SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2018, Vol. 61 100302:1–100302:11
https://doi.org/10.1007/s11432-018-9485-7

Special Focus on Distributed Storage Coding

A class of binary MDS array codes with asymptotically weak-optimal repair†

Hanxu HOU* & Yunghsiang S. HAN

School of Electrical Engineering & Intelligentization, Dongguan University of Technology, Dongguan 523808*, China*

Received 11 March 2018/Revised 27 April 2018/Accepted 12 June 2018/Published online 15 August 2018

Abstract Binary maximum distance separable (MDS) array codes contain k information columns and r parity columns in which each entry is a bit that can tolerate r arbitrary erasures. When a column in an MDS code fails, it has been proven that we must download at least half of the content from each helper column if $k+1$ columns are selected as the helper columns. If the lower bound is achieved such that the $k+1$ helper columns can be selected from any $k+r-1$ surviving columns, then the repair is an optimal repair. Otherwise, if the lower bound is achieved with $k+1$ specific helper columns, the repair is a weak-optimal repair. This paper proposes a class of binary MDS array codes with $k \geqslant 3$ and $r \geqslant 2$ that asymptotically achieve weak-optimal repair of an information column with $k+1$ helper columns. We show that there exist many encoding matrices such that the corresponding binary MDS array codes can asymptotically achieve weak-optimal repair for repairing any information column.

Keywords MDS codes, binary MDS array codes, optimal repair, encoding matrix, asymptotically weak-optimal repair

Citation Hou H X, Han Y S. A class of binary MDS array codes with asymptotically weak-optimal repair. Sci China Inf Sci, 2018, 61(10): 100302, https://doi.org/10.1007/s11432-018-9485-7

1 Introduction

Many storage systems employ array codes to enhance data reliability against failures of storage nodes with a low degree of redundancy. Binary maximum distance separable (MDS) array codes are an important family of array codes with a trade-off between storage redundancy and fault tolerance, where each entry is a bit. In a binary MDS array code, there are k information columns and r parity columns such that all the information columns can be recovered from any k columns. Herein, we use the terms column and node interchangeably.

The literature contains many binary MDS array codes. EVENODD [1] and RDP [2] are two important binary MDS array codes with two parity columns (i.e., $r = 2$). Other binary MDS array codes include STAR codes [3], generalized RDP codes [4], generalized EVENODD codes [5,6], and Rabin-like codes [7,8], all of which have more parity columns (i.e., $r \geqslant 3$).

When a column fails, it should be recovered to maintain the same level of reliability. The amount of bits downloaded in repairing a failed column is termed the repair bandwidth. It is important to reduce the repair bandwidth in data centers or distributed storage systems. Dimakis et al. [9] studied the repair problem by using network coding theory. The optimal repair bandwidth of MDS codes [9] is $\frac{dL}{d-k+1}$,

* Corresponding author (email: houhanxu@163.com)
† Invited paper

where L is the number of bits stored in each column and $k \leqslant d \leqslant k+r-1$. When $d = k+1$, the optimal repair bandwidth becomes

$$\frac{(k+1)L}{2}. \tag{1}$$

We use the lower bound in (1) to distinguish between optimal repair and weak-optimal repair according to [10]. A repair is an optimal repair if the lower bound in (1) is achieved such that the $k+1$ helper nodes can be chosen from any other $k+r-1$ surviving nodes. If the lower bound is achieved by downloading bits from $k+1$ specific nodes, the repair is a weak-optimal repair. There are many constructions [9, 11–15] of MDS codes with the minimum repair bandwidth over a sufficiently large finite field. However, there have been relatively few studies of binary MDS array codes with optimal repair bandwidths [10, 16–20].

Some repair methods have been proposed for reducing the repair bandwidths of RDP codes [21], X-code [22], and EVENODD [23]. However, the repair bandwidth is still 50% larger than the optimal repair bandwidth given in (1). There are some constructions of binary MDS array codes [10, 16–20] with small repair bandwidths. The authors in [16] proposed binary MDS array codes with two parity columns, which achieve an optimal repair bandwidth for information-column failure. The codes given in [16] are optimized and known as ButterFly codes [17]. MDR codes [19, 20] are also binary MDS array codes with $r = 2$ but with optimal repair only for k information columns and one parity column. Binary MDS array codes with three parity columns proposed in [18] have asymptotically weak-optimal repair bandwidths for information-column failure. Note that the failed column in all the above studies is repaired using $k+1$ helper columns. Constructions of binary MDS array codes with more parity columns and asymptotically weak-optimal repair were given in [10], where the parameters satisfy $k+1 \leqslant d \leqslant k+\lfloor (r-1)/2 \rfloor$.

In the present study, we focus on constructing binary MDS array codes with any number of parity columns and asymptotically weak-optimal repair. By exploiting the essential property of an encoding matrix over the quotient ring proposed in [10], we observe that there exist many encoding matrices in the quotient ring such that the constructed codes have asymptotically weak-optimal repair bandwidths for any information-column failure. We characterize the property of the encoding matrix and show that our binary MDS array codes can achieve the weak-optimal repair bandwidth in (1) asymptotically for any failed information column when k is sufficiently large. In a repair procedure, we must download many bits to repair all the bits in the failed column. If some downloaded bits used to repair different bits in the failed column are the same, we need only download these bits once and the repair bandwidth can be reduced. In the proposed codes, the optimal repair bandwidth is achieved by choosing some encoding matrices in which the bits downloaded for repairing a failed column intersect with each other as much as possible.

The differences between the codes proposed herein and the existing binary MDS array codes with weak-optimal repair are as follows. In contrast to existing constructions with two parity columns such as those given in [16, 17, 19, 20], a quotient ring with cyclic structure is employed in the proposed construction. Although both the proposed binary MDS array codes and other binary MDS array codes [10, 18] with asymptotically weak-optimal repair employ the quotient ring $\mathbb{F}_2[x]/(1+x^{p\tau})$, their main results are different. In [18], an encoding matrix is chosen to form a binary MDS array code with three parity columns over $\mathbb{F}_2[x]/(1+x^{p2^{k-2}})$. In [10], a general approach of designing binary array codes over $\mathbb{F}_2[x]/(1+x^{p\tau})$ is presented and two explicit constructions based on the design approach are proposed. The first construction of [10] has an odd number of parity columns and asymptotically weak-optimal repair for k information columns, while the second construction of [10] has an even number of parity columns and asymptotically weak-optimal repair for all $k+r$ columns. In the present study, based on the design approach in [10], we focus on $d = k+1$ and explore the property of the encoding matrix such that the corresponding codes have asymptotically weak-optimal repair. We show the required property of the encoding matrix and propose that many encoding matrices satisfy this property. The binary MDS array codes proposed in [18] can be viewed as a special case of the binary MDS array codes proposed herein.

2 Sufficient condition of encoding matrices for asymptotically weak-optimal repair

In this section, we first review the design approach of binary MDS array codes proposed in [10]. We thereafter provide a sufficient condition for encoding matrices that can enable asymptotically weak-optimal repair of information columns.

2.1 Review of binary array codes given in [10]

The design approach in [10] constructs an array code of size $(p-1)\tau \times (k+r)$, where $k \geqslant 3, r \geqslant 2$, p is a prime such that 2 is a primitive element in the field $\mathbb{F}_p$, and τ is an integer that will be specified later for choosing the encoding matrix. In the $(p-1)\tau \times (k+r)$ array, the first k columns are information columns and the last r columns are parity columns. The i-th entry of column j is denoted by $a_{i,j}$ for $i = 0, 1, \ldots, (p-1)\tau$ and $j = 1, 2, \ldots, k+r$. Note that we consider modulo $p\tau$ of the subscripts throughout unless stated otherwise.

We must append τ extra bits to each column to represent the proposed array over the quotient ring $\mathbb{F}_2[x]/(1+x^{p\tau})$, with the extra bit $a_{(p-1)\tau+\mu,j}$ defined as

$$a_{(p-1)\tau+\mu,j} = \sum_{i=0}^{p-2} a_{i\tau+\mu,j}, \tag{2}$$

for information column j, where $j = 1, 2, \ldots, k$ and $\mu = 0, 1, \ldots, \tau-1$. We also append τ extra bits $a_{(p-1)\tau,j}, a_{(p-1)\tau+1,j}, \ldots, a_{p\tau-1,j}$ to parity column $j-k$ during the encoding procedure, where $j = k+1, k+2, \ldots, k+r$. We will show in the encoding process that the appended extra bit $a_{(p-1)\tau+\mu,j}$ also satisfies (2) for $j = k+1, k+2, \ldots, k+r$ and $\mu = 0, 1, \ldots, \tau-1$.

The $p\tau$ bits $a_{0,j}, a_{1,j}, \ldots, a_{p\tau-1,j}$ are represented by a polynomial $a_j(x)$ over $\mathbb{F}_2[x]$:

$$a_j(x) = a_{0,j} + a_{1,j}x + a_{2,j}x^2 + \cdots + a_{p\tau-1,j}x^{p\tau-1},$$

where $j = 1, 2, \ldots, k+r$. For $j = 1, 2, \ldots, k$, $a_j(x)$, which corresponds to information column j, is called a data polynomial. Similarly, $a_j(x)$ for $j = k+1, k+2, \ldots, k+r$, which corresponds to parity column $j-k$, is called a coded polynomial. The encoding of the array codes can be described as the product

$$[a_1(x), a_2(x), \ldots, a_{k+r}(x)] = [a_1(x), a_2(x), \ldots, a_k(x)] \cdot \boldsymbol{G},$$

over the quotient ring $R_{p\tau} = \mathbb{F}_2[x]/(1+x^{p\tau})$. The generator matrix $\boldsymbol{G}$ is of size $k \times (k+r)$ and is composed of the $k \times k$ identity matrix $\boldsymbol{I}_{k\times k}$ and a $k \times r$ encoding matrix $\boldsymbol{P}_{k\times r}$. Thus, the parity columns can be determined by choosing a proper encoding matrix $\boldsymbol{P}_{k\times r}$.

In $R_{p\tau}$, the variable x represents the cyclic-right-shift operator on a polynomial. The cyclic structure and a well-designed encoding matrix are crucial for reducing the repair bandwidth for the failure of a single information column. Note that the extra bits need not be stored in a storage node; we can calculate them when necessary.

Consider a sub-ring $C_{p\tau}$ of $R_{p\tau}$, which consists of polynomials in $R_{p\tau}$ with $1+x^\tau$ being a factor, namely

$$C_{p\tau} = \{a(x)(1+x^\tau) \bmod (1+x^{p\tau}) \,|\, a(x) \in R_{p\tau}\}. \tag{3}$$

Lemma 1 ([10, Theorem. 1])**.** The coefficients of polynomial $a_i(x)$ satisfy (2) if and only if $a_i(x) \in C_{p\tau}$.

According to Lemma 1, there are k data polynomials in $C_{p\tau}$. In the encoding procedure, after appending τ extra bits to each information column, we obtain k data polynomials in $C_{p\tau}$. Subsequently, the r coded polynomials are determined by $\boldsymbol{P}_{k\times r}$. Therefore, the key issue is to design a proper encoding matrix. We subsequently exploit the property of the encoding matrices such that the corresponding codes are MDS codes and the repair bandwidth of a single information column is asymptotically weak-optimal.

2.2 Design of encoding matrix

Furthermore, we present a sufficient condition for designing encoding matrices that can achieve asymptotically weak-optimal repair bandwidths. We begin by considering the case of array codes with two parity columns. A sufficient condition for designing an encoding matrix that can achieve asymptotically weak-optimal repair of all the information columns is given in the following theorem.

Theorem 1. Consider the encoding matrix

$$\boldsymbol{P}_{k\times 2}=\begin{bmatrix}1 & 1 & \cdots & 1\\ x^{e_1} & x^{e_2} & \cdots & x^{e_k}\end{bmatrix}^{\mathrm{T}}. \tag{4}$$

Let $e_f\neq 0$ for $f=1,2,\ldots,k$. If τ is a multiple of $2e_f$ for $f=1,2,\ldots,k$ and e_i is a multiple of $2e_{i-1}$ for $i=2,3,\ldots,k$, then there exists a repair algorithm with $d=k+1$ such that the repair bandwidth of column f for $f=1,2,\ldots,k$ is

$$k(p-1)\tau/2+(p-1)\tau/2+\frac{(p-1)\tau}{2e_f}\cdot\left(\sum_{i=1}^{f-1}e_i\right), \tag{5}$$

which is asymptotically weak-optimal when k is sufficiently large.

Proof. According to Lemma 1, we obtain $a_i(x)\in C_{p\tau}$ for $i=1,2,\ldots,k$. Because $x^{\ell}a_i(x)\in C_{p\tau}$ for $i=1,2,\ldots,k$ and any integer ℓ, we obtain $a_{k+1}(x)=\sum_{i=1}^{k}a_i(x)\in C_{p\tau}$ and $a_{k+2}(x)=\sum_{i=1}^{k}x^{e_i}a_i(x)\in C_{p\tau}$, i.e., both coded polynomials are in $C_{p\tau}$. Thus, the coefficients of $a_{k+1}(x)$ and $a_{k+2}(x)$ satisfy (2).

Furthermore, we assume that column f fails, where $1\leqslant f\leqslant k$, and we will provide a repair algorithm such that the repair bandwidth is asymptotically weak-optimal. Because the coefficients of $a_i(x)$ satisfy (2), where $i=1,2,\ldots,k+2$, we denote the bits of column i as all $p\tau$ bits, as we can calculate the extra bits using (2) when necessary.

According to the encoding matrix $\boldsymbol{P}_{k\times 2}$, the bits in columns $k+1$ and $k+2$ are computed as follows:

$$\begin{aligned}&a_{\ell,1}+a_{\ell,2}+\cdots+a_{\ell,k}=a_{\ell,k+1},\\ &a_{\ell-e_1,1}+a_{\ell-e_2,2}+\cdots+a_{\ell-e_k,k}=a_{\ell,k+2},\end{aligned}$$

where $\ell=0,1,\ldots,p\tau-1$. Recall that herein we consider modulo $p\tau$ of all the subscripts. From the above equations, we can repair the bit $a_{\ell,f}$ using either

$$a_{\ell,f}=a_{\ell,1}+a_{\ell,2}+\cdots+a_{\ell,f-1}+a_{\ell,f+1}+\cdots+a_{\ell,k}+a_{\ell,k+1}, \tag{6}$$

or

$$a_{\ell,f}=a_{\ell+e_f-e_1,1}+a_{\ell+e_f-e_2,2}+\cdots+a_{\ell+e_f-e_{f-1},f-1}+a_{\ell+e_f-e_{f+1},f+1}+\cdots+a_{\ell+e_f-e_k,k}+a_{\ell+e_f,k+2}. \tag{7}$$

If

$$\ell \bmod 2e_f\in\{0,1,\ldots,e_f-1\}, \tag{8}$$

and $\ell\in\{0,1,\ldots,(p-1)\tau-1\}$, we recover the bits $a_{\ell,f}$ using (6), i.e., by downloading $a_{\ell,i}$ for $i=1,\ldots,f-1,f+1,\ldots,k+1$ with ℓ satisfying (8). If

$$\ell \bmod 2e_f\in\{e_f,e_f+1,\ldots,2e_f-1\}, \tag{9}$$

and $\ell\in\{0,1,\ldots,(p-1)\tau-1\}$, we recover the bits $a_{\ell,f}$ using (7), i.e., by downloading $a_{\ell+e_f-e_i,i}$ for $i=1,\ldots,f-1,f+1,\ldots,k$ and $a_{\ell+e_f,k+2}$ with ℓ satisfying (9).

Because τ is a multiple of $2e_f$, $(p-1)\tau$ is also a multiple of $2e_f$ and $\ell \bmod 2e_f$ is uniformly distributed over $\{0,1,\ldots,2e_f-1\}$. Thus, $(p-1)\tau/2$ bits are recovered using (6) and another $(p-1)\tau/2$ bits are recovered using (7). Because

$$\{0,1,\ldots,e_f-1\}\cap\{e_f,e_f+1,\ldots,2e_f-1\}=\emptyset,$$

all $(p-1)\tau$ information bits $a_{\ell,f}$ can be recovered using the above method. In the repair procedure, we can first download $k(p-1)\tau/2$ bits $a_{\ell,i}$ for $i=1,\ldots,f-1,f+1,\ldots,k+1$ and ℓ satisfying (8) to recover the corresponding $(p-1)\tau/2$ bits $a_{\ell,f}$. To recover another $(p-1)\tau/2$ bits $a_{\ell,f}$ with ℓ satisfying (9), the corresponding bits $a_{\ell+e_f-e_i,i}$ for $i=1,\ldots,f-1,f+1,\ldots,k$ and $a_{\ell+e_f,k+2}$ are required. Thus, we show that we do not need to download all $k(p-1)\tau/2$ bits $a_{\ell+e_f-e_i,i}$ and $a_{\ell+e_f,k+2}$, with ℓ satisfying (9) and $i=1,\ldots,f-1,f+1,\ldots,k$, because many of them have been downloaded in repairing $(p-1)\tau/2$ bits $a_{\ell,f}$ with ℓ satisfying (8).

We first consider the bits $a_{\ell+e_f-e_i,i}$ for $i=1,\ldots,f-1$ and ℓ satisfying (9). If $\ell \bmod 2e_f = e_f$, then there exists an integer m such that $\ell = m\cdot 2e_f + e_f$. Recall that the indices of the required bits $a_{\ell+e_f-e_i,i}$ are $\ell' = (\ell+e_f-e_i) \bmod p\tau$. We have

$$\begin{aligned}\ell' \bmod 2e_f &= ((\ell+e_f-e_i) \bmod p\tau) \bmod 2e_f\\ &= (m\cdot 2e_f+e_f+e_f-e_i) \bmod 2e_f \text{ (as } \tau \text{ is a multiple of } 2e_f)\\ &= 2e_f-e_i \text{ (as } e_f \text{ is a multiple of } 2e_i \text{ for } i=1,2,\ldots,f-1 \text{ and } e_f\neq 0).\end{aligned}$$

If $\ell \bmod 2e_f = 2e_f-1$, then there exists an integer m such that $\ell = m\cdot 2e_f + 2e_f - 1$. Because the indices of the required bits $a_{\ell+e_f-e_i,i}$ are $\ell' = (\ell+e_f-e_i) \bmod p\tau$, we have

$$\begin{aligned}\ell' \bmod 2e_f &= ((\ell+e_f-e_i) \bmod p\tau) \bmod 2e_f\\ &= (m\cdot 2e_f+2e_f-1+e_f-e_i) \bmod 2e_f \text{ (as } \tau \text{ is a multiple of } 2e_f)\\ &= e_f-e_i-1 \text{ (as } e_f \text{ is a multiple of } 2e_i \text{ for } i=1,2,\ldots,f-1 \text{ and } e_f\neq 0).\end{aligned}$$

By repeating the above procedure for $\ell \bmod 2e_f = e_f+1,\ldots,2e_f-2$, we can obtain

$$\ell' \bmod 2e_f = 2e_f-e_i, 2e_f-e_i+1,\ldots,2e_f-1,0,1,\ldots,e_f-e_i-1, \tag{10}$$

when $\ell \bmod 2e_f$ runs from e_f to $2e_f-1$. Thus, we require the bits $a_{\ell',i}$ with $i=1,2,\ldots,f-1$ and ℓ' in (10). Because $e_f-e_i-1<2e_f-e_i$, the elements in (10) can be rearranged as

$$\ell' \bmod 2e_f = 0,1,\ldots,e_f-e_i-1,2e_f-e_i,2e_f-e_i+1,\ldots,2e_f-1. \tag{11}$$

Recall that $k(p-1)\tau/2$ bits $a_{\ell,i}$ for $i=1,\ldots,f-1,f+1,\ldots,k+1$ and ℓ satisfying (8) have already been downloaded. Because $e_f-e_i-1\leqslant e_f-1<2e_f-e_i$,

$$\{0,1,\ldots,e_f-e_i-1,2e_f-e_i,2e_f-e_i+1,\ldots,2e_f-1\}\backslash\{0,1,\ldots,e_f-1\} = \{2e_f-e_i,2e_f-e_i+1,\ldots,2e_f-1\}.$$

We thus need only download $e_i\frac{(p-1)\tau}{2e_f}$ information bits $a_{\ell',i}$ from columns i for $i=1,2,\ldots,f-1$ with $\ell' \bmod 2e_f \in \{2e_f-e_i,2e_f-e_i+1,\ldots,2e_f-1\}$.

We thereafter consider the bits $a_{\ell+e_f-e_i,i}$ for $i=f+1,f+2,\ldots,k$ and ℓ satisfying (9). We can express ℓ satisfying (9) as $\ell = m\cdot 2e_f+e_f+t$, where m is an integer and $t=0,1,\ldots,e_f-1$. Recall that the indices of the required bits $a_{\ell+e_f-e_i,i}$ are $\ell' = (\ell+e_f-e_i) \bmod p\tau$. We have

$$\begin{aligned}\ell' \bmod 2e_f &= ((\ell+e_f-e_i) \bmod p\tau) \bmod 2e_f\\ &= (m\cdot 2e_f+e_f+t+e_f-e_i) \bmod 2e_f \text{ (as } \tau \text{ is a multiple of } 2e_f)\\ &= t \text{ (as } e_i \text{ is a multiple of } e_f \text{ for } i=f+1,f+2,\ldots,k).\end{aligned}$$

Recall that $(k-1)(p-1)\tau/2$ bits $a_{\ell,i}$ for $i=f+1,f+2,\ldots,k$ and ℓ satisfying (8) have been downloaded. Therefore, there is no need to download the bits $a_{\ell',i}$ for $i=f+1,f+2,\ldots,k$ and $\ell' \bmod 2e_f \in \{0,1,\ldots,e_f-1\}$. For column $k+2$, we must download $(p-1)\tau/2$ parity bits $a_{\ell,k+2}$ with ℓ satisfying (9).

Therefore, we must download $k(p-1)\tau/2+(p-1)\tau/2+\frac{(p-1)\tau}{2e_f}\cdot(\sum_{i=1}^{f-1}e_i)$ bits from $k+1$ columns to recover column f. Note that the value in (5) is strictly less than $(k+2)(p-1)\tau/2$ because e_i is a multiple of $2e_{i-1}$ and $e_i\neq 0$ for $i=2,3,\ldots,f$. In other words, the repair bandwidth of column f is strictly less than $\frac{k+2}{k+1}$ times the value in (1). Therefore, the repair bandwidth of column f can achieve weak-optimal repair in (1) asymptotically when k is sufficiently large, where $1\leqslant f\leqslant k$. This completes the proof.

There are many selections of e_i for $i = 1, 2, \ldots, k$ such that the condition given in Theorem 1 is satisfied: for example, $e_i = 2^{i-1}$ for $i = 1, 2, \ldots, k$ and $\tau = 2^k$. Because $e_1 \neq 0$, the minimum positive integer e_1 is 1. For $i = 2, 3, \ldots, k$, e_i is a multiple of $2e_{i-1}$ and $e_i \neq 0$; thus, we have the minimum value of e_i as $e_i = 2^{i-1}$. Recall that τ should be a multiple of $2e_i$ for $i = 1, 2, \ldots, k$ and the minimum positive value of e_i is 2^{i-1}. Therefore, the minimum value of τ is 2^k.

From Theorem 1, we can use two columns of the encoding matrix to recover each of the first k columns with asymptotically weak-optimal repair. Furthermore, we consider the repair with r parity columns. We divide the k information columns into several groups, each of which contains some columns. We intend to recover each column in one group by using two column vectors of the $k \times r$ encoding matrix that satisfy the condition given in Theorem 1. By carefully choosing the $k \times r$ encoding matrix, we can recover each of the k information columns that can achieve the weak-optimal repair asymptotically.

Let $\boldsymbol{e}$ denote the vector

$$\boldsymbol{e} = \left[x^{e_1} \ x^{e_2} \ \cdots \ x^{e_k}\right].$$

We define the right cyclic shift of the vector $\boldsymbol{e}$ by i positions as

$$\boldsymbol{e}(i) = \left[x^{e_{k-i+1}} \ x^{e_{k-i+2}} \ \cdots \ x^{e_k} \ x^{e_1} \ x^{e_2} \ \cdots \ x^{e_{k-i}}\right],$$

where $i = 1, 2, \ldots, k$. For example, when $i = 1$,

$$\boldsymbol{e}(1) = \left[x^{e_k} \ x^{e_1} \ \cdots \ x^{e_{k-1}}\right].$$

If $i = k$, then $\boldsymbol{e}(k)$ is $\boldsymbol{e}$ itself. Based on the above notation, we choose the $k \times r$ encoding matrix $\boldsymbol{P}_{k\times r}$ to be

$$\boldsymbol{P}_{k\times r} = \left[\boldsymbol{I}_k^{\mathrm{T}} \ \boldsymbol{e}^{\mathrm{T}} \ \boldsymbol{e}(\lfloor \tfrac{k}{r-1} \rfloor)^{\mathrm{T}} \ \boldsymbol{e}(2\lfloor \tfrac{k}{r-1} \rfloor)^{\mathrm{T}} \ \cdots \ \boldsymbol{e}((r-2)\lfloor \tfrac{k}{r-1} \rfloor)^{\mathrm{T}}\right], \tag{12}$$

where $\boldsymbol{I}_k$ is an all-one row vector of length k. We show in the following theorem that the binary MDS array codes with the encoding matrix in (12) have weak-optimal repair of information columns asymptotically. Throughout the paper, we will use the notation $t = k - (r-1)\lfloor \frac{k}{r-1} \rfloor$.

Theorem 2. The $k \times r$ encoding matrix $\boldsymbol{P}_{k\times r}$ in (12) is considered, where $0 < e_1$ and τ is a multiple of $2e_i$ for $i = 1, 2, \ldots, \lfloor \frac{k}{r-1} \rfloor + t$. For $i = 1, 2, \ldots, \lfloor \frac{k}{r-1} \rfloor + t$, if e_j is a multiple of $2e_i$ for $j = i+1, i+2, \ldots, k$, then there exists a repair algorithm such that the repair bandwidth of the k information columns is asymptotically weak-optimal when $d = k + 1$ and k is sufficiently large.

Proof. Assume that column f fails, where $1 \leqslant f \leqslant k$. When we state that a bit $a_{\ell,f}$ is repaired using the encoding column i, where $i = 1, 2, \ldots, r$, it indicates that we download all the bits determined using the encoding column i except the erased bit $a_{\ell,f}$. For example, the bit $a_{\ell,f}$ can be repaired using the first encoding column, i.e., by downloading bits $a_{\ell,i}$ for $i = 1, 2, \ldots, , f-1, f+1, \ldots, k+1$.

Algorithm 1 Algorithm for repairing the failure of a single information column

1: The information column f fails.
2: **if** $f \in \{1, 2, \ldots, (r-2)\lfloor \frac{k}{r-1} \rfloor\}$ **then**
3: Repair the bit $a_{\ell,f}$ using the first encoding column, for $\ell \mod 2e_{1+(f-1 \bmod \lfloor \frac{k}{r-1} \rfloor)} \in \{0, 1, \ldots, e_{1+(f-1 \bmod \lfloor \frac{k}{r-1} \rfloor)} - 1\}$. Otherwise, repair the bit $a_{\ell,f}$ using the encoding column $1 + \lceil \frac{f}{\lfloor \frac{k}{r-1} \rfloor} \rceil$, for $\ell \mod 2e_{1+(f-1 \bmod \lfloor \frac{k}{r-1} \rfloor)} \in \{e_{1+(f-1 \bmod \lfloor \frac{k}{r-1} \rfloor)}, \ldots, 2e_{1+(f-1 \bmod \lfloor \frac{k}{r-1} \rfloor)} - 1\}$.
4: **end if**
5: **return**
6: **if** $f \in \{(r-2)\lfloor \frac{k}{r-1} \rfloor + 1, \ldots, k\}$ **then**
7: Repair the bit $a_{\ell,f}$ using the first parity, for $\ell \mod 2e_{f-(r-2)\lfloor \frac{k}{r-1} \rfloor} \in \{0, 1, \ldots, e_{f-(r-2)\lfloor \frac{k}{r-1} \rfloor - 1}\}$. Otherwise, repair the bit $a_{\ell,f}$ using the encoding column r, for $\ell \mod 2e_{f-(r-2)\lfloor \frac{k}{r-1} \rfloor} \in \{e_{f-(r-2)\lfloor \frac{k}{r-1} \rfloor}, \ldots, 2e_{f-(r-2)\lfloor \frac{k}{r-1} \rfloor} - 1\}$.
8: **end if**
9: **return**

The repair procedure is shown in Algorithm 1. In the algorithm, we consider two cases of column f with $1 \leqslant f \leqslant (r-2)\lfloor \frac{k}{r-1} \rfloor$ and $(r-2)\lfloor \frac{k}{r-1} \rfloor + 1 \leqslant f \leqslant k$.

Consider the case of $1 \leqslant f \leqslant (r-2)\lfloor \frac{k}{r-1} \rfloor$. It is evident that $1 \leqslant 1 + (f-1 \bmod \lfloor \frac{k}{r-1} \rfloor) \leqslant \lfloor \frac{k}{r-1} \rfloor$. For a fixed f, there exists an i, $i \in \{0, 1, \ldots, r-3\}$, such that $1 + i\lfloor \frac{k}{r-1} \rfloor \leqslant f \leqslant (i+1)\lfloor \frac{k}{r-1} \rfloor$. We thus have $\lceil \frac{f}{\lfloor \frac{k}{r-1} \rfloor} \rceil = 1 + i$. In steps 2 and 3, column f is repaired using the first encoding column and encoding column $i+2$. Thus, we divide the first $(r-2)\lfloor \frac{k}{r-1} \rfloor$ information columns into $r-2$ parts, each of which has $\lfloor \frac{k}{r-1} \rfloor$ columns. If a column belongs to the $i+1$-th part, we repair the failure column using the first encoding column and encoding column $i+2$, where $i = 0, 1, \ldots, r-3$.

Furthermore, we consider the repair bandwidth of column f. When $i = 0$, we have $1 \leqslant f \leqslant \lfloor \frac{k}{r-1} \rfloor$ and column f is repaired using the first two encoding columns. In other words, the bits $a_{\ell,f}$ are recovered using (6), when $\ell = 0, 1, \ldots, (p-1)\tau - 1$ and $\ell \bmod 2e_f$ in (8). According to Theorem 1, the repair bandwidth of column f is (5), which is asymptotically weak-optimal when k is sufficiently large.

When $i \in \{1, 2, \ldots, r-3\}$, we have $i\lfloor \frac{k}{r-1} \rfloor + 1 \leqslant f \leqslant (i+1)\lfloor \frac{k}{r-1} \rfloor$, $1 + (f-1 \bmod \lfloor \frac{k}{r-1} \rfloor) = f - i\lfloor \frac{k}{r-1} \rfloor$, and $1 + \lceil \frac{f}{\lfloor \frac{k}{r-1} \rfloor} \rceil = i + 2$. In steps 2 and 3, column f is repaired using the first encoding column and encoding column $i+2$. Recall that the transpose of encoding column $i+2$ is the right cyclic shifting of the transpose of encoding column 2 by $i\lfloor \frac{k}{r-1} \rfloor$ positions, i.e.,

$$\boldsymbol{e}\left(i\left\lfloor \frac{k}{r-1} \right\rfloor\right)^{\mathrm{T}} = \left[\underbrace{x^{e_{k-i\lfloor \frac{k}{r-1} \rfloor+1}} \cdots x^{e_k}}_{i\lfloor \frac{k}{r-1} \rfloor}\ x^{e_1}\ x^{e_2}\ \cdots x^{e_{k-i\lfloor \frac{k}{r-1} \rfloor}}\right]^{\mathrm{T}}.$$

From Theorem 1, the repair bandwidth of column $i\lfloor \frac{k}{r-1} \rfloor + f'$ is

$$k(p-1)\tau/2 + (p-1)\tau/2 + \frac{(p-1)\tau}{2e_{f'}} \cdot \left(\sum_{i=1}^{f'-1} e_i\right), \tag{13}$$

where $f' = 1, 2, \ldots, \lfloor \frac{k}{r-1} \rfloor$. Therefore, the repair bandwidth of column f is (5), which is asymptotically weak-optimal when k is sufficiently large.

Now, we consider the case of $(r-2)\lfloor \frac{k}{r-1} \rfloor + 1 \leqslant f \leqslant k$. Note that $1 \leqslant f - (r-2)\lfloor \frac{k}{r-1} \rfloor \leqslant k - (r-2)\lfloor \frac{k}{r-1} \rfloor$. In steps 6 and 7, column f is repaired using the first encoding column and encoding column r. The transpose of encoding column r is the right cyclic shifting of the transpose of encoding column 2 by $(r-2)\lfloor \frac{k}{r-1} \rfloor$ positions, i.e.,

$$\boldsymbol{e}\left((r-2)\left\lfloor \frac{k}{r-1} \right\rfloor\right)^{\mathrm{T}} = \left[\underbrace{x^{e_{k-(r-2)\lfloor \frac{k}{r-1} \rfloor+1}} \cdots x^{e_k}}_{(r-2)\lfloor \frac{k}{r-1} \rfloor}\ x^{e_1}\ x^{e_2}\ \cdots x^{e_{k-(r-2)\lfloor \frac{k}{r-1} \rfloor}}\right].$$

From Theorem 1, the repair bandwidth of column $(r-2)\lfloor \frac{k}{r-1} \rfloor + f'$ is (13), where $f' = 1, 2, \ldots, \lfloor \frac{k}{r-1} \rfloor + t$. We thus obtain the repair bandwidth of column f as (5), which is asymptotically weak-optimal when k is sufficiently large. Therefore, the repair bandwidth of k information column is asymptotically weak-optimal according to Algorithm 1 when k is sufficiently large. This completes the proof.

Considering the condition given in Theorem 2, the minimum positive value of e_i is 2^{i-1} for $i = 1, 2, \ldots, k-1$ and the minimum positive value of e_k is zero. Because τ should be a multiple of $2e_i$ for $i = 1, 2, \ldots, \lfloor \frac{k}{r-1} \rfloor + t$, the minimum value of τ is $2^{\lfloor \frac{k}{r-1} \rfloor + t}$, which is strictly less than $2^{\frac{k}{r-1}+r-1}$. The code in [18] can be viewed as a special case of the proposed codes with $e_i = 2^{i-1}$ for $i = 1, 2, \ldots, k-1$, $e_k = 0$, $\tau = 2^{k-2}$, and $r = 3$.

2.3 Example

Consider the example of $k = 3$, $p = 3$, and $r = 3$. Let $e_1 = 1, e_2 = 2, e_3 = 4$, and $\tau = 4$. The encoding matrix in this case is

$$\begin{bmatrix} 1 & x & x^4 \\ 1 & x^2 & x \\ 1 & x^4 & x^2 \end{bmatrix}.$$

Table 1 An example of the code with $k = 3, r = 3, p = 3$, and $\tau = 4$, where $\boldsymbol{a}_{8,j} = a_{0,j} + a_{4,j}$, $\boldsymbol{a}_{9,j} = a_{1,j} + a_{5,j}$, $\boldsymbol{a}_{10,j} = a_{2,j} + a_{6,j}$, and $\boldsymbol{a}_{11,j} = a_{3,j} + a_{7,j}$ for $j = 1, 2, \ldots, 6$

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4} = a_{0,1} + a_{0,2} + a_{0,3}$	$a_{0,5} = \boldsymbol{a}_{11,1} + \boldsymbol{a}_{10,2} + \boldsymbol{a}_{8,3}$	$a_{0,6} = \boldsymbol{a}_{8,1} + \boldsymbol{a}_{11,2} + \boldsymbol{a}_{10,3}$
$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4} = a_{1,1} + a_{1,2} + a_{1,3}$	$a_{1,5} = a_{0,1} + \boldsymbol{a}_{11,2} + \boldsymbol{a}_{9,3}$	$a_{1,6} = \boldsymbol{a}_{9,1} + a_{0,2} + \boldsymbol{a}_{11,3}$
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4} = a_{2,1} + a_{2,2} + a_{2,3}$	$a_{2,5} = a_{1,1} + a_{0,2} + \boldsymbol{a}_{10,3}$	$a_{2,6} = \boldsymbol{a}_{10,1} + a_{1,2} + a_{0,3}$
$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4} = a_{3,1} + a_{3,2} + a_{3,3}$	$a_{3,5} = a_{2,1} + a_{1,2} + \boldsymbol{a}_{11,3}$	$a_{3,6} = \boldsymbol{a}_{11,1} + a_{2,2} + a_{1,3}$
$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4} = a_{4,1} + a_{4,2} + a_{4,3}$	$a_{4,5} = a_{3,1} + a_{2,2} + a_{0,3}$	$a_{4,6} = a_{0,1} + a_{3,2} + a_{2,3}$
$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4} = a_{5,1} + a_{5,2} + a_{5,3}$	$a_{5,5} = a_{4,1} + a_{3,2} + a_{1,3}$	$a_{5,6} = a_{1,1} + a_{4,2} + a_{3,3}$
$a_{6,1}$	$a_{6,2}$	$a_{6,3}$	$a_{6,4} = a_{6,1} + a_{6,2} + a_{6,3}$	$a_{6,5} = a_{5,1} + a_{4,2} + a_{2,3}$	$a_{6,6} = a_{2,1} + a_{5,2} + a_{4,3}$
$a_{7,1}$	$a_{7,2}$	$a_{7,3}$	$a_{7,4} = a_{7,1} + a_{7,2} + a_{7,3}$	$a_{7,5} = a_{6,1} + a_{5,2} + a_{3,3}$	$a_{7,6} = a_{3,1} + a_{6,2} + a_{5,3}$

The 24 information bits are represented by $a_{0,i}, a_{1,i}, \ldots, a_{7,i}$ for $i = 1, 2, 3$. The example is illustrated in Table 1, wherein the bits in bold are extra bits.

Suppose that the first information column fails, i.e., $f = 1$. Recall that $k = 3, p = 3, r = 3$, and $\tau = 4$ and that $e_1 = 1, e_2 = 2$, and $e_3 = 4$. We have $e_{1+(f-1 \bmod \lfloor \frac{k}{r-1} \rfloor)} = e_1 = 1$ and $1 + \lceil \frac{f}{\lfloor \frac{k}{r-1} \rfloor} \rceil = 2$. Based on steps 2–4 in Algorithm 1, we can repair the bits $a_{\ell,1}$ using the first encoding column for $\ell \equiv 0 \mod 2$ and using the second encoding column for $\ell \equiv 1 \mod 2$, where $0 \leqslant \ell \leqslant 7$. More specifically, the bits $a_{0,1}, a_{2,1}, a_{4,1}$, and $a_{6,1}$ are rebuilt using

$$\begin{aligned}
a_{0,1} &= a_{0,2} + a_{0,3} + (a_{0,4} = a_{0,1} + a_{0,2} + a_{0,3}),\\
a_{2,1} &= a_{2,2} + a_{2,3} + (a_{2,4} = a_{2,1} + a_{2,2} + a_{2,3}),\\
a_{4,1} &= a_{4,2} + a_{4,3} + (a_{4,4} = a_{4,1} + a_{4,2} + a_{4,3}),\\
a_{6,1} &= a_{6,2} + a_{6,3} + (a_{6,4} = a_{6,1} + a_{6,2} + a_{6,3}),
\end{aligned}$$

and the bits $a_{1,1}, a_{3,1}, a_{5,1}, a_{7,1}$ are rebuilt using

$$\begin{aligned}
a_{1,1} &= a_{0,2} + \boldsymbol{a}_{10,3} + (a_{2,5} = a_{1,1} + a_{0,2} + \boldsymbol{a}_{10,3}),\\
a_{3,1} &= a_{2,2} + a_{0,3} + (a_{4,5} = a_{3,1} + a_{2,2} + a_{0,3}),\\
a_{5,1} &= a_{4,2} + a_{2,3} + (a_{6,5} = a_{5,1} + a_{4,2} + a_{2,3}),\\
a_{7,1} &= \boldsymbol{a}_{10,2} + \boldsymbol{a}_{8,3} + a_{2,2} + a_{0,3} + (a_{0,5} + a_{4,5}) = \mathrm{a}_{11,1} + a_{3,1}.
\end{aligned}$$

Therefore, we can repair $a_{0,1}, a_{2,1}, a_{4,1}$, and $a_{6,1}$ by downloading 12 bits $a_{0,j}, a_{2,j}, a_{4,j}$, and $a_{6,j}$ for $j = 2, 3, 4$. Because we can compute $\boldsymbol{a}_{10,2}$ by $a_{2,2} + a_{6,2}$, $\boldsymbol{a}_{10,3}$ by $a_{2,3} + a_{6,3}$, and $\boldsymbol{a}_{8,3}$ by $a_{0,3} + a_{4,3}$, we require the bits

$$a_{0,2}, a_{2,2}, a_{4,2}, a_{6,2}, a_{0,3}, a_{2,3}, a_{4,3}, a_{6,3}, a_{0,5}, a_{2,5}, a_{4,5}, a_{6,5},$$

to repair $a_{1,1}, a_{3,1}, a_{5,1}$, and $a_{7,1}$. Recall that the bits $a_{0,j}, a_{2,j}, a_{4,j}$, and $a_{6,j}$ for $j = 2, 3, 4$ have already been downloaded to repair $a_{0,1}, a_{2,1}, a_{4,1}$, and $a_{6,1}$. Therefore, it is sufficient to download $a_{0,5}, a_{2,5}, a_{4,5}$, and $a_{6,5}$ to repair $a_{1,1}, a_{3,1}, a_{5,1}$, and $a_{7,1}$. Sixteen bits are downloaded from four columns to repair the bits of the first information column.

For the code given in Table 1, we can verify that the information bits stored in the second and third information columns can be rebuilt by accessing 16 and 18 bits, respectively.

3 The MDS property

When we state that an $L \times (k + 2)$ array code is MDS, it indicates that any k columns are sufficient to recover all the information bits. In this section, we present the MDS property condition.

From the encoding procedure, we have $a_\ell(x) \in C_{p\tau}$ for all $\ell = 1, 2, \ldots, k$ according to (2) and Lemma 1. Because $a_\ell(x) \in C_{p\tau}$ and $x^i a_\ell(x) \in C_{p\tau}$ for integer i and $\ell = 1, 2, \ldots, k$, we determine that the r coded

polynomials are in $C_{p\tau}$. Thus, we are effectively considering the ring $C_{p\tau}$. Because the ring $C_{p\tau}$ is isomorphic to $\mathbb{F}_2[x]/M_p^\tau(x)$ [10, Lemma 3], where

$$M_p^\tau(x) = x^{(p-1)\tau} + x^{(p-2)\tau} + \cdots + x^\tau + 1,$$

it is sufficient to show that the determinant of any square sub-matrix of the encoding matrix $\boldsymbol{P}_{k\times r}$ in (12) is invertible over $\mathbb{F}_2[x]/M_p^\tau(x)$.

Theorem 3. Let $M_p^\tau(x)$ be factorized as a product of powers of irreducible polynomials over $\mathbb{F}_2$:

$$M_p^\tau(x) = (f_1(x))^{\ell_1} \cdot (f_2(x))^{\ell_2} \cdots (f_t(x))^{\ell_t}, \tag{14}$$

where $\deg(f_1(x)) \leqslant \deg(f_2(x)) \leqslant \cdots \leqslant \deg(f_t(x))$. If $e_i \neq e_j$ for $1 \leqslant i \neq j \leqslant k$ satisfies the condition in Theorem 1 and $\deg(f_1(x))$ is larger than $(r-1)\max_{\ell\in\{1,2,\ldots,k\}}(e_\ell)$, then the proposed code with the encoding matrix given in (12) is an MDS code for $k \geqslant r$.

Proof. From Theorem 6 in [10], the ring $\mathbb{F}_2[x]/M_p^\tau(x)$ is isomorphic to the direct sum of t rings $\mathbb{F}_2[x]/(f_1(x))^{\ell_1}$, $\mathbb{F}_2[x]/(f_2(x))^{\ell_2}, \ldots, \mathbb{F}_2[x]/(f_t(x))^{\ell_t}$. It is sufficient to show that the determinants of all sub-matrices are invertible in $\mathbb{F}_2[x]/(f_i(x))$ for $i = 1, 2, \ldots, t$. Note that we can view $e_1, e_2, \ldots, e_k$ as k distinct variables that satisfy the condition in Theorem 1. It can be observed that the determinant of any $\ell \times \ell$ sub-matrix is non-zero, where $1 \leqslant \ell \leqslant r$. If the maximum degree of the non-zero determinant is less than $\deg(f_1(x))$, then the determinant is invertible in $\mathbb{F}_2[x]/(f_i(x))$.

As any $\ell \times \ell$, $1 \leqslant \ell \leqslant r$, the sub-matrix of (12) is contained in an $r \times r$ sub-matrix, and the maximum degree among all the determinants of the $\ell \times \ell$ sub-matrices is no larger than that among all the determinants of the $r \times r$ sub-matrices. It is sufficient to calculate the maximum degree among the determinants of all the $r \times r$ sub-matrices of (12).

Note that the size of the matrix in (12) is $k \times r$, where $k \geqslant r$. We must first choose r rows from the k rows to form an $r \times r$ sub-matrix and thereafter calculate the maximum exponent of the determinant of the $r \times r$ sub-matrix. The determinant is computed as the summation (with plus or minus signs) of all possible multiplications of r entries present in different rows and different columns. Because the first column of (12) is an all-one vector, we can choose any one entry in the first column that does not affect the determinant. The second column of (12) is $\boldsymbol{e}^{\mathrm{T}}$ and the maximum exponent in all the entries of the second column is $\max_{\ell\in\{1,2,\ldots,k\}}(e_\ell)$. We denote the row whose entry has the maximum exponent as $\ell_{\max}$, i.e., $e_{\ell_{\max}} = \max_{\ell\in\{1,2,\ldots,k\}}(e_\ell)$, where $1 \leqslant \ell_{\max} \leqslant k$. The ith column of (12) is the cyclic shift of the second column of (12) by $(i-2)\lfloor\frac{k}{r-1}\rfloor$ positions, where $i = 3, 4, \ldots, r$, and therefore the entry with the maximum exponent in the i-th column is in row $((i-2)\lfloor\frac{k}{r-1}\rfloor + \ell_{\max}) \bmod k$. For $2 \leqslant i \neq j \leqslant k$, we have

$$\left((i-2)\left\lfloor\frac{k}{r-1}\right\rfloor + \ell_{\max}\right) \bmod k \neq \left((j-2)\left\lfloor\frac{k}{r-1}\right\rfloor + \ell_{\max}\right) \bmod k.$$

Otherwise, we can obtain the contradiction that $i = j$. Therefore, the entries with the maximum exponent in columns $2, 3, \ldots, r$ are in different rows and the maximum degree of the determinant can be computed by summing all the maximum degrees of the entries in r columns. The maximum degree is $(r-1)\max_{\ell\in\{1,2,\ldots,k\}}(e_\ell)$. This completes the proof.

From Theorem 2, the minimum value of τ is $2^{\lfloor\frac{k}{r-1}\rfloor+t}$. If τ is a power of 2, then we have

$$M_p^\tau(x) = 1 + x^\tau + x^{2\tau} + \cdots + x^{(p-1)\tau} = (1 + x + \cdots + x^{p-1})^\tau.$$

According to Theorem 3, the sufficient condition is that p is larger than $(r-1)\max_{\ell\in\{1,2,\ldots,k\}}(e_\ell)$ when τ is a power of 2.

4 Comparison and conclusion

In Table 2, we summarize the comparison of binary MDS array codes in terms of repair bandwidth, the number of helper columns, and the number of parity columns. We can observe from Table 2 that

Table 2 Comparison of binary MDS array codes

Code	Repair bandwidth	Number of helper columns d	Number of parity columns r
ButterFly code [16, 17]	Optimal	$k+1$	2
MDR code [19, 20]	Optimal	$k+1$	2
Code in [18]	Asymptotically weak-optimal	$k+1$	3
Code-I in [10]	Asymptotically weak-optimal	$k+\frac{r-1}{2}$	$r \geqslant 3$ is odd
Code-II in [10]	Asymptotically weak-optimal	$k+\frac{r}{2}$	$r \geqslant 4$ is even
Proposed code	Asymptotically weak-optimal	$k+1$	$r \geqslant 2$

our constructed code is the first binary MDS array code with any positive $r \geqslant 2$ and an asymptotically weak-optimal repair bandwidth.

When $r = 3$, the encoding complexity of the proposed code is the same as that of the code in [18] and is comparable to that of the existing binary MDS array codes such as STAR codes [3]. Please refer to [18] for a detailed discussion about the encoding complexity of the proposed code with $r = 3$.

Herein, we present new binary MDS array codes over a specific binary quotient ring with more than two parity columns. The property of the encoding matrix that can asymptotically achieve weak-optimal repair of an information column with $d = k + 1$ helper nodes is exploited. We have shown that many encoding matrices satisfy the property.

We summarize possible future work as follows. First, in the present paper we focused only on the property of the encoding matrix for $d = k + 1$ helper nodes. It would be interesting to investigate the property of the encoding matrix that can asymptotically achieve weak-optimal repair of an information column with $n - 1 \geqslant d \geqslant k + 1$ helper nodes. Second, the proposed codes achieve asymptotically weak-optimal repair only for any information column. Modifying the construction to achieve asymptotically weak-optimal repair for any parity column would be interesting future work. A generic transformation proposed in [12] can achieve weak-optimal repair bandwidths for non-binary MDS codes. Future work will be to apply that generic transformation to achieve weak-optimal repair bandwidths for binary MDS array codes.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61701115, 61671007).

References

1 Blaum M, Brady J, Bruck J, et al. EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures. IEEE Trans Comput, 1995, 44: 192–202
2 Corbett P, English B, Goel A, et al. Row-diagonal parity for double disk failure correction. In: Proceedings of the 3rd USENIX Conference on File and Storage Technologies, San Jose, 2004. 1–14
3 Huang C, Xu L. STAR: an efficient coding scheme for correcting triple storage node failures. IEEE Trans Comput, 2008, 57: 889–901
4 Blaum M. A family of MDS array codes with minimal number of encoding operations. In: Proceedings of IEEE International Symposium on Information Theory, Seattle, 2006. 2784–2788
5 Blaum M, Brady J, Bruck J, et al. The EVENODD code and its generalization. In: Proceedings of High Performance Mass Storage and Parallel I/O. Hoboken: John Wiley & Sons, Inc., 2001. 187–208
6 Blaum M, Bruck J, Vardy A. MDS array codes with independent parity symbols. IEEE Trans Inform Theor, 1996, 42: 529–542
7 Feng G-L, Deng R-H, Bao F, et al. New efficient MDS array codes for RAID. Part II. Rabin-like codes for tolerating multiple (= 4) disk failures. IEEE Trans Comput, 2005, 54: 1473–1483
8 Hou H, Han Y S. A new construction and an efficient decoding method for rabin-like codes. IEEE Trans Commun, 2018, 66: 521–533
9 Dimakis A, Godfrey P, Wu Y, et al. Network coding for distributed storage systems. IEEE Trans Inf Theory, 2010, 56: 4539–4551
10 Hou H X, Han Y-S, Lee P-P-C, et al. A new design of binary MDS array codes with asymptotically weak-optimal repair. 2018. ArXiv:1802.07891
11 Hou H X, Shum K W, Chen M, et al. BASIC codes: low-complexity regenerating codes for distributed storage systems. IEEE Trans Inform Theor, 2016, 62: 3053–3069
12 Li J, Tang X H, Tian C. A generic transformation for optimal repair bandwidth and rebuilding access in MDS codes. In: Proceedings of IEEE International Symposium on Information Theory (ISIT), Aachen, 2017. 1623–1627

13 Rashmi K V, Shah N B, Kumar P V. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. IEEE Trans Inform Theor, 2011, 57: 5227–5239
14 Tamo I, Wang Z, Bruck J. Zigzag codes: MDS array codes with optimal rebuilding. IEEE Trans Inform Theor, 2013, 59: 1597–1616
15 Ye M, Barg A. Explicit constructions of high-rate MDS array codes with optimal repair bandwidth. IEEE Trans Inform Theor, 2017, 63: 2001–2014
16 Gad E-E, Mateescu R, Blagojevic F, et al. Repair-optimal MDS array codes over GF(2). In: Proceedings of IEEE International Symposium Information Theory, Istanbul, 2013. 887–891
17 Pamies J-L, Blagojevic F, Mateescu R, et al. Opening the chrysalis: on the real repair performance of MSR codes. In: Proceedings of 14th USENIX Conference on File and Storage Technologies, Santa Clara, 2016. 81–94
18 Hou H, Lee P-P-C, Han Y-S, et al. Triple-fault-tolerant binary MDS array codes with asymptotically optimal repair. In: Proceedings of IEEE International Symposium Information Theory, Aachen, 2017. 839–843
19 Wang Y, Yin X, Wang X. MDR codes: a new class of RAID-6 codes with optimal rebuilding and encoding. IEEE J Sele Areas Commun, 2014, 32: 1008–1018
20 Wang Y, Yin X, Wang X. Two new classes of two-parity MDS array codes with optimal repair. IEEE Commun Lett, 2016, 20: 1293–1296
21 Xiang L, Xu Y, Lui J, et al. Optimal recovery of single disk failure in RDP code storage systems. In: Proceedings of ACM SIGMETRICS Performance Evaluation Rev, New York, 2010. 119–130
22 Xu S, Li R, Lee P P C, et al. Single disk failure recovery for X-code-based parallel storage systems. IEEE Trans Comput, 2014, 63: 995–1007
23 Wang Z Y, Dimakis A-G, Bruck J. Rebuilding for array codes in distributed storage systems. In: Proceedings of GLOBECOM Workshops, Miami, 2010. 1905–1909