisfies the cancel-out condition.

Note that, if p is a prime number greater than seven, then each of the above relation has a unique solution, since the coefficient of the variable in the relation is relatively prime to p. Consequently, we can determine a unique matrix that is shown in the form of Fig. 2(C), and satisfies the cancel-out condition. If we may use multiplicative inverses, then the matrix in Fig. 2(C) is written as in Fig. 3 under the above conditions. For the sake of readability, some components in the matrix are written in a reduced form; for example we write $7 \cdot 4^{-1}$ instead of $1 + 3 \cdot 4^{-1}$. Finally, remind the property that the check matrix $H_A(p, j)$ has p^2 different column vectors, and hence an arbitrary column vector which is represented as an arithmetic sequence belongs to $H_A(p, j)$ as a column vector. Consequently, $H_A(p, j)$ with $p > 7$ contains the above described unique matrix.

Summarizing the above discussion, we have the following theorem.

Theorem 4.1: $d(p, 4) \leq 10$ for any prime p with $p > 7$.

Combine this theorem with Yang's lower-bound limit $d(p, 4) \geq 10$ for $p > 7$, and we can fully clarified the minimum weights of SFA-LDPC codes with $j = 4$. The results, together with already known results, are summarized in the following corollary.

Corollary 4.2: $d(p, 4) = 8$ for $p = 5$ and 7, and $d(p, 4) = 10$ for any prime $p > 7$.

4.2 The Upper Bound of $d(p, 5)$

The upper-bound limit of $d(p, 5)$ can be derived in almost the same way as the case of $d(p, 4)$. The difference from the $j = 4$ case is that we need to consider support matrices with twelve column vectors instead of ten. We have collected support matrices in a normal form, categorized the cancel-out patterns and found a universal structure as in the case $j = 4$. To avoid lengthy step-by-step derivation of the conditions, we present only the final result in Fig. 4.

Theorem 4.3: $d(p, 5) \leq 12$ for any prime p with $p > 7$.

$$\begin{bmatrix} 0 & 0 & -1 & -2 & -2^{-1} & -2^{-1} & -2 & -1 & -5 \cdot 2^{-1} & -5 \cdot 2^{-1} \\ 0 & 4^{-1} & 0 & -1 & -3^{-1} & 4^{-1} & -13 \cdot 12^{-1} & -3^{-1} & -13 \cdot 12^{-1} & -1 \\ 0 & 2^{-1} & 1 & 0 & -6^{-1} & 1 & -6^{-1} & 3^{-1} & 3^{-1} & 2^{-1} \\ 0 & 3 \cdot 4^{-1} & 2 & 1 & 0 & 7 \cdot 4^{-1} & 3 \cdot 4^{-1} & 1 & 7 \cdot 4^{-1} & 2 \end{bmatrix}$$

Fig. 3 The support matrix in a normal form for $j = 4$.

$$\begin{bmatrix} 0 & 0 & -1 & -11 \cdot 3^{-1} & -3 & -2 \cdot 3^{-1} & -2 \cdot 3^{-1} & -3 & -11 \cdot 3^{-1} & -1 & -8 \cdot 3^{-1} & -8 \cdot 3^{-1} \\ 0 & 6^{-1} & 0 & -11 \cdot 6^{-1} & -2 & -2^{-1} & 6^{-1} & -4 \cdot 3^{-1} & -2 & -2^{-1} & -11 \cdot 6^{-1} & -4 \cdot 3^{-1} \\ 0 & 3^{-1} & 1 & 0 & -1 & -3^{-1} & 1 & 3^{-1} & -3^{-1} & 0 & -1 & 0 \\ 0 & 2^{-1} & 2 & 11 \cdot 6^{-1} & 0 & -6^{-1} & 11 \cdot 6^{-1} & 2 & 4 \cdot 3^{-1} & 2^{-1} & -6^{-1} & 4 \cdot 3^{-1} \\ 0 & 2 \cdot 3^{-1} & 3 & 11 \cdot 3^{-1} & 1 & 0 & 8 \cdot 3^{-1} & 11 \cdot 3^{-1} & 3 & 1 & 2 \cdot 3^{-1} & 8 \cdot 3^{-1} \end{bmatrix}$$

Fig. 4 The support matrix in a normal form for $j = 5$.

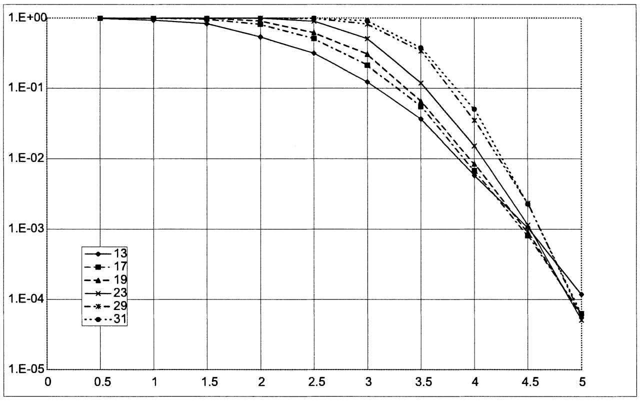

Fig. 5 Block error rate.

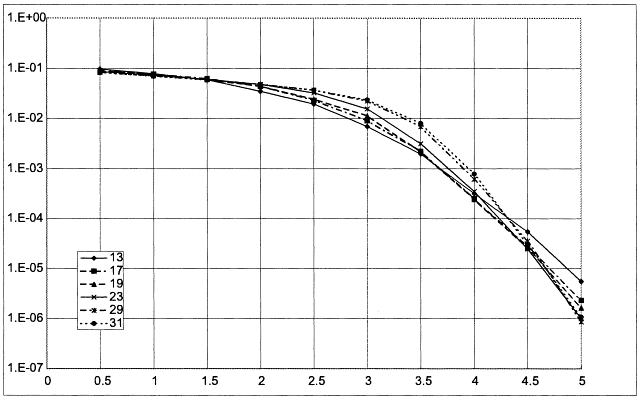

Fig. 6 Bit error rate.

Together with Yang's lower bound, we obtain following corollary.

Corollary 4.4: $10 \leq d(p, 5) \leq 12$ for $p > 7$.

5. Concluding Remarks

The minimum weights of SFA-LDPC codes are studied. The problem has been investigated by Mittelholzer [8] and Yang et al. [12], and this paper gives an answer to the problem for the case $j = 4$, namely the minimum weight of $C_A(p, 4)$ is exactly 10 for $p > 7$. For $j = 5$ case, we show that the minimum weight of $C_A(p, 5)$ is 12 or less, which significantly improves the upper-bound limit of 20 shown by Mittelholzer. We note that the approach considered in this study is applicable to the case with $j = 6$ or more, but we may face to difficulty in generating sufficient number of support matrices from that we can find a useful structure.

For the cases $j = 4$ and $j = 5$, the parameter p does not affect the minimum weight of the code unless $p \leq 7$. If we fix the parameter j and increase the value of the prime number p, then the code length of $C_A(p, j)$ increases while the minimum weight of the code stays unchanged. Obviously this degrades the error correcting capability of the code. On the other hand, increasing p also increases the code rate and thus the degradation of the error correcting capability is an unavoidable expense for increasing p. For interested readers, we present in Figs. 5 and 6 the block and bit error performances of $C_A(p, 4)$ for several p . The parameters of the used codes are given in Table 2. The performance results are obtained by the standard sum-product decoding algorithm with twenty iterations at maximum. For low SNR, shorter codes show better performance than longer codes, which is a natural consequence of the observation on the code length and the minimum weight. On the other hand, short codes suffer for degradation of the performance for high SNR. This result clearly illustrates that the minimum distance is just one of parameters that affect the performance of the code. We can see that longer code length gives significant contribution on the error correcting performance especially when use an iterative decoding.

Table 2 Code parameters.

p	N	K	rate	$d(p, 4)$
13	169	120	0.710	10
17	289	224	0.775	10
19	361	288	0.798	10
23	529	440	0.832	10
29	841	728	0.866	10
31	961	840	0.874	10

References

[1] M. Blaum and R.M. Roth, "New array codes for multiple phased burst correction," IEEE Trans. Inf. Theory, vol.39, no.1, pp.66–77, Jan. 1993.

[2] J.L. Fan, "Array codes as low-density parity-check codes," Proc. 2nd Int. Symp. on Turbo Codes, pp.543–546, 2000.

[3] P.G. Farrell, "A survey of array error control codes," European Trans. Telecommunications, vol.3, no.5, pp.441–454, 1992.

[4] M.P.C. Fossorier, "Quasi-cyclic low density parity check codes from circulant permutation matrices," IEEE Trans. Inf. Theory, vol.50, no.8, pp.1788–1793, Aug. 2004.

[5] M. Hirotomo, M. Mohri, and M. Morii, "A probabilistic computation method for the weight distribution of low-density parity-check codes," Proc. IEEE Intl. Symp. Information Theory, pp.2166–2170, Adelaide, Australia, Sept. 2005.

[6] X.Y. Hu and M.P.C. Fossorier, "On the computation of the minimum distance of low-density parity-check codes," Proc. IEEE International Conference on Communications, Paris, June 2004.

[7] D.J.C. MacKay and M. Davey, "Evaluation of Gallager codes for short block length and high rate applications," IMA Workshop on Codes, Systems and Graphical Models, 1999.

[8] T. Mittelholzer, "Efficient encoding and minimum distance bounds of reed-solomon-type array codes," Proc. 2002 IEEE Int. Symp. on Information Theory, p.282, 2002.

[9] K. Sugiyama and Y. Kaji, "A minimum weight test for a certain subclass of array LDPC codes," Proc. of Intl. Symp. Information Theory and Its Applications, pp.366–371, Seoul, Korea, Nov. 2006.

[10] K. Sugiyama and Y. Kaji, "A minimum weight test procedure for a certain subclass of array LDPC codes," NAIST technical report, no.2008001, Jan. 2008, http://isw3.naist.jp/IS/TechReport

[11] R.M. Tanner, "Minimum-distance bounds by graph analysis," IEEE Trans. Inf. Theory, vol.47, no.2, pp.808–821, Dec. 2003.

[12] K. Yang and T. Helleseth, "On the minimum distance of array codes as LDPC codes," IEEE Trans. Inf. Theory, vol.49, no.12, pp.3268–3271, Dec. 2003.

Kenji Sugiyama was born in Osaka, Japan, on November 27, 1970. He received the bachelor's degree in mathematics from Yamaguchi University in 1994, and the M.E. degree in information and computer sciences from Osaka Electro-Communication University in 2004. He is a doctor course student of Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan.

Yuichi Kaji was born in Osaka, Japan, on December 23, 1968. He received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1991, 1992 and 1994, respectively. In 1994, he joined Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. In 2003 and 2004, he visited the University of California Davis and the University of Hawaii at Manoa as a visiting researcher. His current research interests include the theory of error correcting codes, fundamental techniques for information security, and the theory of automata and rewriting systems. He is a member of IPSJ, SITA and IEEE.