Noname manuscript No.
(will be inserted by the editor)

On Encoding Symbol Degrees of Array BP-XOR Codes

Maura B. Paterson · Douglas R. Stinson · Yongge Wang

Received: May 28, 2014 / First Revised: November 14, 2014 / Second Revised: April 7, 2015

Abstract Low density parity check (LDPC) codes, LT codes and digital fountain techniques have received significant attention from both academics and industry in the past few years. By employing the underlying ideas of efficient Belief Propagation (BP) decoding process (also called iterative message passing decoding process) on binary erasure channels (BEC) in LDPC codes, Wang has recently introduced the concept of array BP-XOR codes and showed the necessary and sufficient conditions for MDS $[k+2, k]$ and $[n, 2]$ array BP-XOR codes. In this paper, we analyze the encoding symbol degree requirements for array BP-XOR codes and present new necessary conditions for array BP-XOR codes. These new necessary conditions are used as a guideline for constructing several array BP-XOR codes and for presenting a complete characterization (necessary and sufficient conditions) of degree two array BP-XOR codes and for designing new edge-colored graphs. Meanwhile, these new necessary conditions are used to show that the codes by Feng, Deng, Bao, and Shen in IEEE Transactions on Computers are incorrect.

Key words: array codes; encoding symble degrees; MDS array codes; bounds on codes; error corecting codes

MSC Subject Classification: 94B65, 94A45, 94B05

Maura B. Paterson
Dept. Economics, Math. & Statistics
Birkbeck University of London
E-mail: m.paterson@bbk.ac.uk

Douglas R. Stinson
David R. Cheriton School of Computer Science
University of Waterloo, Canada
E-mail: dstinson@math.uwaterloo.ca

Yongge Wang
Dept. SIS, UNC Charlotte
Charlotte, NC 28223, USA
E-mail: yongge.wang@uncc.edu

1 Introduction

Low-density parity-check (LDPC) codes were invented by Gallager [14] in his PhD thesis. After being invented, they were largely forgotten and have been reinvented multiple times for the next 30 years (see, e.g., [33,34,1,21,22,20,32,28,24,25,23,2]). For example, based on expander graph results by Lubotzky, Phillips and Sarnak [19] and Margulis [27], Sipser and Spielman [33], Spielman [34], Alon et al. [1], and others introduced asymptotically linear LDPC error-correcting and erasure codes. Luby et al. [21,22] introduced LDPC Tornado codes, Luby [20] introduced LT-code, and Shokrollahi [32] introduced Raptor codes.

Array codes have been studied extensively for burst error correction in communication systems and storage systems (see, e.g., [4–7,10,18,42,43]). Array codes are linear codes where information and parity data are placed in a two dimensional matrix array. We first give a formal definition of the array BP-XOR codes. A $b \times n$ linear array code $\mathcal{C}$ over F_2 is a linear subspace of the vector space F_2^{nb}. If we regard the code $\mathcal{C}$ as a code over the alphabet F_2^b, where F_2^b denotes binary vectors of length b, then the minimum distance of $\mathcal{C}$ is defined as the minimum Hamming distance of the length n code over F_2^b. A linear array code $\mathcal{C}$ could be specified by a $bk \times bn$ matrix $G_{\mathcal{C}} = [G_1, G_2, \cdots, G_n]$ where each G_i is a $bk \times b$ binary matrix. If we use $(x_1, \cdots, x_{bk})$ to denote the binary information symbols, then we regard $(y_{i,1}, \cdots, y_{i,b}) = (x_1, \cdots, x_{bk})G_i$ as the ith column of $\mathcal{C}$. In other words, we could consider G_i as the generator matrix for the ith column of $\mathcal{C}$. By the above specification, we can alternatively regard the code $\mathcal{C}$ as a $b \times n$ matrix $\mathcal{C} = [a_{i,j}]_{1 \le i \le b, 1 \le j \le n}$ where $a_{i,j} \in \{0, 1\}$ are the encoding symbols and the ith column of $\mathcal{C}$ is generated by the generator matrix G_i.

A $b \times n$ linear array code $\mathcal{C}$ is called t-erasure tolerating $[n, k]$ array code if the information symbols $(x_1, \cdots, x_{bk})$ can be recovered from any $n - t$ columns of encoding symbols in the matrix $\mathcal{C}$. For an encoding symbol $a_{i,j} = x_{i_1} \oplus \cdots \oplus x_{i_\sigma}$, we call x_{i_j} $(1 \le j \le \sigma)$ a neighbor of $a_{i,j}$ and call σ the degree of $a_{i,j}$. A t-erasure tolerating $[n, k]$ $b \times n$ array code $\mathcal{C}$ is said to be maximum distance separable (MDS) if $k = n - t$.

The Belief Propagation decoding process (also called message passing iterative decoding) for binary symmetric channels (BSC) is present in Gallager [14] and is also used in artificial intelligence community [30]. The details of BP decoding process could be found in Cassuto and Shokrollahi [9]. In particular, Cassuto and Shokrollahi [9] presented a detailed discussion of BP decoding process for array codes. The reader is referred there for a formal definition of BP decoding process. The following is a high level informal description.

> (cited from [20]) “If there is at least one encoding symbol that has exactly one neighbor then the neighbor can be recovered immediately. The value of the recovered information symbol is XORed into any remaining encoding symbols that have this information symbol as a neighbor. The recovered information symbol is removed as a neighbor of these encoding symbols and the degree of each such encoding symbol is decreased by one to reflect this removal.”

Wang [37,38] recently studied array codes that could be decoded using BP decoding process: An $[n, k]$ array code $\mathcal{C} = [a_{i,j}]_{1 \le i \le b, 1 \le j \le n}$ is called a t-erasure tolerating $[n, k]$ array BP-XOR code if all information symbols $v_1, \cdots, v_{bk}$ can be recovered from any $n - t$ columns of the matrix using the BP-decoding process on the BEC.

In this paper, we analyze the encoding symbol degree requirements for array BP-XOR codes, present new necessary conditions for general array codes and array BP-XOR codes, and give a complete characterization of degree two BP- XOR codes. These necessary conditions are used as a guideline for constructing several array BP-XOR codes and the characterization of degree two BP-XOR codes are used to design new edge-colored graphs. Meanwhile, these necessary conditions are used to show that the codes by Feng, Deng, Bao, and Shen [11,12] are incorrect.

The structure of the paper is as follows. Section 2 establishes the degree requirements for weakly systematic array codes. Section 3 proves necessary conditions for the existence of array BP-XOR codes. Section 4 shows that the necessary conditions in Section 3 is sufficient for degree two encoding symbol based array BP-XOR codes. Bounds for high degree encoding symbol based array BP-XOR codes are briefly discussed in Section 5. Using the results in Section 2, Section 6 shows that the codes in [11] are incorrect.

2 Degree requirements for weakly systematic array codes

Let $\mathcal{C}$ be an MDS $b\times n$ array code with the $bk\times bn$ generator matrix $G_{\mathcal{C}} = [G_1, G_2, \cdots, G_n]$. An array code $\mathcal{C}$ is called *systematic* if there exist $1 \leq i_1, \cdots, i_k \leq n$ such that $[G_{i_1}, \cdots, G_{i_k}]$ is the $kb \times kb$ identity matrix I_{kb}. An array code $\mathcal{C}$ is called *weakly systematic* if there exists a $kb \times kb$ permutation matrix P such that $G_{\mathcal{C}}P = [I_{kb}|A_{\mathcal{C}}]$ where $A_{\mathcal{C}}$ is a $kb \times (n-k)b$ binary matrix.

A $bt \times bn$ binary matrix matrix H is said to be a parity-check matrix of a $b \times n$ array code $\mathcal{C}$ if we have $H\mathbf{y}^T = 0$ where $\mathbf{y} = (a_{1,1}, \cdots, a_{b,1}, \cdots, a_{1,n}, \cdots a_{b,n})$. By [26], we have the following proposition.

Proposition 1 *(MacWilliams and Sloane [26]) If $G_{\mathcal{C}} = [I_{kb}|A]$ is the generator matrix for a systematic array code $\mathcal{C}$, then $H_{\mathcal{C}} = [A^T|I_{(n-k)b}]$ is the parity check matrix for $\mathcal{C}$.*

By Proposition 1, it is straightforward to get the following proposition.

Proposition 2 *If $G_{\mathcal{C}} = [I_{kb}|A]P^{-1}$ is the generator matrix for a weakly systematic array code $\mathcal{C}$, then $H_{\mathcal{C}} = [A^T|I_{(n-k)b}]P^T$ is the parity check matrix for $\mathcal{C}$.*

Next we present a theorem on the minimal number of nonzero elements in the generator matrix of a weakly systematic array codes. It should be noted that Blaum and Roth [7, page 52, Proposition 3.4] presented similar results for systematic array codes.

Theorem 1 *For a weakly systematic $b \times n$ MDS array code $\mathcal{C}$ with generator matrix $G_{\mathcal{C}} = [G_1, G_2, \cdots, G_n] = [I_{kb}|A]P^{-1}$ and parity check matrix $H_{\mathcal{C}} = [A^T|I_{(n-k)b}]P^T$, each row of A contains at least $n-k$ nonzero elements and each column of A contains at least k nonzero elements.*

Proof. For a weakly systematic $b \times n$ MDS array code $\mathcal{C}$ with generator matrix $G_{\mathcal{C}} = [G_1, G_2, \cdots, G_n] = [I_{kb}|A]P^{-1}$ and parity check matrix $H_{\mathcal{C}} = [A^T|I_{(n-k)b}]P^T$, the information symbols could be recovered from any k columns of encoding symbols in the array code $\mathcal{C}$. For a contradiction, assume that there exists $i \in [1, kb]$, such that the ith row of A contains at most $n-k-1$ nonzero elements. That is, the ith row of $[I_{kb}|A]$ contains at most $n-k$ nonzero elements which are located in $G_{j_1}, \cdots, G_{j_{n-k}}$. In this case, the ith rows of the matrices $G_{j'_1}, \cdots, G_{j'_k}$ with $j'_i \neq j_1, \cdots, j_{n-k}$ are all zero rows. In other words, the $j'_1, \cdots, j'_k$-th columns of $\mathcal{C}$ contain no information about the information symbol x_i which means that $\mathcal{C}$ is not MDS. This is a contradiction.

The dual code of the weakly systematic $b \times n$ MDS array code $\mathcal{C}$ is a $b \times n$ MDS array code $\mathcal{C}^D$ with $H_{\mathcal{C}} = [A^T|I_{(n-k)b}]P^T$ as the generator matrix and all the information symbols could be recovered from any $n-k$ columns of encoding symbols in the array code

$\mathcal{C}^D$. Thus the similar argument as in the previous paragraph could be used to show that each row of A^T should have at least k nonzero elements. In other words, each column of A should have at least k nonzero elements. □

For a weakly systematic MDS $b \times n$ array code $\mathcal{C} = [a_{i,j}]$ with generator matrix $G_{\mathcal{C}} = [G_1, G_2, \cdots, G_n] = [I_{kb}|A]P^{-1}$, we have $a_{i,j} = (x_1, \cdots, x_{bk})\mathbf{b}_{ij}$ where $(x_1, \cdots, x_{bk})$ is the information symbol list and $\mathbf{b}_{ij}$ is a column vector from the matrix $G_{\mathcal{C}}$. Since $\mathbf{b}_{ij}$ contains either one single nonzero element or at least k nonzero elements (following Theorem 1), the degree of the encoding symbol $a_{i,j}$ is either one or $k' \geq k$. Our examples in Table 8 of Section 4 show some linear array codes with encoding symbols having degree less than k but larger than one.

3 Necessary Conditions on degrees of array BP-XOR codes

Wang [37,38] showed the equivalence between edge-colored graphs and array BP-XOR codes with degree two encoding symbols. In particular, degree two encoding symbols are sufficient to construct $[n, 2]$ MDS $b \times n$ array BP-XOR codes. Generally, we are interested in $[n, k]$ MDS $b \times n$ array BP-XOR codes for any $k < n$.

For an $[n, k]$ MDS $b \times n$ array BP-XOR code, we assume that there are bk information symbols, each of which is a variable that takes value from $M = \{0, 1\}^l$. The following theorem provides a necessary condition for the existence of array BP-XOR codes.

Theorem 2 *Let $\mathcal{C} = [a_{i,j}]_{1\leq i\leq b, 1\leq j\leq n}$ be an $[n, k]$ MDS $b \times n$ array BP-XOR code such that the degree of each encoding symbol $a_{i,j}$ is less than or equal to $\sigma < k+(k-1)/(b-1)$. Then we have*

$$n \leq k + \sigma - 1 + \left\lfloor \frac{\sigma(\sigma-1)(b-1)}{(k-\sigma)b + \sigma - 1} \right\rfloor \tag{1}$$

Proof. By the fact that there are $n - k$ column erasures, each information symbol must occur in at least $n-k+1$ columns. Since there are kb information symbols (data fragments) to encode, the total number of information symbol occurrences in the array BP-XOR code $\mathcal{C}$ is at least $kb(n - k + 1)$.

In order for the BP decoding process to work, we must start from a degree one encoding symbol. Thus we need to have at least $n - k + 1$ degree one encoding symbols in distinct columns of $\mathcal{C}$. This implies that we could use at most $bn - (n - k + 1)$ entries of the code to hold encoding symbols for degree two to σ. In other words, $\mathcal{C}$ contains at most $\sigma(bn - (n - k + 1)) + n - k + 1$ occurrences of information symbols. By the above fact, we must have

$$kb(n - k + 1) \leq \sigma(bn - (n - k + 1)) + n - k + 1.$$

By rearranging the terms, we get

$$kbn - kb(k - 1) \leq \sigma bn - (\sigma - 1)(n - k + 1).$$

If we move all terms to the right hand side and rewrite the inequality in terms of b and n, we get

$$k(k - 1)b - ((k - \sigma)b + (\sigma - 1))n + (\sigma - 1)(k - 1) \geq 0.$$

That is,

$$n((k - \sigma)b + \sigma - 1) \leq (k - 1)(kb + \sigma - 1). \tag{2}$$

By $\sigma < k + (k-1)/(b-1)$, we have $(k-\sigma)b + \sigma - 1 > 0$. Since n must be an integer, (2) implies (3)

$$\begin{aligned}
n &\leq \left\lfloor \frac{(k-1)(kb+\sigma-1)}{(k-\sigma)b+\sigma-1} \right\rfloor \\
&= \left\lfloor \frac{(k-\sigma)kb + k(\sigma-1) + (\sigma-1)kb - (\sigma-1)}{(k-\sigma)b+\sigma-1} \right\rfloor \\
&= k + \left\lfloor \frac{(\sigma-1)(kb-1)}{(k-\sigma)b+\sigma-1} \right\rfloor \\
&= k + \left\lfloor \frac{(\sigma-1)(kb-b\sigma+\sigma-1+b\sigma-\sigma)}{(k-\sigma)b+\sigma-1} \right\rfloor \\
&= k + \sigma - 1 + \left\lfloor \frac{\sigma(\sigma-1)(b-1)}{(k-\sigma)b+\sigma-1} \right\rfloor
\end{aligned} \tag{3}$$

Thus (1) holds. □

It is easy to see that the hypotheses of Theorem 2 are satisfied if $k \geq \sigma \geq 2$. So we have the following corollary.

Corollary 1 *Suppose that $k \geq \sigma \geq 2$. Then (1) holds.*

Next, we observe that equation (1) can be strengthened if $\sigma > 2$.

Theorem 3 *Suppose that $(k-\sigma)b+\sigma-1 > 0$, $\sigma > 2$, and $\sigma(\sigma-1)(b-1)/((k-\sigma)b+\sigma-1)$ is an integer. Then equality cannot hold in (1).*

Proof. If equality holds in (1), then the following conditions must be satisfied:

- There are $n-k+1$ encoding symbols having degree 1 and the remaining $bn-(n-k+1)$ encoding symbols all have degree σ.
- The encoding symbols of degree 1 occur in $n-k+1$ different columns of the array.

Suppose we choose k columns such that only one of these columns contains an encoding symbol of degree 1. Then within these k columns, all but one of the encoding symbols have degree 3 or greater. It therefore follows that the BP process cannot succeed. □

When $k = \sigma$, (1) can be simplified.

Corollary 2 *1. If $k = \sigma = 2$, then*

$$n \leq 2b + 1. \tag{4}$$

2. If $k = \sigma > 2$, then

$$n \leq kb + k - 2. \tag{5}$$

Proof. The equation (4) follows from (1). The equation (5) follows from Theorem 3. □

As an example, the code in Table 1 shows that the equality can hold in (4).

Table 1 array BP-XOR code for $b = 2, n = 5, k = 2, \sigma = 2$

v_1	v_2	v_3	v_4	$v_1 \oplus v_2$
$v_2 \oplus v_3$	$v_1 \oplus v_4$	$v_2 \oplus v_4$	$v_1 \oplus v_3$	$v_3 \oplus v_4$

4 Degree two MDS array BP-XOR codes and edge-colored graphs

By Corollary 2 and Theorem 3, Table 2 lists the the necessary upper bounds of n for the existence of $[n, k]$ MDS array BP-XOR codes with $\sigma = 2$. In this section, we give a complete characterization of degree two MDS array BP-XOR codes by showing that the bounds in Table 2 are sufficient. We first describe the edge-colored graph model by Wang and Desmedt [39,40]. The reader should be reminded that the edge-colored graph model in [39] is slightly different from the edge-colored graph definition in most papers. In most papers, the coloring of the edges is required to meet the condition that no two adjacent edges have the same color. This condition is not required in the definition of [39].

Table 2 Upper bounds of n for $[n, k]$ MDS array BP-XOR codes with $\sigma = 2$

k	2	3	3	$[4, \infty]$
n	$2b + 1$	4 if $b \leq 2$	5 if $b \geq 3$	$k + 1$

Definition 1 (Wang and Desmedt [39]) An edge-colored graph is a tuple $G = (V, E, C, f)$, with V the node set, E the edge set, C the color set, and f a map from E onto C. For any set $Z \subseteq E$, let $f(Z) = \{f(e) : e \in Z\}$. The structure

$$\mathcal{Z}_{C,t} = \{Z : Z \subseteq E \text{ and } |f(Z)| \leq t\}.$$

is called a *t-color* adversary structure. Let $A, B \in V$ be distinct nodes of G. A, B are called $(t+1)$-*color connected* for $t \geq 1$ if for any color set $C_t \subseteq C$ of size t, there is a path p from A to B in G such that the edges on p do not contain any color in C_t. An edge-colored graph G is $(t+1)$-*color connected* if and only if for any two nodes A and B in G, they are $(t+1)$-color connected.

In [37,38], Wang showed the equivalence of degree two encoding symbol based array BP-XOR codes and edge-colored graphs. In the following, we use an example to show how to convert an edge colored graph to an array BP-XOR codes. Figure 1 shows a 2-color connected edge-colored graph $G(V, E)$ with seven nodes, eight edges, and four colors. The edge-colored graph $G(V, E)$ in Figure 1 is also represented by the Table 3 where the edges with the same color are put in the same column. The Table 3 can be converted to a $[4, 3]$ MDS

Fig. 1 2-color connected edge-colored graph (the colors of edges are also shown in Table 3

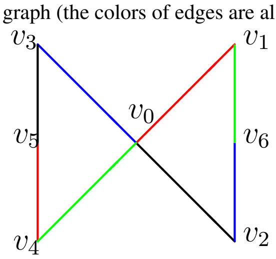

Table 3 Table representation of edge-colored graph in Figure 1

red	black	blue	green
(v_0, v_1)	(v_0, v_2)	(v_0, v_3)	(v_0, v_4)
(v_5, v_4)	(v_5, v_3)	(v_6, v_2)	(v_6, v_1)

array BP-XOR code in Table 4 by using each node variable to represent an information symbol and by converting each edge as the exclusive-or of the adjacent two information symbols (that is, the adjacent node variables). Note that to obtain the BP-decoding property, we remove all occurrence of the information symbol v_0 in the code of Table 4.

Table 4 Array BP-XOR code for $b = 2, n = 4, k = 3$ corresponding to edge-colored graph in Figure 1

v_1	v_2	v_3	v_4
$v_5 \oplus v_4$	$v_5 \oplus v_3$	$v_6 \oplus v_2$	$v_6 \oplus v_1$

4.1 $[n, 2]$ MDS array BP-XOR codes with $\sigma = 2$ from [37,13]

By Corollary 2, a necessary condition for the existence of $[n, 2]$ MDS array BP-XOR codes with $\sigma = 2$ is $n \leq 2b+1$. Wang [37,38] constructed $[n, 2]$ MDS $b \times n$ array BP-XOR codes with $n = 2b + 1$ using edge-colored graphs based on perfect one-factorization of complete graphs.

We first briefly review the construction of $[n, 2]$ MDS array BP-XOR codes in Wang [37,38]. Let p be a prime number with $n \leq p$. Using perfect one-factorization of K_{p+1}, Wang [37,38] constructed the $(p-1)$-color connected edge-colored graph in Table 5 where edges in the i-th column have the color c_i.

Table 5 $(p-1)$-color connected edge-colored graphs

(v_1, v_{p-1})	$\cdots$	(v_p, v_{p-2})
(v_2, v_{p-2})	$\cdots$	(v_1, v_{p-3})
$\cdots$	$\cdots$	$\cdots$
$(v_{(p-1)/2}, v_{(p+1)/2})$	$\cdots$	$(v_{(p-3)/2}, v_{(p-1)/2})$

The edge-colored graph in Table 5 is converted to the $b \times p$ array BP-XOR code in Table 6 by mapping each edge to a degree two encoding symbol and removing the occurrence of node v_p, and the $[n, 2]$ MDS $b \times n$ BP-XOR code is obtained by taking any of the n columns in Table 6, where $b = (p-1)/2$.

It should also be noted that the $(p-1)/2 \times p$ array BP-XOR code in Table 6 is equivalent to the code designed by Zaitsev, Zinov'ev, and Semakov [13] which was reformulated later as the dual of B-code in [42] using perfect one-factorization of complete graphs.

Table 6 $(p-1)/2 \times p$ BP-XOR code

$v_1 \oplus v_{p-1}$	$\cdots$	$v_{p-1} \oplus v_{p-3}$	v_{p-2}
$v_2 \oplus v_{p-2}$	$\cdots$	v_{p-4}	$v_1 \oplus v_{p-3}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
$v_b \oplus v_{b+1}$	$\cdots$	$v_{b-2} \oplus v_{b-1}$	$v_{b-1} \oplus v_b$

In the following sections, we show the construction of degree two $[n,k]$ MDS array BP-XOR codes and the corresponding edge-colored graphs for $2 < k < n$ when such kind of codes exist.

4.2 $[n,k]$ MDS array BP-XOR codes with $\sigma = 2$ and $n = k+1$

Wang and Desmedt [39] constructed the 2-color connected edge-colored cycle graph as described in the first row of Table 7. For $n = k+1$, the edge-colored graph in the first row of Table 7 could be used to obtain the $[n,k]$ MDS array BP-XOR codes with $\sigma = 2$ in the second row of Table 7.

Table 7 2-colored connected edge-colored graph and corresponding $[k+1,k]$ MDS array BP-XOR codes

(v_0, v_1)	(v_1, v_2)	$\cdots$	(v_{n-1}, v_n)	(v_n, v_0)
v_1	$v_1 \oplus v_2$	$\cdots$	$v_{n-1} \oplus v_n$	v_n

Based on the construction in Wang and Desmedt [39], one can obtain general $[k+1,k]$ MDS $b \times n$ array BP-XOR codes with $\sigma = 2$ by gluing together the v_0 nodes of b copies of edge-colored cycle graphs. For the example of $b = 2$ and $n = 4$, the array code in Table 4 is a $[4,3]$ MDS array BP-XOR code and the corresponding edge-colored graph is shown in Figure 1.

4.3 $[n,3]$ MDS array BP-XOR codes with $\sigma = 2$

By Theorem 1, there is no weakly systematic $[n,3]$ array BP-XOR codes for $\sigma = 2$. Theorem 2 shows that a necessary condition for the existence of $[n,3]$ MDS array BP-XOR codes with $\sigma = 2$ is $n = 4, b \geq 1$ or $n = 5, b \geq 3$.

For the case of $n = 4, b \geq 1$, the codes in Section 4.2 show that there exist $[4,3]$ MDS $b \times 4$ array BP-XOR codes.

For the case of $n = 5, b = 3$, Table 8 contains two $[5,3]$ MDS 3×5 array BP-XOR codes with $\sigma = 2$. The corresponding 3-color connected edge-colored graphs are shown in Figure 2 (removal of any two colors will not disconnect the graph).

The first graph in Figure 2 contains a four node cycle (v_4, v_5, v_9, v_8) while the second graph in Figure 2 does not contain any four node cycle. Thus the two $[5,3]$ MDS 3×5 array BP-XOR codes in Table 8 are not isomorphic.

Table 8 Two array BP-XOR codes for $b = 3, n = 5, k = 3$

v_1	$v_1 \oplus v_2$	$v_2 \oplus v_3$	v_7	v_3
$v_3 \oplus v_4$	$v_4 \oplus v_5$	$v_5 \oplus v_6$	$v_9 \oplus v_1$	$v_2 \oplus v_6$
$v_6 \oplus v_7$	$v_7 \oplus v_8$	$v_8 \oplus v_9$	$v_4 \oplus v_8$	$v_9 \oplus v_5$

v_1	v_2	v_3	$v_2 \oplus v_5$	$v_5 \oplus v_8$
$v_6 \oplus v_7$	$v_1 \oplus v_4$	$v_4 \oplus v_5$	$v_3 \oplus v_7$	$v_4 \oplus v_7$
$v_3 \oplus v_8$	$v_8 \oplus v_9$	$v_2 \oplus v_6$	$v_1 \oplus v_9$	$v_6 \oplus v_9$

Fig. 2 3-color connected edged-colored graphs (the colors of edges are also shown in Table 9)

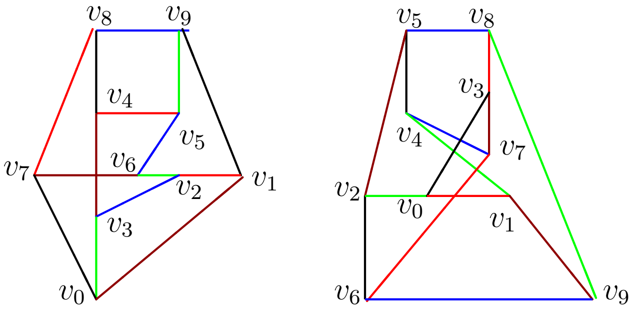

Table 9 Table representations of edge-colored graphs in Figure 2 with marked edge colors

brown	red	blue	black	green
(v_0, v_1)	(v_1, v_2)	(v_2, v_3)	(v_0, v_7)	(v_0, v_3)
(v_3, v_4)	(v_4, v_5)	(v_5, v_6)	(v_9, v_1)	(v_2, v_6)
(v_6, v_7)	(v_7, v_8)	(v_8, v_9)	(v_4, v_8)	(v_9, v_5)

red	green	black	brown	blue
(v_0, v_1)	(v_0, v_2)	(v_0, v_3)	(v_2, v_5)	(v_5, v_8)
(v_6, v_7)	(v_1, v_4)	(v_4, v_5)	(v_3, v_7)	(v_4, v_7)
(v_3, v_8)	(v_8, v_9)	(v_2, v_6)	(v_1, v_9)	(v_6, v_9)

For any integer $u \geq 1$, the graphs in Figure 2 could be used to construct 3-color connected edge-colored graphs with $9u + 1$ nodes, 5 colors, and $15u$ edges by gluing together the v_0 nodes of u copies of the graphs in Figure 2.

4.4 $[n, k]$ MDS array BP-XOR codes with $\sigma = 2$ and $k \geq 4$

By Theorem 1, there is no weakly systematic $[n, k]$ array BP-XOR codes for $\sigma = 2$ and $k \geq 4$. Theorem 2 shows that a necessary condition for the existence of $[n, k]$ MDS array BP-XOR codes with $\sigma = 2$ and $k \geq 4$ is $n \leq k + 1$. Since we also have $k < n$, it must be that $n = k + 1$. The codes in Section 4.2 show that there exist $[n, k]$ MDS $1 \times n$ array BP-XOR codes with $n = k + 1$ and $\sigma = 2$.

5 High degree MDS array BP-XOR codes

5.1 Upper bounds for higher degree MDS array BP-XOR codes

By Theorem 2, Theorem 3, and Corollary 2, Table 10 lists the upper bounds of n for the existence of $[n, k]$ MDS array BP-XOR codes with $\sigma = 3, 4, 5$. It should be noted that the upper bounds in Table 10 are obtained without any constraint on the values of b. In other words, we assume that b could take any values when necessary. When there are restrictions on the largest values that b could take, then Theorem 2 could be used to get stronger upper bounds on n. As an example, for $\sigma = 3, k = 4$, Theorem 2 gives $n \leq 12 - 18/(b + 2)$. When $b \geq 17$, this gives $n \leq 11$ which is the bound in the table. However, for $b < 17$, the upper bound on n will be smaller than 11. We should also mention that the bounds in Table 10 are upper bounds (necessary conditions). At present, it is not known whether any of these bounds could be achieved.

Table 10 Upper bounds of n for $[n, k]$ MDS array BP-XOR codes with $\sigma = 3, 4, 5$

$\sigma = 3$		$\sigma = 4$		$\sigma = 5$	
k	n	k	n	k	n
3	$3b + 1$	4	$4b + 2$	5	$5b + 3$
4	11	5	19	6	29
5	9	6	14	7	20
$[6, 8]$	$k + 3$	$[7, 8]$	13	8	18
$[9, \infty]$	$k + 2$	$[9, 15]$	$k + 4$	9	17
		$[16, \infty]$	$k + 3$	$[10, 11]$	$k + 7$
				$[12, 13]$	$k + 6$
				$[14, 24]$	$k + 5$
				$[25, \infty]$	$k + 4$

From Theorem 2, it is easy to show for any b and any $k \geq \sigma^2$ that the upper bound for the existence of $[n, k]$ MDS degree σ array BP-XOR codes is $n \leq k + \sigma - 1$.

5.2 Comparison with bounds for linear MDS codes

As mentioned in [7, Introduction], each $[n, k]$ MDS linear code over the finite field $GF(2^b)$ could be considered as an MDS $b \times n$ array code (not necessarily array BP-XOR code). However, the converse is not true (this follows the results in [18]). Table 11 lists some known maximum value of n (see, e.g., [15,31]) for the existence of $[n, k]$ MDS linear codes over $GF(2^b)$ with $b \geq 2$. For other values of $5 < k < 2^b - 1$, the well-known MDS conjecture

Table 11 Maximum value of n for $[n, k]$ MDS linear codes over $GF(2^b)$

k	2	3	4	5	$[2^b, \infty]$
n	$2^b + 1$	$2^b + 2$	$2^b + 1$	$2^b + 2$	$k + 1$

states that the maximum value for n is 2^b+1. For $k=2^b-1$, the MDS conjecture states that the maximum value for n is 2^b+2. This conjecture was proved to be true for $b \leq 4$. Furthermore, Bush [8] showed that $n \leq 2^b+k-1$ for $2 \leq k < 2^b$. This upper bound has been improved to $n \leq 2^b+k-3$ for $k \geq 4$ in [17] (see also [29]). Comparing the analysis in the previous sections and the values in Table 11, we see a big gap for the existence of MDS $b \times n$ array BP-XOR codes over $GF(2)$ and MDS linear codes over $GF(2^b)$.

6 MDS array codes with independent parity symbols

Blaum and Roth [6] introduced a general approach for constructing array codes using various slop diagonal redundancy. Using horizontal redundancy and 45 degree diagonal redundancy, Blaum, Brady, Bruck, and Menon [4] designed the celebrated EVEN-ODD codes that can tolerate double disk failures. EVEN-ODD codes were extended to tolerate three disk failures in Blaum, Bruck, and Vardy [5] and Huang and Xu [16]. One of the crucial ideas in these constructions is the use of imaginary rows in the array codes. In particular, Blaum, Bruck, and Vardy [5] gives a very good discussion on the roles of imaginary rows for array code design. The reader may also refer to Blaum [3] for a good summary of MDS array codes with minimal numbers of encoding operations.

Feng, Deng, Bao, and Shen [11,12] introduced extended Reed-Solomon MDS array codes to tolerate three column faults [11] and multiple (≥ 4) column faults [12] respectively. But the array codes in [11,12] cannot decode (thus they are not MDS) since they do not satisfy the minimal degree requirements for weakly MDS array codes of our Theorem 1.

We first give the reason why the codes in [11,12] are not MDS and then show that the example code in [11] cannot decode (thus it is not MDS). The following observation was communicated to us by Zhiying Wang [41]. The construction in [11] is similar to the construction in [5], where the symbols are not computed modulo the polynomial $M_p(x) = x^{p-1}+x^{p-2}+\cdots+x+1$. Thus the constructed codes are not MDS. To address this challenge, the authors in [5] used the following strategy:

> (Blaum, Bruck, and Vardy [5]) “It is also convenient to assume that the array has an imaginary row of zeros, which makes it a $p \times n$ array. A cyclic shift of a column in such array, that is, a multiplication by x modulo x^p-1, can cause the bit corresponding to the last row to be nonzero. However, in this case, the arithmetic modulo $M_p(x)$ forces to take the complement of the shifted column, restoring the zero in the last position”.

That is, when a parity bit in the imaginary row is 1, the whole parity column takes its complement. However, the authors of [11] simply throw away the parity bits in the imaginary row, which makes their codes non-MDS. Authors of [11] provided a formal proof for their MDS property. However, their decoding procedure does not take into account of the fact that their matrix $\tilde{I}^T$ throws away the imaginary row of the parity columns. Thus when the first parity and two information columns are erased, the code cannot be decoded. In order to illustrate the role of imaginary rows in the above discussion, we briefly describe the celebrated EVEN-ODD code which is sufficient for the reader to understand the critical role of the imaginary rows.

An EVEN-ODD code is a $(p-1) \times (p+2)$ array code $\mathcal{C} = [a_{i,j}]$ where p is a prime. The first p columns of $\mathcal{C}$ are information symbols, the $(p+1)$th column of $\mathcal{C}$ contains the horizontal redundancy, and the $(p+2)$th column of $\mathcal{C}$ contains the diagonal redundancy. In other words, for $1 \leq i \leq p-1$ and $1 \leq j \leq p$, $a_{i,j}$ are information symbols. In order to describe the parity columns, we assume that there is an imaginary 0-row after the last row in $\mathcal{C}$. With this convention, the array $\mathcal{C}$ is now a $p \times (p+1)$ array. We first introduce the notation $\langle \cdot \rangle_p$ where $\langle i \rangle_p = j$ if and only if $i = j \mod p$ and $1 \leq j \leq p$. For each l, $1 \leq l \leq p-1$,

$$a_{l,p+1} = a_{l,1} \oplus a_{l,2} \oplus \cdots \oplus a_{l,p} \tag{6}$$

and

$$a_{l,p+2} = S \oplus \left(\oplus_{t=1}^{p} a_{\langle l+1-t \rangle_p, t} \right) \tag{7}$$

where

$$S = \oplus_{t=2}^{p} a_{p+1-t,t} \tag{8}$$

Note that in the above definition, the diagonal redundancies $a_{l,p+2}$ are obtained by adding up the information symbols in various diagonals and then adding the bit S to it where S is given by the parity of the diagonal $(p-1,2), \cdots, (1,p)$. In this example of EVEN-ODD code, the imaginary 0-row and the symbol S are used to address the challenge that we have mentioned in the previous paragraph. It is straightforward to check that if we drop the imaginary 0-row and the symbol S in the equation (7), then the resulting "EVEN-ODD" code cannot decode since when the $(p+1)$th parity column and any information column are erased, then the code cannot decode.

In the following, we use the original example code in [11] to show this fact. In particular, we show that their example codes simply do not satisfy the degree requirements in our Theorem 1. Using circular permutation matrices as blocks, Vandermonde-like matrices are constructed as parity check matrices for extended Reed-Solomon codes to tolerate three column faults in [11]. In particular, the authors used a sequence of Example 2.1 [11, pages 1072-1073], Examples 2.2 [11, pages 1073], Examples 2.3 [11, pages 1074], Examples 3.1 [11, pages 1075], and Examples 3.2 [11, pages 1076] to show how to construct a 4×8 array codes to tolerate three column erasure. After the detailed code is constructed, a general decoding procedure is presented in [11, Section 4 on page 1076]. However, the authors in [11] did not try to decode their example code using their decoding procedure. Our above analysis shows their decoding process does not work. Thus it could not be used to decode their example code in Examples 3.2. In the following, we show that the codes in Examples 3.2 [11, pages 1076] will not decode at all. Indeed, since all the codes in [11] do not meet the degree requirements for general array codes in Theorem 1, these codes will not decode.

The parity check matrix in Examples 3.2 [11, pages 1076] is defined as $H = [I|A]$ where I is $4 \cdot 3 \times 4 \cdot 3$ (i.e., 12×12) identity matrix and A is the following $4 \cdot 5 \times 4 \cdot 5$ (i.e., 12×20) matrix.

$$A = \begin{bmatrix} 1000\ 1000\ 1000\ 1000\ 1000 \\ 0100\ 0100\ 0100\ 0100\ 0100 \\ 0010\ 0010\ 0010\ 0010\ 0010 \\ 0001\ 0001\ 0001\ 0001\ 0001 \\ \\ 1000\ 0000\ 0001\ 0010\ 0100 \\ 0100\ 1000\ 0000\ 0001\ 0010 \\ 0010\ 0100\ 1000\ 0000\ 0001 \\ 0001\ 0010\ 0100\ 1000\ 0000 \\ \\ 1000\ 0001\ 0100\ 0000\ 0010 \\ 0100\ 0000\ 0010\ 1000\ 0001 \\ 0010\ 1000\ 0001\ 0100\ 0000 \\ 0001\ 0100\ 0000\ 0010\ 1000 \end{bmatrix}$$

For the 4×8 array coded defined by the parity check matrix $H = [I|A]$, it is claimed that the code distance equals 4 (that is, $k = 5$) in [11]. That is, it will tolerate 3 column erasures. By Theorem 1, each column of $H = [I|A]$ should contain at least 3 nonzero elements. However, each of the columns in $7, 8, 9, 11, 14, 16, 17, 18$ contains 2 non-zero element. In other words, the code defined by the parity check matrix $H = [I|A]$ could not tolerate $n - k = 3$ erasure columns.

As an example, we show why the code could not be decoded. The code defined by the above parity check matrix $H = [I|A]$ could be represented in Table 12. It is straightforward to check that the variable v_7 only appears in columns $2, 6, 7$. Thus if we remove columns $2, 6$, and 7, then the variable v_7 could not be recovered from the remaining 5 columns (i.e., columns $1, 3, 4, 5, 8$). Similarly, each of the variables $v_8, v_9, v_{11}, v_{14}, v_{16}$, and v_{17} only appears in three columns. Thus these variables could not be recovered when the corresponding columns with their occurrences are missing.

Table 12 Array code for $b = 4, n = 8, k = 5$ in [11, Examples 3.2]

v_1	v_5	v_9	v_{13}	v_{17}	$v_1 \oplus v_5 \oplus v_9 \oplus v_{13} \oplus v_{17}$	$v_1 \oplus v_{12} \oplus v_{15} \oplus v_{18}$	$v_1 \oplus v_8 \oplus v_{10} \oplus v_{19}$
v_2	v_6	v_{10}	v_{14}	v_{18}	$v_2 \oplus v_6 \oplus v_{10} \oplus v_{14} \oplus v_{18}$	$v_2 \oplus v_5 \oplus v_{16} \oplus v_{19}$	$v_2 \oplus v_{11} \oplus v_{13} \oplus v_{20}$
v_3	v_7	v_{11}	v_{15}	v_{19}	$v_3 \oplus v_7 \oplus v_{11} \oplus v_{15} \oplus v_{19}$	$v_3 \oplus v_6 \oplus v_9 \oplus v_{20}$	$v_3 \oplus v_5 \oplus v_{12} \oplus v_{14}$
v_4	v_8	v_{12}	v_{16}	v_{20}	$v_4 \oplus v_8 \oplus v_{12} \oplus v_{16} \oplus v_{20}$	$v_4 \oplus v_7 \oplus v_{10} \oplus v_{13}$	$v_4 \oplus v_6 \oplus v_{15} \oplus v_{17}$

Similarly, the dual code of [11, Examples 3.2] in Table 12 is a 4×8 array code which is shown in Table 13. It is also straightforward to check that the code in Table 13 could not tolerate 5 column erasures. In other words, the original information symbols could not be recovered from any three columns. Specifically, each of the variables $v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}$, and v_{12} appears only in 5 columns. For example, v_5 only appears in columns $2, 4, 6, 7, 8$. Thus v_5 could not be recovered from columns $1, 3, 5$.

Table 13 Dual array code of [11, Examples 3.2] with $b = 4, n = 8, k = 3$

v_1	v_5	v_9	$v_1 \oplus v_5 \oplus v_9$	$v_1 \oplus v_6 \oplus v_{11}$	$v_1 \oplus v_7$	$v_1 \oplus v_8 \oplus v_{10}$	$v_1 \oplus v_{12}$
v_2	v_6	v_{10}	$v_2 \oplus v_6 \oplus v_{10}$	$v_2 \oplus v_7 \oplus v_{12}$	$v_2 \oplus v_8 \oplus v_9$	$v_2 \oplus v_{11}$	$v_2 \oplus v_5$
v_3	v_7	v_{11}	$v_3 \oplus v_7 \oplus v_{11}$	$v_3 \oplus v_8$	$v_3 \oplus v_{10}$	$v_3 \oplus v_5 \oplus v_{12}$	$v_3 \oplus v_6 \oplus v_9$
v_4	v_8	v_{12}	$v_4 \oplus v_8 \oplus v_{12}$	$v_4 \oplus v_9$	$v_4 \oplus v_5 \oplus v_{11}$	$v_4 \oplus v_6$	$v_4 \oplus v_7 \oplus v_{10}$

7 Conclusion

In this paper, we presented new upper bounds for the existence of $[n, k]$ MDS array BP-XOR codes and showed that these bounds could be achieved for $k = 2$. It is an open question to show that these bounds are also achievable for other values of $k \in [3, n)$.

Acknowledgment

The authors would like to thank the anonymous reviewers for detailed comments on improving the presentation of this paper.

References

1. N. Alon, J. Edmonds, and M. Luby. Linear time erasure codes with nearly optimal recovery. In *Proc. 36th FOCS*, pages 512–. IEEE Computer Society, 1995.
2. C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *Communications, IEEE Transactions on*, 44(10):1261–1271, 1996.
3. M. Blaum. A family of mds array codes with minimal number of encoding operations. In *Proc. IEEE ISIT 2006*, pages 2784–2788. IEEE, 2006.
4. M. Blaum, J. Brady, J. Bruck, and J. Menon. EVENODD: An efficient scheme for tolerating double disk failures in raid architectures. *IEEE Trans. Computers*, 44(2):192–202, 1995.
5. M. Blaum, J. Bruck, and E. Vardy. MDS array codes with independent parity symbols. *IEEE Trans. on Information Theory*, 42:529–542, 1996.
6. M. Blaum and R. M. Roth. New array codes for multiple phased burst correction. *IEEE Trans. on Information Theory*, 39(1):66–77, 1993.
7. M. Blaum and R. M. Roth. On lowest-density MDS codes. *IEEE Trans. on Information Theory*, 45:46–59, 1999.
8. K.A. Bush. Orthogonal arrays of index unity. *The Annals of Mathematical Statistics*, 23(3):426–434, 1952.
9. Y. Cassuto and A. Shokrollahi. Ldpc codes for 2d arrays. *Information Theory, IEEE Transactions on*, 60(6):3279–3291, June 2014.
10. Yuval Cassuto and Jehoshua Bruck. Cyclic lowest density mds array codes. *IEEE Trans. Inf. Theor.*, 55(4):1721–1729, April 2009.
11. G.L. Feng, R.H. Deng, F. Bao, and J.C. Shen. New efficient MDS array codes for RAID. Part I. Reed-Solomon-like codes for tolerating three disk failures. *IEEE Trans. Computers*, 54(9):1071–1080, 2005.
12. G.L. Feng, R.H. Deng, F. Bao, and J.C. Shen. New efficient MDS array codes for RAID. Part II. Rabin-like codes for tolerating multiple (≥ 4) disk failures. *IEEE Trans. Computers*, 54(12):1473–1483, 2005.
13. N. V. Semakov G. V. Zaitsev, V. A. Zinov'ev. Minimum-check-density codes for correcting bytes of errors, erasures, or defects. *Problems Inform. Transmission*, 19(3):197–204, 1983.
14. R. G. Gallager. *Low density Parity Check Codes*. MIT Press, 1963.
15. J.W.P. Hirschfeld, L. Storme, et al. The packing problem in statistics, coding theory and finite projective spaces: update 2001. *Developments in Mathematics*, 3:201–246, 2000.
16. C. Huang and L. Xu. STAR: an efficient coding scheme for correcting triple storage node failures. In *FAST*, pages 197–210, 2005.
17. S. Kounias and CI Petros. Orthogonal arrays of strength three and four with index unity. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 228–240, 1975.
18. E. Louidor and R. M. Roth. Lowest density MDS codes over extension alphabets. *IEEE Trans. Inf. Theor.*, 52(7):3186–3197, 2006.
19. A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.
20. M. Luby. LT codes. In *Proc. FOCS*, pages 271–280, 2002.
21. M. Luby, M. Mitzenmacher, M. Shokrollahi, D. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proc. 29th ACM STOC*, pages 150–159. ACM, 1997.
22. M. G. Luby, M. Mitzenmacher, and M. Amin Shokrollahi. Analysis of random processes via and-or tree evaluation. In *In Proc. 9th Annual ACM-SIAM SODA*, pages 364–373, 1998.
23. D. MacKay and R. Neal. Good codes based on very sparse matrices. *Cryptography and Coding*, pages 100–111, 1995.

24. D.J.C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Infor. Theory*, 45(2):399–431, 1999.
25. D.J.C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
26. F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. NH Pub. Company, 1978.
27. G. Margulis. Explicit construction of concentrators. *Probl. Inform. transm.*, 9:325–332, 1975.
28. G.A. Margulis. Explicit constructions of graphs without short cycles and low density codes. *Combinatorica*, 2(1):71–78, 1982.
29. MinT. Bound for oas with index unity. *http://mint.sbg.ac.at/desc_CBoundT0.html*, 2012.
30. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
31. R. Roth. *Introduction to coding theory*. Cambridge University Press, 2006.
32. A. Shokrollahi. Raptor codes. *IEEE Trans. on Inform. Theory*, 52(6):2551–2567, 2006.
33. M. Sipser and D. Spielman. Expander codes. *IEEE Trans. Infor. Theo.*, 42(6):1710–1722, 1996.
34. D.A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Information Theory*, 42(6):1723–1731, 1996.
35. Yongge Wang. Resource bounded randomness and computational complexity. *Theoret. Comput. Sci.*, 237:33–55, 2000.
36. Yongge Wang. Insecure "provably" secure network coding and homomorphic authentication sechemes for network coding. *Available at http://coitweb.uncc.edu/~yonwang/papers/ncattack.pdf*, 2010.
37. Yongge Wang. Array BP-XOR codes for reliable cloud storage systems. In *Proc IEEE ISIT 2013*, pages 326–330. IEEE Press, 2013.
38. Yongge Wang. Privacy-preserving data storage in cloud using array BP-XOR codes. *IEEE Trandactions on Cloud Computing*, 2015.
39. Yongge Wang and Yvo Desmedt. Edge-colored graphs with applications to homogeneous faults. *Inf. Process. Lett.*, 111(13):634–641, 2011.
40. Yongge Wang and Yvo Desmedt. Homogeneous faults, colored edge graphs, and cover free families. In *ICITS*, pages 58–72, 2011.
41. Zhiying Wang. Private communication. 2014.
42. L. Xu, V. Bohossian, J. Bruck, and D. Wagner. Low density mds codes and factors of complete graphs. *IEEE Trans. Inf. Theor.*, 45:1817–1826, 1998.
43. L. Xu and J. Bruck. X-code: Mds array codes with optimal encoding. *IEEE Trans. on Information Theory*, 45:272–276, 1999.