Constructing (h, d) cooperative MSR codes with sub-packetization $(d-k+h)(d-k+1)^{\lceil n/2 \rceil}$

Zihao Zhang, Guodong Li, and Sihuang Hu

Abstract

We address the multi-node failure repair challenges for MDS array codes. Presently, two primary models are employed for multi-node repairs: the centralized model where all failed nodes are restored in a singular data center, and the cooperative model where failed nodes acquire data from auxiliary nodes and collaborate amongst themselves for the repair process. This paper focuses on the cooperative model, and we provide explicit constructions of optimal MDS array codes with d helper nodes under this model. The sub-packetization level of our new codes is $(d-k+h)(d-k+1)^{\lceil n/2 \rceil}$ where h is the number of failed nodes, k the number of information nodes and n the code length. This improves upon recent constructions given by Liu *et al.* (IEEE Transactions on Information Theory, Vol. 69, 2023).

I. Introduction

Erasure codes are widely used in current distributed storage systems, where they enhance data robustness by adding redundancy to tolerate data node failures. Common erasure codes include maximum distance separable (MDS) codes and locally repairable codes (LRC). Particularly, MDS codes have garnered significant attention because they provide the maximum failure tolerance for a given amount of storage overhead.

An (n, k, ℓ) *array code* has k information coordinates and $r = n - k$ parity check coordinates, where each coordinate is a vector in $\mathbb{F}_q^\ell$ for some finite field $\mathbb{F}_q$. Formally, a (linear) (n, k, ℓ) array code $\mathcal{C}$ can be defined by its parity check equations, i.e.,

$$\mathcal{C} = \{(C_0, \ldots, C_{n-1}) : H_0 C_0 + \cdots + H_{n-1} C_{n-1} = \mathbf{0}\},$$

where each C_i is a column vector of length ℓ over $\mathbb{F}_q$, and each H_i is a $r\ell \times \ell$ matrix over $\mathbb{F}_q$. We call $\mathcal{C}$ an *MDS array code* if any r out of its n coordinates can be recovered from the other k coordinates. To be specific, let $\mathcal{F} = \{i_1, i_2, \ldots, i_r\} \subset [n]$ be the collection of indices of r failed nodes, we have

$$\sum_{i \in \mathcal{F}} H_i C_i = - \sum_{i \in [n] \setminus \mathcal{F}} H_i C_i,$$

where we use $[n]$ to denote the set $\{0, 1, \ldots, n-1\}$. Then we know that the r coordinates C_i $(i \in \mathcal{F})$ can be recovered from the other k coordinates C_i $(i \in [n] \setminus \mathcal{F})$ if and only if the square matrix $[H_{i_1} \quad H_{i_2} \quad \ldots \quad H_{i_r}]$ is invertible. Equivalently, we say a set of n matrices $H_0, H_1, \ldots, H_{n-1}$ in $\mathbb{F}_q^{r\ell \times \ell}$ defines an (n, k, ℓ) MDS array code if

$$[H_{i_1} \quad H_{i_2} \quad \cdots \quad H_{i_r}] \text{ is invertible } \forall \, \{i_1, i_2, \ldots, i_r\} \subset [n].$$

With the emergence of large-scale distributed storage systems, the notion of *repair bandwidth* was introduced to measure the efficiency of recovering the erasure of a single codeword coordinate. The seminal work by Dimakis *et. al.* [1] pointed out that we can repair a single failed node by smaller repair bandwidths than the trivial MDS repair scheme. More precisely, for an (n, k, ℓ) MDS array code, the optimal repair bandwidth for a single node failure by downloading data from $d \geq k$ helper nodes is

$$\frac{d\ell}{d-k+1}. \tag{1}$$

We call an (n, k, ℓ) MDS array code *minimum storage regenerating (MSR) code with repair degree* d if it achieves the lower bound (1) for the repair of any single erased coordinate from any d out of $n-1$ remaining coordinates. Please see [2]–[12] and references therein for the constructions and studies of MSR codes.

MSR codes can efficiently recover a single failed node using the smallest possible bandwidth. Naturally, new variants of MSR codes are adopted to handle the case when $h > 1$ nodes fail simultaneously. Under the centralized repair, a single repair center downloads helper data from d helper nodes and uses this data to produce h replacement nodes (please see [13]–[24] and references therein). Another scheme of repairing multiple failed nodes simultaneously is cooperative repair, where failed nodes

Research partially funded by National Key R&D Program of China under Grant No. 2021YFA1001000, National Natural Science Foundation of China under Grant No. 12001322 and 12231014, a Taishan scholar program of Shandong Province, and CCF-Huawei Populus Grove Fund.

Zihao Zhang, Guodong Li and Sihuang Hu are with Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Qingdao, Shandong, 266237, China and School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, 266237, China. S. Hu is also with Quan Cheng Laboratory, Jinan 250103, China. Email: {zihaozhang, guodongli}@mail.sdu.edu.cn, husihuang@sdu.edu.cn

acquire data from auxiliary nodes and collaborate amongst themselves for the repair process. Notably, the cooperative model has demonstrated greater robustness compared to its centralized counterpart, being able to deduce a corresponding centralized model under equivalent parameters. Please refer to [25]–[32] and references therein for the results on cooperative MSR codes.

This paper primarily focuses on the cooperative model, and all subsequent references to repair bandwidth and cut-set bounds are made within this context.

Theorem 1. *(Cut-set bound [25]) For an (n, k, ℓ) MDS array code, the optimal repair bandwidth for h failed nodes by downloading information from d helper nodes under the cooperative repair scheme is*

$$\frac{h(d+h-1)\ell}{d-k+h}. \tag{2}$$

We say that an (n, k, ℓ) MDS array code $\mathcal{C}$ is an (h, d)-*MSR code under the cooperative model* if any h failed nodes can be recovered from any other d helper nodes with total bandwidth achieving the lower bound (2). Note that a $(1, d)$-MSR code is just an MSR code with repair degree d.

A. Previous works on cooperative MSR codes

In [29], Ye and Barg provided an explicit construction for cooperative MSR codes with all admissible parameters. The sub-packetization level of the construction in [29] is given by $((d-k)^{h-1}(d-k+h))^{\binom{n}{h}}$. Subsequent work has been focused on reducing the sub-packetization of cooperative MSR codes. In [30], Zhang *et al.* introduced a construction with optimal access property, where $\ell = (d-k+h)^{\binom{n}{h}}$. Subsequently, in the work of Ye [31], the sub-packetization was further reduced to $(d-k+h)(d-k+1)^n$. More recently, Liu's work [32] achieved even lower sub-packetization for the case $d = k+1$: the sub-packetization of the new construction is $o \cdot 2^n$ where o is the largest odd number such that $o \mid (h+1)$.

Codes	Sub-packetization ℓ	Restrictions
Ye and Barg 2019 [29]	$((d-k)^{h-1}m)^{\binom{n}{h}}$	
Zhang *et al.* 2020 [30]	$m^{\binom{n}{h}}$	
Ye 2020 [31]	ms^n	
Liu *et al.* 2023 [32]	os^n	$d = k+1$
This paper	$ms^{\lceil n/2 \rceil}$	

TABLE I

SUB-PACKETIZATIONS AND RESTRICTIONS OF DIFFERENT CONSTRUCTIONS OF (h, d)-COOPERATIVE MSR CODES, WHERE $s = d-k+1$, $m = d-k+h$ AND o IS THE LARGEST ODD NUMBER SATISFYING $o \mid m$.

B. Our contributions

In this paper, we present a construction of cooperative MSR codes with all admissible parameters (h, d) and $\ell = (d-k+h)(d-k+1)^{\lceil n/2 \rceil}$. Our approach is inspired by the construction of MSR codes in [12], which introduced a method to design parity check sub-matrices using the so-called kernel matrices and blow-up map. In this work, we first introduce new kernel matrices and then blow up them to construct new $(1, d)$-MSR codes. Then, similarly as [31], we replicate the $(1, d)$-MSR code $d-k+h$ times obtaining an (h, d)-MSR code. The optimal repair scheme is guaranteed by the suitably chosen cooperative pairing matrices.

The rest of this paper is organized as follows: In Section II, we provide necessary definitions and notations for our construction. In Section III, we present our new construction and prove its MDS property. In Section IV, we describe the repair scheme of our new nodes, which achieves the optimal repair bandwidth.

II. PRELIMINARIES

Let $\mathbb{F}_q$ be a finite field with order q. For a positive integer m, we define $[m] := (0, 1, \cdots, m-1)$. For an integer a, we define

$$a + [m] := (a + x : x \in [m]),$$

and we denote the vector $x_{[m]}$ over $\mathbb{F}_q$ as $(x_j : j \in [m])$. Let $\mathbf{I}_m$ be the $m \times m$ identity matrix over $\mathbb{F}_q$. For an element $x \in \mathbb{F}_q$, and a positive integer t, we define a column vector of length t as

$$L^{(t)}(x) := \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{t-1} \end{bmatrix}.$$

Assume that s, t are two positive integers. For each $i \in [s^t]$, we write

$$i = \sum_{z \in [t]} i_z s^z, \ i_z \in [s].$$

Here we use i_z to denote the z-th digit in the t digits base-s expansion of i. To simplify notations, we need the following matrix operator $\boxtimes$ and blow-up map introduced in [12].

Definition 1. *For a matrix A and an $m \times n$ block matrix B written as*

$$B = \begin{bmatrix} B_{0,0} & \cdots & B_{0,n-1} \\ \vdots & \ddots & \vdots \\ B_{m-1,0} & \cdots & B_{m-1,n-1} \end{bmatrix},$$

we define

$$A \boxtimes B := \begin{bmatrix} A \otimes B_{0,0} & \cdots & A \otimes B_{0,n-1} \\ \vdots & \ddots & \vdots \\ A \otimes B_{m-1,0} & \cdots & A \otimes B_{m-1,n-1} \end{bmatrix},$$

where $\otimes$ is the Kronecker product. Note that the result $A \boxtimes B$ is dependent on how the rows and columns of B are partitioned, and we will specify the partition every time we use this notation. If every block entry $B_{i,j}$ is a scalar over $\mathbb{F}_q$, we have $A \boxtimes B = B \otimes A$.

Throughout this paper, when we say that B is an $m \times n$ block matrix, we always assume that B is uniformly partitioned, i.e. each block entry of B is of the same size.

Definition 2 (Blow-up)**.** *Let t be a positive integer. For any $a \in [t]$, we* blow up *an $s \times s$ block matrix*

$$K = \begin{bmatrix} K_{0,0} & \cdots & K_{0,s-1} \\ \vdots & \ddots & \vdots \\ K_{s-1,0} & \cdots & K_{s-1,s-1} \end{bmatrix}$$

to get an $s^t \times s^t$ block matrix via

$$\begin{aligned} \Phi_{t,a}(K) &= \mathbf{I}_{s^{t-a-1}} \otimes (\mathbf{I}_{s^a} \boxtimes K) \\ &= \mathbf{I}_{s^{t-a-1}} \otimes \begin{bmatrix} \mathbf{I}_{s^a} \otimes K_{0,0} & \cdots & \mathbf{I}_{s^a} \otimes K_{0,s-1} \\ \vdots & \ddots & \vdots \\ \mathbf{I}_{s^a} \otimes K_{s-1,0} & \cdots & \mathbf{I}_{s^a} \otimes K_{s-1,s-1} \end{bmatrix}. \end{aligned}$$

The following lemma shows the relationship between an $s \times s$ block matrix K and its blown-up $s^t \times s^t$ block matrix $\Phi_{t,a}(K)$.

Lemma 2. *For $i, j \in [s^t]$, the block entry of $\Phi_{t,a}(K)$ at the ith block row and jth block column*

$$\Phi_{t,a}(K)(i,j) = \begin{cases} K(i_a, j_a) & \text{if } i_z = j_z \ \forall z \in [t] \setminus \{a\} \\ \mathbf{O} & \text{otherwise}, \end{cases}$$

where $K(i_a, j_a)$ is the block entry of K at the i_ath block row and j_ath block column.

Proof. We prove this lemma by induction. It is easy to see that the conclusion holds for the case $t = 1$. Now assume that the conclusion holds for some positive integer t and any $a \in [t]$, that is,

$$\Phi_{t,a}(K)(i,j) = \begin{cases} K(i_a, j_a) & \text{if } i_z = j_z \ \forall z \in [t] \setminus \{a\} \\ \mathbf{O} & \text{otherwise}, \end{cases} \tag{3}$$

where $i, j \in [s^t]$.

We proceed to prove the case $t + 1$. If $a = t$ then $\Phi_{t+1,t}(K) = \mathbf{I}_{s^t} \boxtimes K$, and we can verify that

$$\Phi_{t+1,t}(K)(i,j) = \begin{cases} K(i_t, j_t) & \text{if } i_z = j_z \ \forall z \in [t] \\ \mathbf{O} & \text{otherwise}, \end{cases}$$

where $i, j \in [s^{t+1}]$. If $0 \le a \le t-1$, then by definition $\Phi_{t+1,a}(K) = \mathbf{I}_s \otimes \Phi_{t,a}(K)$. By (3) we get

$$\Phi_{t+1,a}(K)(i,j) = \begin{cases} K(i_a, j_a) & \text{if } i_z = j_z \ \forall z \in [t+1] \setminus \{a\} \\ \mathbf{O} & \text{otherwise}, \end{cases}$$

where $i, j \in [s^{t+1}]$. This concludes the proof. $\square$

The following properties of blown-up matrices will be used for the repair scheme of our codes.

Lemma 3. *Let A, B and C be three $s \times s$ block matrices. If*

$$(\mathbf{I}_s \otimes A)(\mathbf{I}_s \boxtimes B) = (\mathbf{I}_s \boxtimes B)(\mathbf{I}_s \otimes C)$$[1]

then for any positive integer t and $a_0 \neq a_1 \in [t]$,

$$\Phi_{t,a_0}(A)\Phi_{t,a_1}(B) = \Phi_{t,a_1}(B)\Phi_{t,a_0}(C).$$

Proof. By Lemma 2, we have

$$\Phi_{t,a_0}(A)(u,v) = \begin{cases} A(u_{a_0}, v_{a_0}) & \text{if } u_i = v_i, \forall i \in [t] \backslash \{a_0\} \\ \mathbf{O} & \text{otherwise,} \end{cases}$$

$$\Phi_{t,a_1}(B)(u,v) = \begin{cases} B(u_{a_1}, v_{a_1}) & \text{if } u_i = v_i, \forall i \in [t] \backslash \{a_1\} \\ \mathbf{O} & \text{otherwise,} \end{cases}$$

and

$$\Phi_{t,a_0}(C)(u,v) = \begin{cases} C(u_{a_0}, v_{a_0}) & \text{if } u_i = v_i, \forall i \in [t] \backslash \{a_0\} \\ \mathbf{O} & \text{otherwise,} \end{cases}$$

where $u, v \in [s^t]$. We also regard $\Phi_{t,a_0}(A)\Phi_{t,a_1}(B)$ and $\Phi_{t,a_1}(B)\Phi_{t,a_0}(C)$ as $s^t \times s^t$ block matrices. Note that $a_0 \neq a_1$. Then by the above we can verify that

$$\begin{aligned} &[\Phi_{t,a_0}(A)\Phi_{t,a_1}(B)](u,v) \\ =& \sum_{w \in [s^t]} \Phi_{t,a_0}(A)(u,w)\Phi_{t,a_1}(B)(w,v) \\ =& \begin{cases} A(u_{a_0}, v_{a_0})B(u_{a_1}, v_{a_1}) & \text{if } u_i = v_i, \forall i \in [t] \backslash \{a_0, a_1\} \\ \mathbf{O} & \text{otherwise,} \end{cases} \end{aligned}$$

and

$$\begin{aligned} &[\Phi_{t,a_1}(B)\Phi_{t,a_0}(C)](u,v) \\ =& \begin{cases} B(u_{a_1}, v_{a_1})C(u_{a_0}, v_{a_0}) & \text{if } u_i = v_i, \forall i \in [t] \backslash \{a_0, a_1\} \\ \mathbf{O} & \text{otherwise.} \end{cases} \end{aligned}$$

Now we can see that

$$\Phi_{t,a_0}(A)\Phi_{t,a_1}(B) = \Phi_{t,a_1}(B)\Phi_{t,a_0}(C)$$

if and only if for any $(i_0, j_0), (i_1, j_1) \in [s]^2$,

$$A(i_0, j_0)B(i_1, j_1) = B(i_1, j_1)C(i_0, j_0).$$

The latter is equivalent to

$$(\mathbf{I}_s \otimes A)(\mathbf{I}_s \boxtimes B) = (\mathbf{I}_s \boxtimes B)(\mathbf{I}_s \otimes C).$$

This concludes our proof. $\square$

The following result can be obtained easily by the mixed-product property of the Kronecker product, therefore we omit its proof.

Lemma 4. *Let A and B be two $s \times s$ block matrices. Then for any positive integer t and $a \in [t]$, we have*

$$\Phi_{t,a}(A)\Phi_{t,a}(B) = \Phi_{t,a}(AB)$$

if AB is a valid matrix product.

[1]This condition is equivalent to $\Phi_{2,0}(A)\Phi_{2,1}(B) = \Phi_{2,1}(B)\Phi_{2,0}(C)$.

III. Code construction and MDS property

Given code length n, dimension k and repair degree d, we use $r = n - k$ to denote the redundancy of our code, and set $s = d - k + 1$. Assume that the number of failed nodes h satisfies that $k + 1 \le d \le n - h$. In this section, we construct an $(n, k, \ell = (d-k+h)s^{\lceil n/2 \rceil})$ cooperative MSR code with repair degree d for any h failed nodes. Without loss of generality, we always assume that $2|n$. Then $\ell = (d-k+h)s^{n/2}$ and we write $\tilde{\ell} = s^{n/2}$. The codeword $(C_0, C_1, \cdots, C_{n-1})$ of the (n, k, ℓ) array code is divided into $n/2$ groups of size 2. We use $a \in [n/2], b \in [2]$ to denote the group's index and the node's index within its group, respectively. In other words, the group a consists of the two nodes C_{2a} and C_{2a+1}.

To begin with, we select sn distinct elements $\lambda_{[sn]}$ from $\mathbb{F}_q$ and define the following *kernel map*

$$\mathcal{K}^{(t)} : \mathbb{F}_q^s \to \mathbb{F}_q^{st \times s},$$

which maps $x_{[s]}$ to the following $s \times s$ block matrix

$$\begin{aligned}\mathcal{K}^{(t)}(x_{[s]}) &= \mathbf{1}^{(s)} \boxtimes [L^{(t)}(x_0)\ L^{(t)}(x_1)\ \cdots\ L^{(t)}(x_{s-1})] \\ &= \begin{bmatrix} L^{(t)}(x_0) & L^{(t)}(x_1) & \cdots & L^{(t)}(x_{s-1}) \\ \vdots & \vdots & \vdots & \vdots \\ L^{(t)}(x_0) & L^{(t)}(x_1) & \cdots & L^{(t)}(x_{s-1}) \end{bmatrix}.\end{aligned}$$

where $\mathbf{1}^{(s)}$ is the all-one column vector of length s.

Definition 3. *We say a matrix is* entrywise non-zero *if it has no zero entry. Given two entrywise non-zero matrices $U, V \in \mathbb{F}_q^{s \times s}$, we call them* cooperative pairing *matrices if $UV = \mathbf{I}_s$.*

The cooperative pairing matrices will play a pivotal role in our cooperative repair scheme of Section IV. Now we provide a simple method to obtain cooperative pairing (circulant) matrices. We first need the following useful map

$$\begin{aligned}\mathbf{rot}(\cdot): \quad \mathbb{F}_q[x]/(x^s - 1) \quad &\to \quad \mathbb{F}_q^{s \times s} \\ \sum_{i=0}^{s-1} c_i x^i \quad &\mapsto \quad \begin{bmatrix} c_0 & c_1 & \cdots & c_{s-1} \\ c_{s-1} & c_0 & \cdots & c_{s-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{bmatrix},\end{aligned}$$

which maps a polynomial to a circulant matrix. Then the following lemma shows us how to find cooperative pairing (circulant) matrices.

Lemma 5. *Suppose there exists some element $\gamma \in \mathbb{F}_q$ such that*

$$g(\gamma) = \gamma(\gamma - 1)(\gamma + s - 1)(\gamma + s - 2) \neq 0.$$

Set

$$\begin{aligned} F_0 &= x^{s-1} + \cdots + x + \gamma, \\ F_1 &= \frac{x^{s-1} + \cdots + x - (\gamma + s - 2)}{-(\gamma - 1)(\gamma + s - 1)}. \end{aligned}$$

Then $F_0 F_1 = 1$ in $\mathbb{F}_q[x]/(x^s - 1)$ and $\mathbf{rot}(F_0), \mathbf{rot}(F_1)$ are cooperative pairing matrices.

Proof. By direct computations, we can easily check that $F_0 F_1 = 1$ and $\mathbf{rot}(F_0)\mathbf{rot}(F_1) = \mathbf{I}_s$. □

From now on we set

$$\begin{aligned} U_0 &= \mathbf{I}_s, & U_1 &= \mathbf{rot}(F_1), \\ V_0 &= \mathbf{rot}(F_0), & V_1 &= \mathbf{I}_s, \end{aligned}$$

where $\mathbf{rot}(F_0)$ and $\mathbf{rot}(F_1)$ are defined as in Lemma 5. We can check that [2]

$$\begin{aligned} U_b V_b &= \mathbf{rot}(F_b), \\ U_b V_{b \oplus 1} &= \mathbf{I}_s, \end{aligned}$$

for all $b \in [2]$.

Now, we are ready to define the following *kernel matrices*. For $a \in [n/2], b \in [2]$ and a positive integer t, we define

$$K_{a,b}^{(t)} = (V_b \otimes \mathbf{1}^{(t)}) \odot \mathcal{K}^{(t)}(\lambda_{s(2a+b)+[s]})$$

[2] For any integers a and b, the operation $\oplus_s$ is defined as $a \oplus_s b = (a + b) \bmod s$. And we use $\oplus$ as a shorthand for $\oplus_2$.

where $\odot$ is the Hadamard (elementwise) product of two matrices. Then, for a nonempty subset $B \subseteq [2]$, we define the following horizontal concatenation matrix

$$K_{a,B}^{(t)} = [K_{a,b}^{(t)} : b \in B].$$

Next, we blow up the kernel matrix to get

$$M_{a,b}^{(t)} = \Phi_{\frac{n}{2},a}(K_{a,b}^{(t)}) = \mathbf{I}_{s^{\frac{n}{2}-a-1}} \otimes (\mathbf{I}_{s^a} \boxtimes K_{a,b}^{(t)}).$$

Similarly, we define $M_{a,B}^{(t)}$ as that of $K_{a,B}^{(t)}$. Following that, we define

$$\begin{aligned} & f(x_{[2s]}, \gamma) \\ =& \det\left[(V_0 \otimes \mathbf{1}^{(2)}) \odot \mathcal{K}^{(2)}(x_{[s]}) \ \ (V_1 \otimes \mathbf{1}^{(2)}) \odot \mathcal{K}^{(2)}(x_{s+[s]})\right] \\ =& \det\begin{bmatrix} \gamma L(x_0)^{(2)} & L(x_1)^{(2)} & \cdots & L(x_{s-1})^{(2)} & L(x_s)^{(2)} & & & \\ L(x_0)^{(2)} & \gamma L(x_1)^{(2)} & \cdots & L(x_{s-1})^{(2)} & & L(x_{s+1})^{(2)} & & \\ \vdots & \vdots & \ddots & \vdots & & & \ddots & \\ L(x_0)^{(2)} & L(x_1)^{(2)} & \cdots & \gamma L(x_{s-1})^{(2)} & & & & L(x_{2s-1})^{(2)} \end{bmatrix}. \end{aligned}$$

To guarantee the MDS property and the optimal repair scheme, we further require the elements $\lambda_{[sn]}, \gamma$ to satisfy

$$g(\gamma) \cdot \Pi_{a \in [n/2]} f(\lambda_{2sa+[2s]}, \gamma) \neq 0. \tag{4}$$

The existence of such elements in some linear field is guaranteed by the following result.

Lemma 6. *If $q \geq sn+1$, then in $\mathbb{F}_q$ we can always find an element γ and sn distinct elements $\lambda_{[sn]}$ satisfy (4).*

Proof. By $k+1 \leq d \leq n-h$, we have $n \geq k+1+h \geq 3$ because of $k \geq 1$ and $h \geq 1$. Let ω be a primitive element of $\mathbb{F}_q$ with $q \geq sn+1$. Then we set $\lambda_i = \omega^i$ for $0 \leq i \leq sn-1$. We substitute these values and can observe that

$$f(\lambda_{2sa+[2s]}, \gamma) = \omega^{2sa} f(\lambda_{[2s]}, \gamma), \ \ 0 \leq a \leq n/2-1.$$

Write

$$P = \begin{bmatrix} 1 & 0 & & & & & & \\ & & 1 & 0 & & & & \\ & & & & \ddots & & & \\ & & & & & & 1 & 0 \\ -\lambda_s & 1 & & & & & & \\ & & -\lambda_{s+1} & 1 & & & & \\ & & & & \ddots & & & \\ & & & & & & -\lambda_{2s-1} & 1 \end{bmatrix},$$

and

$$Q = \begin{bmatrix} \gamma(\lambda_0 - \lambda_s) & \lambda_1 - \lambda_s & \cdots & \lambda_{s-1} - \lambda_s \\ \lambda_0 - \lambda_{s+1} & \gamma(\lambda_1 - \lambda_{s+1}) & \cdots & \lambda_{s-1} - \lambda_{s+1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_0 - \lambda_{2s-1} & \lambda_1 - \lambda_{2s-1} & \cdots & \gamma(\lambda_{s-1} - \lambda_{2s-1}) \end{bmatrix}.$$

We can check that

$$P\left[(V_0 \otimes \mathbf{1}^{(2)}) \odot \mathcal{K}^{(2)}(x_{[s]}) \ \ (V_1 \otimes \mathbf{1}^{(2)}) \odot \mathcal{K}^{(2)}(x_{s+[s]})\right] = \left[\begin{array}{c|c} \mathbf{rot}\,(F_0) & \mathbf{I}_s \\ \hline Q & \mathbf{O} \end{array}\right].$$

Hence

$$\begin{aligned} f(\lambda_{[2s]}, \gamma) &= \det(P)^{-1} \det\left[\begin{array}{c|c} \mathbf{rot}\,(F_0) & \mathbf{I}_s \\ \hline Q & \mathbf{O} \end{array}\right] \\ &= (-1)^{\frac{s(s+1)}{2}} \det\begin{bmatrix} \gamma(1-\omega^s) & \omega - \omega^s & \cdots & \omega^{s-1} - \omega^s \\ 1 - \omega^{s+1} & \gamma(\omega - \omega^{s+1}) & \cdots & \omega^{s-1} - \omega^{s+1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 - \omega^{2s-1} & \omega - \omega^{2s-1} & \cdots & \gamma(\omega^{s-1} - \omega^{2s-1}) \end{bmatrix} \\ &= (-1)^{\frac{s(s+1)}{2}} \omega^{\frac{s(s-1)}{2}} \det\begin{bmatrix} \gamma(1-\omega^s) & 1 - \omega^{s-1} & \cdots & 1 - \omega \\ 1 - \omega^{s+1} & \gamma(1 - \omega^s) & \cdots & 1 - \omega^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 - \omega^{2s-1} & 1 - \omega^{2s-2} & \cdots & \gamma(1 - \omega^s) \end{bmatrix}. \end{aligned}$$

If we regard $f(\lambda_{[2s]},\gamma)$ as a polynomial in $\mathbb{F}_q[\gamma]$, then $\deg(f)=s$. Write $F(\gamma)=g(\gamma)f(\lambda_{[2s]},\gamma)$. Note that the condition (4) is equivalent to $F(\gamma)\neq 0$. We see that $F(\gamma)$ is a non-zero polynomial in γ with degree at most $s+4$. As $q\geq sn+1$, we can find an element in $\mathbb{F}_q$ such that $F(\gamma)$ is non-zero, and we assign it to γ. This concludes our proof. □

From now, let $\mathbb{F}_q$ be a finite field with order $q\geq sn+1$. Then by Lemma 6 we can select sn distinct elements $\lambda_{[sn]}$ and one element γ satisfying (4) from $\mathbb{F}_q$. Now we write $L_i^{(t)}=L^{(t)}(\lambda_i)$. Then we have the following.

Lemma 7. *Suppose that $a\in[n/2]$, $B\subseteq[2]$ is a nonempty set of size t. For any integer $m>t$, there exists an $\tilde{\ell}m\times\tilde{\ell}m$ matrix V such that:*

(i)
$$VM_{a,B}^{(m)}=\begin{bmatrix} M_{a,B}^{(t)} \\ \mathbf{O} \end{bmatrix}$$
where $\mathbf{O}$ is the $\tilde{\ell}(m-t)\times\tilde{\ell}t$ all-zero matrix.

(ii) *For any $c\in[n/2]\setminus\{a\}, d\in[2]$,*
$$VM_{c,d}^{(m)}=\begin{bmatrix} M_{c,d}^{(t)} \\ \widehat{M}_{c,d}^{(m-t)} \end{bmatrix}$$
where $\widehat{M}_{c,d}^{(m-t)}$ is an $\tilde{\ell}(m-t)\times\tilde{\ell}$ matrix which is column equivalent to $M_{c,d}^{(m-t)}$.

(iii) *For any $\lambda_{i_0},\cdots,\lambda_{i_{s-1}}\notin\{\lambda_{s(2a+b)+x}:b\in B,x\in[s]\}$,*
$$\begin{aligned}&V(\mathbf{I}_{\tilde{\ell}/2}\boxtimes \mathrm{blkdiag}(L_{i_0}^{(m)},\cdots,L_{i_{s-1}}^{(m)})\\ =&\begin{bmatrix} \mathbf{I}_{\tilde{\ell}/2}\boxtimes \mathrm{blkdiag}(L_{i_0}^{(t)},\cdots,L_{i_{s-1}}^{(t)}) \\ (\mathbf{I}_{\tilde{\ell}/2}\boxtimes \mathrm{blkdiag}(L_{i_0}^{(m-t)},\cdots,L_{i_{s-1}}^{(m-t)})Q \end{bmatrix}\end{aligned}$$
[3]*where Q is an $\tilde{\ell}\times\tilde{\ell}$ invertible matrix.*

Lemma 8. *For any z distinct integers $a_0,a_1,\cdots,a_{z-1}\in[n/2]$ and any z nonempty subsets $B_0,B_1,\cdots,B_{z-1}\subseteq[2]$ satisfying $|B_0|+|B_1|+\cdots+|B_{z-1}|=m\leq r$, we have*
$$\det\begin{bmatrix} M_{a_0,B_0}^{(m)} & M_{a_1,B_1}^{(m)} & \cdots & M_{a_{z-1},B_{z-1}}^{(m)}\end{bmatrix}\neq 0.$$

Since the proof of Lemmas 7-8 is exactly the same as that of [12, Lemma 3, Lemma 7], we omit the details.

Before giving the construction of our cooperative MSR code, we define an intermediate $(n,k,\tilde{\ell})$ array code
$$\widetilde{\mathcal{C}}=\{(\widetilde{C}_0,\ldots,\widetilde{C}_{n-1}):\sum_{i\in[n]}\widetilde{H}_i\widetilde{C}_i=\mathbf{0},\widetilde{C}_i\in\mathbb{F}_q^{\tilde{\ell}}\},\tag{5}$$
where $\widetilde{H}_{2a+b}=M_{a,b}^{(r)}$ for $a\in[n/2],b\in[2]$. Note that if we set $m=r$ in Lemma 8, then we obtain the MDS property of the array code (5).

Lemma 9. *The code $\widetilde{\mathcal{C}}$ in (5) is an $(n,k,\tilde{\ell})$ MDS array code.*

Remark 1. *The $(n,k,\tilde{\ell})$ MDS array code $\widetilde{\mathcal{C}}$ in (5) is in fact an MSR code with repair degree $d=s+k-1$. This can be proved similarly by the method of [12].*

Finally, we give the construction of our cooperative MSR code as
$$\mathcal{C}=\{(C_0,\ldots,C_{n-1}):\sum_{i\in[n]}H_iC_i=\mathbf{0},C_i\in\mathbb{F}_q^{\ell}\}\tag{6}$$
where $H_i=\mathbf{I}_{s+h-1}\otimes\tilde{H}_i$ for $i\in[n]$. In other words, we replicate the $(1,d)$-MSR code $\widetilde{\mathcal{C}}$ $s+h-1$ times, obtaining an (h,d)-MSR code.

Lemma 10. *The code $\mathcal{C}$ in (6) is an (n,k,ℓ) MDS array code.*

Proof. This follows directly from the fact that $\widetilde{\mathcal{C}}$ is an MDS array code and $H_i=\mathbf{I}_{s+h-1}\otimes\tilde{H}_i$ for $i\in[n]$. □

[3]Given matrices $A_i,i\in[s]$, $\mathrm{blkdiag}(A_i:i\in[s])$ is the block diagonal matrix obtained by aligning the matrices $A_i,i\in[s]$ along the diagonal.

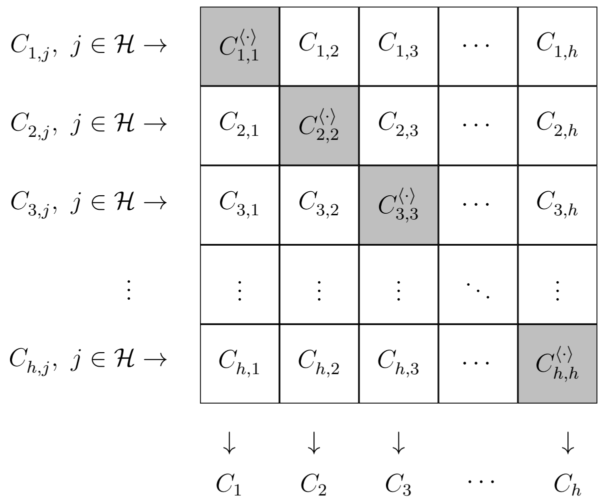

Fig. 1. The repair scheme of our cooperative MSR codes. Without loosing of generality, we assume that $\mathcal{F} = \{1, 2, \ldots, h\}$, and $\mathcal{H} \subseteq [n] \setminus \mathcal{F}$. For each $i \in \mathcal{F}$, we have $C_{i,i}^{\langle g\rangle} = D_{i,i}^{\langle g\rangle} C_i, g \in [s]$, and $C_{i,j} = D_{i,j} C_j, j \in [n] \setminus \{i\}$. Here, for each $i \in \mathcal{F}$, we use $C_{i,i}^{\langle\cdot\rangle}$ to denote the s nodes $C_{i,i}^{\langle 0\rangle}, \cdots, C_{i,i}^{\langle s-1\rangle}$. All the off-diagonal nodes at the ith column will be transmitted to node C_i.

IV. Repair scheme for any h failed nodes

In this section, we describe the cooperative repair scheme of $\mathcal{C}$ defined in (6). Let $\mathcal{F} = \{i_0, i_1, \cdots, i_{h-1}\} \subset [n]$ be the indices of any h failed nodes, where $i_0 < i_1 < \cdots < i_{h-1}$. This naturally induces a bijective map $\mathcal{I}_{\mathcal{F}} : \mathcal{F} \rightarrow [h]$ which maps i_z to z for $z \in [h]$. For simplicity, we write $\hat{i} = \mathcal{I}_{\mathcal{F}}(i)$ for $i \in \mathcal{F}$, i.e., $\hat{i}$ is the index of i in $\mathcal{F}$. Let $\mathcal{H} \subset [n] \backslash \mathcal{F}$ be the collection of the indices of any d helper nodes.

For $a \in [n/2], g \in [s]$, we first introduce the following $\tilde{\ell}/s \times \tilde{\ell}$ row-selection matrix

$$R_{a,g} = \mathbf{I}_{s^{n/2-a-1}} \otimes \boldsymbol{e}_g \otimes \mathbf{I}_{s^a}$$

where $\boldsymbol{e}_g$ is the g-th row of $\mathbf{I}_s$. Multiplying an $\tilde{\ell} \times \tilde{\ell}$ matrix M from the left by $R_{a,g}$ is equivalent to selecting those rows in M whose indices i satisfying that $i_a = g$. We can verify that

$$\sum_{g \in [s]} R_{a,g}^T R_{a,g} = \mathbf{I}_{\tilde{l}}. \tag{7}$$

Then, for $a \in [n/2]$, $g \in [s]$ and $z \in [h]$, we define the following $s \times (s + h - 1)$ block matrix

$$S_{a,g,z}(i,j) = \begin{cases} R_{a,g\oplus_s i} & \text{if } j = i \text{ or } j = z + s \\ \mathbf{O} & \text{otherwise,} \end{cases} \tag{8}$$

where $i \in [s]$, $j \in [s + h - 1]$. Note that for $z = h - 1$, the case $j = z + s$ is impossible. Simply put, for $z \in [h - 1]$,

$$\begin{array}{cc} & \begin{array}{c} (z+s)\text{-th block column} \\ \downarrow \end{array} \\ S_{a,g,z} = & \begin{bmatrix} R_{a,g\oplus_s 0} & \cdots & \mathbf{O} & \mathbf{O} \cdots \mathbf{O} & R_{a,g\oplus_s 0} & \mathbf{O} \cdots \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \ddots \vdots & \vdots & \vdots \ddots \vdots \\ \mathbf{O} & \cdots & R_{a,g\oplus_s (s-1)} & \mathbf{O} \cdots \mathbf{O} & R_{a,g\oplus_s (s-1)} & \mathbf{O} \cdots \mathbf{O} \end{bmatrix} \end{array} \tag{9}$$

and for $z = h - 1$,

$$S_{a,g,h-1} = \begin{bmatrix} R_{a,g\oplus_s 0} & \cdots & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \cdots & R_{a,g\oplus_s (s-1)} & \mathbf{O} & \cdots & \mathbf{O} \end{bmatrix}. \tag{10}$$

For any failed node $i \in \mathcal{F}$, we define the following *repair matrix*

$$\mathcal{R}_i^{\mathcal{F}} = S_{\lfloor \frac{i}{2} \rfloor, 0, \hat{i}} (\mathbf{I}_{s+h-1} \otimes \Phi_{\frac{n}{2}, \lfloor \frac{i}{2} \rfloor}(U_{i \bmod 2})).$$

Note that $\lfloor \frac{i}{2} \rfloor$ is the group's index of node i, and $\hat{i}$ is the index of i in $\mathcal{F}$. We also set the following notations.

Algorithm 1: `repair`$(\mathcal{F},\mathcal{H})$

Input: Two subsets $\mathcal{F},\mathcal{H}\subseteq[n]$ of size $[\mathcal{F}]=h, |\mathcal{H}|=d$ and $\mathcal{F}\cap\mathcal{H}=\emptyset$, which collect the indices of failed nodes and the indices of helper nodes respectively.

Output: The repaired nodes $\{C_i, i\in\mathcal{F}\}$

for $i\in\mathcal{F}$ **do**
  for $j\in\mathcal{H}$ **do**
    Node j computes $C_{i,j}=D_{i,j}C_j$
    Node j **transmits** $C_{i,j}$ to node i
  Node i computes
$$\{C_{i,i}^{\langle g\rangle}, g\in[s], C_{i,j}, j\in\mathcal{F}\setminus\{i\}\}$$
  from the received data $\{C_{i,j}, j\in\mathcal{H}\}$ ▷ Lemma 11

for $i\in\mathcal{F}$ **do**
  for $j\in\mathcal{F}\setminus\{i\}$ **do**
    Node j **transmits** $C_{j,i}$ to node i
  Node i repairs C_i from
$$\{C_{i,i}^{\langle g\rangle}, g\in[s], C_{j,i}, j\in\mathcal{F}\setminus\{i\}\}$$
  ▷ Lemma 12

return $\{C_i, i\in\mathcal{F}\}$

(1) For $g\in[s]$, we define
$$H_{i,i}^{\langle g\rangle}=(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_iS^T_{\lfloor\frac{i}{2}\rfloor,g,h-1}, \tag{11}$$
$$D_{i,i}^{\langle g\rangle}=S_{\lfloor\frac{i}{2}\rfloor,g,\hat{i}}, \tag{12}$$
$$C_{i,i}^{\langle g\rangle}=D_{i,i}^{\langle g\rangle}C_i. \tag{13}$$

(2) For $j\in[n]\setminus\{i\}$ with $\lfloor\frac{j}{2}\rfloor=\lfloor\frac{i}{2}\rfloor$, we define
$$H_{i,j}=(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jS^T_{\lfloor\frac{i}{2}\rfloor,0,h-1}, \tag{14}$$
$$D_{i,j}=S_{\lfloor\frac{i}{2}\rfloor,0,\hat{i}}, \tag{15}$$
$$C_{i,j}=D_{i,j}C_j. \tag{16}$$

(3) For $j\in[n]\setminus\{i\}$ with $\lfloor\frac{j}{2}\rfloor\neq\lfloor\frac{i}{2}\rfloor$, we define
$$H_{i,j}=(S_{\lfloor\frac{j}{2}\rfloor,0,\hat{i}}\otimes\mathbf{I}_r)H_jS^T_{\lfloor\frac{i}{2}\rfloor,0,h-1}, \tag{17}$$
$$D_{i,j}=\mathcal{R}_i^{\mathcal{F}}, \tag{18}$$
$$C_{i,j}=D_{i,j}C_j. \tag{19}$$

The following Lemmas 11-12 will be used in the repair scheme, and their proofs can be find in Appendices.

Lemma 11. *For each $i\in\mathcal{F}$, the following $n+s-1$ matrices*
$$H_{i,0},\ \cdots,\ H_{i,i-1},\ H_{i,i}^{\langle 0\rangle},\ \cdots,\ H_{i,i}^{\langle s-1\rangle},\ H_{i,i+1},\ \cdots,\ H_{i,n-1}$$
define an $(n+s-1,d,\tilde{\ell})$ MDS array code. And for every codeword $(C_0,\dots,C_{n-1})\in\mathcal{C}$ the corresponding vector
$$(C_{i,0},\cdots,C_{i,i-1},C_{i,i}^{\langle 0\rangle},\cdots,C_{i,i}^{\langle s-1\rangle},C_{i,i+1},\cdots,C_{i,n-1})$$
satisfies
$$\sum_{g\in[s]}H_{i,i}^{\langle g\rangle}C_{i,i}^{\langle g\rangle}+\sum_{j\in[n]\setminus\{i\}}H_{i,j}C_{i,j}=\mathbf{0}.$$

Lemma 12. *The $\ell\times\ell$ matrix formed by vertically joining the $s+h-1$ matrices $D_{i,i}^{\langle g\rangle}, g\in[s], D_{j,i}, j\in\mathcal{F}\backslash\{i\}$, is invertible.*

Repair scheme. We illustrate the repair scheme in Fig. 1 and provide the complete steps in Algorithm 1. The repair process is divided into the following two steps.

Step 1. (Row perspective of Fig. 1) For each $i \in \mathcal{F}$, the following steps are executed: Firstly, each helper node $j \in \mathcal{H}$ calculates a vector $C_{i,j} = D_{i,j}C_j$ of length $\tilde{\ell}$ and sends it to node i. Then, by Lemma 11, node i can use the received data $\{C_{i,j}, j \in \mathcal{H}\}$ to compute the $s+h-1$ vectors of length $\tilde{\ell}$, $\{C_{i,i}^{\langle g\rangle}, g \in [s], C_{i,j}, j \in \mathcal{F}\setminus\{i\}\}$. These operations correspond to Lines 1-5 in Algorithm 1.

Step 2. (Column perspective of Fig. 1) For each $i \in \mathcal{F}$, node i can be repaired by the following steps: First, each node $j \in \mathcal{F}\setminus\{i\}$ transmits the length-$\tilde{\ell}$ column vector $C_{j,i}$ computed in Step 1 to node i. Recall that

$$C_{i,i}^{\langle g\rangle} = D_{i,i}^{\langle g\rangle}C_i, g \in [s], C_{j,i} = D_{j,i}C_i, j \in \mathcal{F}\setminus\{i\}.$$

By Lemma 12, C_i can be recovered from $C_{i,i}^{\langle g\rangle}$, $g \in [s]$, and the received data $\{C_{j,i}, j \in \mathcal{F}\setminus\{i\}\}$ from other failed nodes. These operations correspond to Lines 6-9 in Algorithm 1.

It is easy to check that the repair scheme achieves the lower bound of repair bandwidth in Theorem 1. Specifically, the length of each intermediate vector computed during the repair process is $\tilde{\ell} = \ell/(d-k+h)$, and the steps that occupy bandwidth only occur in Line 4 and Line 8 of Algorithm 1. It can be easily calculated that the bandwidth consumed during the repair process is

$$\frac{hd\ell}{d-k+h} + \frac{h(h-1)\ell}{d-k+h}.$$

Here, the left side represents the bandwidth between failed nodes and survival nodes, while the right side represents the bandwidth within the h failed nodes.

V. Conclusion

In this paper, we construct new cooperative MSR codes for any h failed nodes and d helper nodes. The sub-packetization level of our new codes is $(d-k+h)(d-k+1)^{\lceil n/2\rceil}$. We first construct the $(n, k, \tilde{\ell})$ MDS array code $\widetilde{\mathcal{C}}$ in (5) and then replicate $\widetilde{\mathcal{C}}$ $s+h-1$ times, obtaining an (h, d)-MSR code. In general, for any collection of the number of failed nodes $\{h_1, \cdots, h_t\}$, we can replicate $\widetilde{\mathcal{C}}$ $\mathrm{lcm}(d-k+h_1, d-k+h_2, \cdots, d-k+h_t)$ times, obtaining a new cooperative MSR code which can repair any $h \in \{h_1, \cdots, h_t\}$ failed nodes with any d helper nodes and the least possible bandwidth. Furthermore, the sub-packetization of this new code is $\mathrm{lcm}(d-k+h_1, d-k+h_2, \cdots, d-k+h_t)(d-k+1)^{\lceil n/2\rceil}$.

Appendix A
Proof of Lemma 11

The results of Lemma 11 can be divided into the following two lemmas.

Lemma 13. *For each $i \in \mathcal{F}$, the $n+s-1$ matrices of size $r\tilde{\ell} \times \tilde{\ell}$,*

$$H_{i,0}, \ldots, H_{i,i-1}, H_{i,i}^{\langle 0\rangle}, \cdots, H_{i,i}^{\langle s-1\rangle}, H_{i,i+1}, \ldots, H_{i,n-1}$$

defines an $(n+s-1, d, \tilde{\ell})$ MDS array code.

Lemma 14. *For $(C_0, \ldots, C_{n-1}) \in \mathcal{C}$, we have*

$$\begin{aligned}&(R_i^{\mathcal{F}} \otimes I_r)(\sum_{j\in[n]} H_jC_j)\\ &= \sum_{g\in[s]} H_{i,i}^{\langle g\rangle}C_{i,i}^{\langle g\rangle} + \sum_{j\in[n]\setminus\{i\}} H_{i,j}C_{i,j} = \mathbf{0}.\end{aligned}$$

We first need the following technical lemma. The proof of it is exactly the same as that of [12, Lemma 4], and so we omit its proof. Let

$$K = \begin{bmatrix} K_{0,0} & \cdots & K_{0,s-1} \\ \vdots & \ddots & \vdots \\ K_{s-1,0} & \cdots & K_{s-1,s-1}\end{bmatrix}$$

be a $s \times s$ block matrix in which each block entry is a column vector of length r.

Lemma 15. *For any $a, c \in [n/2]$, $b, z \in [s]$, we have*

(i) If $c = a$,

$$(R_{a,b} \otimes \mathbf{I}_r)\Phi_{\frac{n}{2},c}(K)R_{a,z} = \mathbf{I}_{s^{\tilde{c}}} \otimes K_{b,z}.$$

(ii) If $c \neq a$,

$$(R_{a,b} \otimes \mathbf{I}_r)\Phi_{\frac{n}{2},c}(K)R_{a,z} = \begin{cases}\Phi_{\frac{n}{2}-1,\tilde{c}}(K) & \text{if } b = z\\ \mathbf{O} & \text{otherwise.}\end{cases}$$

Here

$$\tilde{c}=\begin{cases} c & \text{if } c<a \\ \frac{n}{2}-1 & \text{if } c=a \\ c-1 & \text{if } c>a. \end{cases}$$

The following result follows directly from the above.

Lemma 16. *For $a,c\in[n/2]$, and $z\in[h]$, we have*

$$\begin{aligned} &(S_{a,0,z}\otimes\mathbf{I}_r)(\mathbf{I}_{s+h-1}\otimes\Phi_{\frac{n}{2},c}(K))S_{a,g,h-1}^T \\ =&\begin{cases} \Phi_{\frac{n}{2},\tilde{c}}(\mathrm{blkdiag}(K_{i,g\oplus_s i}:i\in[s])) & \text{if } a=c \\ \Phi_{\frac{n}{2},\tilde{c}}(K) & \text{if } a\neq c, g=0 \\ \mathbf{O} & \text{if } a\neq c, g\neq 0, \end{cases}\end{aligned}$$

where $\tilde{c}$ is defined in Lemma 15.

Proof. By (9)-(10) we can compute that

$$\begin{aligned} &(S_{a,0,z}\otimes\mathbf{I}_r)(\mathbf{I}_{s+h-1}\otimes\Phi_{\frac{n}{2},c}(K))S_{a,g,h-1}^T \\ =&\,\mathrm{blkdiag}\left((R_{a,i}\otimes\mathbf{I}_r)\Phi_{\frac{n}{2},c}(K)R_{a,g\oplus_s i}):i\in[s]\right). \end{aligned}$$

The rest follows directly from Lemma 15. ☐

A. Proof of Lemma 13

To begin with, we fix some $i\in\mathcal{F}$ and set $i=2a+b$. Therefore $a=\lfloor\frac{i}{2}\rfloor$ and $b=i \bmod 2$. We first give alternative expressions of the $n+s-1$ matrices

$$H_{i,0},\cdots,H_{i,i-1},H_{i,i}^{\langle 0\rangle},\cdots,H_{i,i}^{\langle s-1\rangle},H_{i,i+1},\cdots,H_{i,n-1}. \tag{20}$$

For all $j\in[n]$, let

$$\widetilde{\lfloor\frac{j}{2}\rfloor}=\begin{cases} \lfloor\frac{j}{2}\rfloor & \text{if } \lfloor\frac{j}{2}\rfloor<a \\ \lfloor\frac{n}{2}\rfloor-1 & \text{if } \lfloor\frac{j}{2}\rfloor=a \\ \lfloor\frac{j}{2}\rfloor-1 & \text{if } \lfloor\frac{j}{2}\rfloor>a. \end{cases}$$

1) For any $g\in[s]$, by Lemma 4, we have

$$\begin{aligned} H_{i,i}^{\langle g\rangle}=&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_iS_{a,g,h-1}^T \\ =&(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)(\mathbf{I}_{s+h-1}\otimes\Phi_{\frac{n}{2},a}(K))S_{a,g,h-1}^T \end{aligned}$$

where $K=(U_b\otimes\mathbf{I}_r)K_{a,b}^{(r)}$. Then we can compute that

$$\begin{aligned} K=&(U_b\otimes\mathbf{I}_r)\left((V_b\otimes\mathbf{1}^{(r)})\odot\ \mathcal{K}^{(r)}(\lambda_{si+[s]})\right) \\ =&(U_bV_b\otimes\mathbf{1}^{(r)})\odot\ \mathcal{K}^{(r)}(\lambda_{si+[s]}) \\ =&(\mathbf{rot}\,(F_b)\otimes\mathbf{1}^{(r)})\odot\ \mathcal{K}^{(r)}(\lambda_{si+[s]}) \end{aligned} \tag{21}$$

where $L_i^{(r)}=L^{(r)}(\lambda_i)$. Using Lemma 16, we can compute that for all $g\in[s]$,

$$H_{i,i}^{\langle g\rangle}=c_{b,g}\Phi_{\frac{n}{2},\widetilde{\lfloor\frac{i}{2}\rfloor}}(\mathrm{blkdiag}(L_{si+(g\oplus_s x)}:x\in[s])). \tag{22}$$

where $c_{b,g}$ is the coefficient of x^g in F_b.

2) For $j\in[n]\setminus\{i\}$ with $\lfloor\frac{j}{2}\rfloor=a$, we have $j \bmod 2=b\oplus 1$ and

$$\begin{aligned} H_{i,j}=&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jS_{a,0,h-1}^T \\ =&(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)(\mathbf{I}_{s+h-1}\otimes\Phi_{\frac{n}{2},a}(K))S_{a,0,h-1}^T \end{aligned}$$

where $K=(U_b\otimes\mathbf{I}_r)K_{a,b\oplus 1}^{(r)}$. Then we can compute that

$$\begin{aligned} K=&(U_b\otimes\mathbf{I}_r)\left((V_{b\oplus 1}\otimes\mathbf{1}^{(r)})\odot\ \mathcal{K}^{(r)}(\lambda_{sj+[s]})\right) \\ =&(U_bV_{b\oplus 1}\otimes\mathbf{1}^{(r)})\odot\ \mathcal{K}^{(r)}(\lambda_{sj+[s]}) \\ =&(\mathbf{I}_s\otimes\mathbf{1}^{(r)})\odot\ \mathcal{K}^{(r)}(\lambda_{sj+[s]}). \end{aligned} \tag{23}$$

Using Lemma 16, we can compute that

$$H_{i,j} = \Phi_{\frac{n}{2},\widetilde{\lfloor \frac{j}{2} \rfloor}}(\text{blkdiag}(L_{sj+x} : x \in [s])) \tag{24}$$

$$= \Phi_{\frac{n}{2},\frac{n}{2}-1}(\text{blkdiag}(L_{sj+x} : x \in [s])). \tag{25}$$

3) For $j \in [n] \setminus \{i\}$ with $\lfloor \frac{j}{2} \rfloor \neq a$,

$$\begin{aligned} H_{i,j} =&(S_{a,0,\hat{i}} \otimes \mathbf{I}_r) H_j S^T_{a,0,h-1} \\ =&(S_{a,0,\hat{i}} \otimes \mathbf{I}_r)(\mathbf{I}_{s+h-1} \otimes \Phi_{\frac{n}{2},\lfloor \frac{j}{2} \rfloor}(K)) S^T_{a,0,h-1}. \end{aligned}$$

where $K = K^{(r)}_{\lfloor \frac{j}{2} \rfloor, j \bmod 2}$. And by Lemma 16, we can directly compute that

$$H_{i,j} = \Phi_{\frac{n}{2},\widetilde{\lfloor \frac{j}{2} \rfloor}}(K^{(r)}_{\lfloor \frac{j}{2} \rfloor, j \bmod 2}). \tag{26}$$

From (22), (24), and (26), we can observe that the structure of $n+s-1$ matrices defined in (20) is similar to that of parity check sub-matrices of (5). Using Lemma 7 and the same approach as in Lemma 8, we can prove Lemma 13.

B. Proof of Lemma 14

Lemma 17. *For each $i \in \mathcal{F}$, we write $i = 2a + b$, where $a \in [n/2]$ and $b \in [2]$. Then for any $j \in [n]$, we have*

$$\begin{aligned} &(\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_j C_j \\ =& \begin{cases} \sum_{g \in [s]} \left[(\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_i S^T_{a,g,h-1}\right](S_{a,g,\hat{i}} C_i) & j = i, \\ \left[(\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_j S^T_{a,0,h-1}\right](S_{a,0,\hat{i}} C_j) & j \neq i, \lfloor \frac{j}{2} \rfloor = a, \\ \left[(S_{a,0,\hat{i}} \otimes \mathbf{I}_r) H_j S^T_{a,0,h-1}\right](\mathcal{R}^{\mathcal{F}}_i C_j) & j \neq i, \lfloor \frac{j}{2} \rfloor \neq a. \end{cases} \end{aligned}$$

Proof. Firstly, for $z \in [h]$, we define an $(s+h-1) \times (s+h-1)$ block matrix

$$Q_z(i,j) = \begin{cases} \mathbf{I}_{\tilde{\ell}} & \text{if } i = j \in [s+h-1] \backslash [s] \\ -\mathbf{I}_{\tilde{\ell}} & \text{if } i \in [s],\ j = z+s \\ \mathbf{O} & \text{otherwise}, \end{cases} \tag{27}$$

and we can see that Q_z is an $\ell \times \ell$ matrix. Furthermore, we have the following two conclusions, which can be proved directly by (7), (8) and (27):

1) For any $a \in [n/2]$ and $z \in [h]$,

$$\sum_{g \in [s]} S^T_{a,g,h-1} S_{a,g,z} + Q_z = \mathbf{I}_\ell. \tag{28}$$

2) For any $z \in [h]$, $a \in [n/2]$ and $r\tilde{\ell} \times \tilde{\ell}$ matrix M, we have

$$(S_{a,0,z} \otimes \mathbf{I}_r)(\mathbf{I}_{s+h-1} \otimes M) Q_z = \mathbf{O}. \tag{29}$$

We write $E_{a,b} = \mathbf{I}_{s+h-1} \otimes \Phi_{\frac{n}{2},a}(U_b)$. Then $R^{\mathcal{F}}_i = S_{a,0,\hat{i}} E_{a,b}$.

1) If $j = i$,

$$\begin{aligned} &(\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_j C_j \\ =&(\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_i (\sum_{g \in [s]} S^T_{a,g,h-1} S_{a,g,\hat{i}} + Q_{\hat{i}}) C_i \\ =&\sum_{g \in [s]} \left[(\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_i S^T_{a,g,h-1}\right](S_{a,g,\hat{i}} C_i) \\ &+ (\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_i Q_{\hat{i}} C_i \end{aligned}$$

By (29), we have

$$\begin{aligned} &(\mathcal{R}^{\mathcal{F}}_i \otimes \mathbf{I}_r) H_i Q_{\hat{i}} \\ =&(S_{a,0,\hat{i}} \otimes \mathbf{I}_r)(\mathbf{I}_{s+h-1} \otimes \Phi_{\frac{n}{2},a}(K)) Q_{\hat{i}} \end{aligned}$$

$$=\mathbf{O},$$

where $K=(U_b\otimes\mathbf{I}_r)K_{a,b}^{(r)}$, computed in (21). Therefore,

$$\begin{aligned}&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jC_j\\=&\sum_{g\in[s]}\left[(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_iS_{a,g,h-1}^T\right](S_{a,g,\hat{i}}C_i).\end{aligned}$$

2) For $j\in[n]\backslash\{i\}$ and $\lfloor j/2\rfloor=a$, similarly as above, we have

$$\begin{aligned}&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jC_j\\=&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_j(\sum_{g\in[s]}S_{a,g,h-1}^TS_{a,g,\hat{i}}+Q_{\hat{i}})C_j\\=&\sum_{g\in[s]}\left[(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jS_{a,g,h-1}^T\right](S_{a,g,\hat{i}}C_j)\\&+(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jQ_{\hat{i}}C_j.\end{aligned}$$

Let $K=(U_b\otimes\mathbf{I}_r)K_{a,b\oplus1}^{(r)}$, computed in (23). By (29), we have

$$\begin{aligned}&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jQ_{\hat{i}}\\=&(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)(\mathbf{I}_{s+h-1}\otimes\Phi_{\frac{n}{2},a}(K))Q_{\hat{i}}\\=&\mathbf{O}.\end{aligned}$$

By Lemma 16 we can get that for any $g\in[s]\backslash\{0\}$,

$$\begin{aligned}&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jS_{a,g,h-1}^T\\=&(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)(\mathbf{I}_{s+h-1}\otimes\Phi_{\frac{n}{2},a}(K))S_{a,g,h-1}^T\\=&\mathbf{O}.\end{aligned}$$

Combining the above we have

$$\begin{aligned}&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_j(\sum_{g\in[s]}S_{a,g,h-1}^TS_{a,g,\hat{i}}+Q_{\hat{i}})C_j\\=&\left[(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jS_{a,0,h-1}^T\right](S_{a,0,\hat{i}}C_j).\end{aligned}$$

3) For $j\in[n]\backslash\{i\}$ and $\lfloor j/2\rfloor\neq a$. Using Lemma 3 directly, we have

$$(E_{a,b}\otimes\mathbf{I}_r)H_j=H_jE_{a,b}.$$

Then

$$\begin{aligned}&(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_jC_j\\=&(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)(E_{a,b}\otimes\mathbf{I}_r)H_jC_j\\=&(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)H_jE_{a,b}C_j\\=&(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)H_j(\sum_{g\in[s]}S_{a,g,h-1}^TS_{a,g,\hat{i}}+Q_{\hat{i}})E_{a,b}C_j\\=&\sum_{g\in[s]}\left[(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)H_jS_{a,g,h-1}^T\right](S_{a,g,\hat{i}}E_{a,b}C_j)\\&+(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)H_jQ_{\hat{i}}E_{a,b}C_j.\end{aligned}$$

Because $H_j=\mathbf{I}_{s+h-1}\otimes\Phi_{\frac{n}{2},\lfloor\frac{j}{2}\rfloor}(K_{\lfloor\frac{j}{2}\rfloor,j\bmod 2}^{(r)})$, using Lemma 16 and (29), we have

(i) for any $g\in[s]\backslash\{0\}$,

$$(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)H_jS_{a,g,h-1}^T=\mathbf{O}.$$

(ii) $(S_{a,0,\hat{i}}\otimes\mathbf{I}_r)H_jQ_{\hat{i}}=\mathbf{O}$.

Therefore, we have

$$(\mathcal{R}_i^{\mathcal{F}}\otimes\mathbf{I}_r)H_j(\sum_{g\in[s]}S_{a,g,h-1}^TS_{a,g,\hat{i}}+Q_{\hat{i}})C_j$$

$$= \left[(S_{a,0,\hat{i}} \otimes \mathbf{I}_r) H_j S^T_{a,0,h-1}\right] (S_{a,0,\hat{i}} E_{a,b} C_j)$$

□

In summary we have

$$(R_i^{\mathcal{F}} \otimes \mathbf{I}_r)(\sum_{j\in[n]} H_j C_j) \tag{30}$$

$$= \sum_{j\in[n]} (R_i^{\mathcal{F}} \otimes \mathbf{I}_r) H_j C_j \tag{31}$$

$$\begin{aligned}
= & \sum_{g\in[s]} \left[(\mathcal{R}_i^{\mathcal{F}} \otimes \mathbf{I}_r) H_i S^T_{a,g,h-1}\right] (S_{a,g,\hat{i}} C_i) \\
& + \left[(\mathcal{R}_i^{\mathcal{F}} \otimes \mathbf{I}_r) H_{2a+(b\oplus 1)} S^T_{a,0,h-1}\right] (S_{a,0,\hat{i}} C_{2a+(b\oplus 1)}) \\
& + \sum_{j\in[n]\setminus(2a+[2])} \left[(S_{a,0,\hat{i}} \otimes \mathbf{I}_r) H_j S^T_{a,0,h-1}\right] (\mathcal{R}_i^{\mathcal{F}} C_j)
\end{aligned} \tag{32}$$

$$= \sum_{g\in[s]} H_{i,i}^{\langle g\rangle} C_{i,i}^{\langle g\rangle} + \sum_{j\in[n]\setminus\{i\}} H_{i,j} C_{i,j} \tag{33}$$

$$= \mathbf{0}. \tag{34}$$

Using Lemma 17, we can deduce (32) from (31). By applying notations (11), (14) and (17), we can transform (32) to (33).

Appendix B
Proof of Lemma 12

For any $i \in \mathcal{F}$, we write $i = 2a + b$ where $a \in [n/2]$ and $b \in [2]$. For any $i, j \in \mathcal{F}$ we define

$$P_{j,i} = \begin{cases} \begin{bmatrix} R_{\lfloor \frac{j}{2} \rfloor,0} \\ \vdots \\ R_{\lfloor \frac{j}{2} \rfloor,s-1} \end{bmatrix} & \text{if } \lfloor \frac{j}{2} \rfloor = a, \\ \begin{bmatrix} R_{\lfloor \frac{j}{2} \rfloor,0} \\ \vdots \\ R_{\lfloor \frac{j}{2} \rfloor,s-1} \end{bmatrix} \Phi_{\frac{n}{2},\lfloor \frac{j}{2} \rfloor}(U_{j \bmod 2}) & \text{if } \lfloor \frac{j}{2} \rfloor \neq a, \end{cases}$$

which is an invertible matrix.

We also define that $E_z = \epsilon_z \otimes \mathbf{I}_{\tilde{\ell}}$ where ϵ_z is the z-th row of $\mathbf{I}_{s+h-1}$. We can easily check that the the $\ell \times \ell$ matrix formed by vertically joining the $s + h - 1$ matrices $E_z, z \in [s + h - 1]$, is invertible. For $x, y \in [s]$, set $W_{x,y}$ to be the $s \times s$ block matrix with block entry of size $\tilde{\ell}/s$ and for all $i, j \in [s]$,

$$W_{x,y}(i,j) = \begin{cases} \mathbf{I}_{\tilde{\ell}/s} & i = x, j = y \\ \mathbf{O} & \text{otherwise.} \end{cases} \tag{35}$$

We now split the proof into two cases.

Case 1: $\hat{i} \in [h-1]$. We can see for all $g \in [s]$

$$D_{i,i}^{\langle g\rangle} = \begin{array}{c} \hspace{6em} (\hat{i}+s)\text{-th block column} \\ \hspace{6em} \downarrow \\ \begin{bmatrix} R_{a,g\oplus_s 0} & \cdots & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} & R_{a,g\oplus_s 0} & \mathbf{O} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \cdots & R_{a,g\oplus_s (s-1)} & \mathbf{O} & \cdots & \mathbf{O} & R_{a,g\oplus_s (s-1)} & \mathbf{O} & \cdots & \mathbf{O} \end{bmatrix} \end{array}$$

By performing operations on the rows of the matrices, we can get for $z \in [s]$,

$$M_z := P_{i,i}^{-1} \left(\sum_{g\in[s]} W_{g\oplus_s z, z} D_{i,i}^{\langle g\rangle} \right) = E_z + E_{\hat{i}+s}.$$

Let $k \in \mathcal{F}$ be the failed node with $\hat{k} = h-1$. Then we can check that

$$E_{\hat{i}+s} = P_{k,i}^{-1}\left(\sum_{z\in[s]} W_{z,z}P_{k,i}M_z - D_{k,i}\right)$$

and for all $z \in [s]$,

$$E_z = M_z - E_{\hat{i}+s}.$$

For any $j \in \mathcal{F}\backslash\{i,k\}$, i.e. $\hat{j} \neq h-1, \hat{i}$, we can also check that

$$E_{\hat{j}+s} = P_{j,i}^{-1}\left(D_{j,i} - \sum_{z\in[s]} W_{z,z}P_{j,i}E_z\right).$$

Therefore, we can see that every $E_z, z \in [s+h-1]$ can be written as a linear combination of the $s+h-1$ matrices $D_{i,i}^{\langle g\rangle}, g \in [s], D_{j,i}, j \in \mathcal{F}\backslash\{i\}$. This implies that the $\ell \times \ell$ matrix formed by vertically joining the $s+h-1$ matrices, which includes $D_{i,i}^{\langle g\rangle}, g \in [s], D_{j,i}, j \in \mathcal{F}\backslash\{i\}$, is invertible for all $i \in \mathcal{F}$ satisfying $\hat{i} \in [h-1]$.

Case 2: $\hat{i} = h-1$. In this case, we can see for all $g \in [s]$,

$$D_{i,i}^{\langle g\rangle} = \begin{bmatrix} R_{a,g\oplus_s 0} & \cdots & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \cdots & R_{a,g\oplus_s(s-1)} & \mathbf{O} & \cdots & \mathbf{O} \end{bmatrix}$$

As same as case 1, we can get for all $z \in [s]$,

$$E_z = P_{i,i}^{-1}\left(\sum_{g\in[s]} W_{g\oplus_s z,z}D_{i,i}^{\langle g\rangle}\right).$$

And then for all $j \in \mathcal{F}\backslash\{i\}$, we have

$$E_{\hat{j}+2} = P_{j,i}^{-1}\left(D_{j,i} - \sum_{z\in[s]} W_{z,z}P_{j,i}E_z\right).$$

As above, we can get all E_z for $z \in [s+h-1]$ by linear combination of the $s+h-1$ matrices $D_{i,i}^{\langle g\rangle}, g \in [s], D_{j,i}, j \in \mathcal{F}\backslash\{i\}$ again, which means the $\ell \times \ell$ matrix formed by vertically joining the $s+h-1$ matrices $D_{i,i}^{\langle g\rangle}, g \in [s], D_{j,i}, j \in \mathcal{F}\backslash\{i\}$, is invertible for $\hat{i} = h-1$.

References

[1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[2] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and mbr points via a product-matrix construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[3] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1597–1616, 2013.

[4] I. Tamo, Z. Wang, and J. Bruck, "Access versus bandwidth in codes for storage," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2028–2037, 2014.

[5] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2017.

[6] B. Sasidharan, M. Vajha, and P. V. Kumar, "An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and $d < (n-1)$," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2048–2052, 2017.

[7] M. Ye and A. Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6307–6317, 2017.

[8] I. Tamo, M. Ye, and A. Barg, "Optimal repair of Reed-Solomon codes: Achieving the cut-set bound," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 216–227, 2017.

[9] J. Li, X. Tang, and C. Tian, "A generic transformation to enable optimal repair in MDS codes for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6257–6267, 2018.

[10] S. B. Balaji and P. V. Kumar, "A tight lower bound on the sub-packetization level of optimal-access MSR and MDS codes," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2381–2385, 2018.

[11] N. Wang, G. Li, S. Hu, and M. Ye, "Constructing MSR codes with subpacketization $2^{n/3}$ for $k+1$ helper nodes," *IEEE Transactions on Information Theory*, vol. 69, no. 6, pp. 3775–3792, 2023.

[12] G. Li, N. Wang, S. Hu, and M. Ye, "MSR codes with linear field size and smallest sub-packetization for any number of helper nodes," *arXiv preprint arXiv:2303.10467*, 2023.

[13] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of MDS codes in distributed storage," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.

[14] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, “Centralized repair of multiple node failures with applications to communication efficient secret sharing,” *IEEE Transactions on Information Theory*, vol. 64, no. 12, pp. 7529–7550, 2018.
[15] Z. Wang, I. Tamo, and J. Bruck, “Optimal rebuilding of multiple erasures in MDS codes,” *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1084–1101, 2017.
[16] M. Zorgui and Z. Wang, “Centralized multi-node repair for minimum storage regenerating codes,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2213–2217, 2017.
[17] I. Tamo, M. Ye, and A. Barg, “The repair problem for Reed–Solomon codes: Optimal repair of single and multiple erasures with almost optimal node size,” *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 2673–2695, 2019.
[18] M. Zorgui and Z. Wang, “On the achievability region of regenerating codes for multiple erasures,” in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2067–2071, 2018.
[19] N. Mital, K. Kralevska, C. Ling, and D. Gündüz, “Practical functional regenerating codes for broadcast repair of multiple nodes,” in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 221–225, 2019.
[20] R. Li, J. Lin, and P. P. Lee, “Enabling concurrent failure recovery for regenerating-coding-based storage systems: From theory to practice,” *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1898–1911, 2015.
[21] H. Dau, I. M. Duursma, H. M. Kiah, and O. Milenkovic, “Repairing Reed-Solomon codes with multiple erasures,” *IEEE Transactions on Information Theory*, vol. 64, no. 10, pp. 6567–6582, 2018.
[22] J. Mardia, B. Bartan, and M. Wootters, “Repairing multiple failures for scalar MDS codes,” *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 2661–2672, 2019.
[23] S. Li, M. Gadouleau, J. Wang, and D. Zheng, “A new centralized multi-node repair scheme of MSR codes with error-correcting capability,” *arXiv preprint arXiv:2309.15668*, 2023.
[24] Y. Zhang, “New centralized MSR codes with small sub-packetization,” *ArXiv*, vol. abs/2401.16866, 2024.
[25] K. W. Shum and Y. Hu, “Cooperative regenerating codes,” *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7229–7258, 2013.
[26] A.-M. Kermarrec, N. Le Scouarnec, and G. Straub, “Repairing multiple failures with coordinated and adaptive regenerating codes,” in *2011 International Symposium on Networking Coding*, pp. 1–6, 2011.
[27] J. Li and B. Li, “Cooperative repair with minimum-storage regenerating codes for distributed storage,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 316–324, 2014.
[28] K. W. Shum and J. Chen, “Cooperative repair of multiple node failures in distributed storage systems,” *Int. J. Inf. Coding Theory*, vol. 3, p. 299–323, jan 2016.
[29] M. Ye and A. Barg, “Cooperative repair: Constructions of optimal MDS codes for all admissible parameters,” *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1639–1656, 2019.
[30] Y. Zhang, Z. Zhang, and L. Wang, “Explicit constructions of optimal-access MSCR codes for all parameters,” *IEEE Communications Letters*, vol. 24, no. 5, pp. 941–945, 2020.
[31] M. Ye, “New constructions of cooperative MSR codes: Reducing node size to $\exp(O(n))$,” *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7457–7464, 2020.
[32] Y. Liu, H. Cai, and X. Tang, “A new cooperative repair scheme with $k+1$ helper nodes for (n, k) hadamard MSR codes with small sub-packetization,” *IEEE Transactions on Information Theory*, vol. 69, no. 5, pp. 2820–2829, 2023.