Minimum Storage Regenerating Codes
For All Parameters

Sreechakra Goparaju  Arman Fazeli  Alexander Vardy
University of California San Diego, La Jolla, CA 92037, USA
Email: {sgoparaju, afazeli, avardy}@ucsd.edu

Abstract—Regenerating codes for distributed storage have attracted much research interest in the past decade. Such codes trade the bandwidth needed to repair a failed node with the overall amount of data stored in the network. Minimum storage regenerating (MSR) codes are an important class of optimal regenerating codes that minimize (first) the amount of data stored per node and (then) the repair bandwidth. Specifically, an $[n, k, d]_{\alpha}(a)$ MSR code $C$ over $\mathbb{F}_q$ stores a file $F$ consisting of $a k$ symbols over $\mathbb{F}_q$ among $n$ nodes, each storing $a$ symbols, in such a way that:

- the file $F$ can be recovered by downloading the content of any $k$ of the $n$ nodes; and
- the content of any failed node can be reconstructed by accessing any $d$ of the remaining $n - 1$ nodes and downloading $n/(d - k + 1)$ symbols from each of these nodes.

In practice, the file $F$ is typically available in uncoded form on some $k$ of the $n$ nodes, known as systematic nodes, and the defining node-repair condition above can be relaxed to requiring the optimal repair bandwidth for systematic nodes only. Such codes are called systematic-repair MSR codes.

Unfortunately, finite-$\alpha$ constructions of $[n, k, d]$ MSR codes are known only for certain special cases: either low rate, namely $k/n \leq 0.5$, or high repair connectivity, namely $d = n - 1$. Our main result in this paper is a finite-$\alpha$ construction of systematic-repair $[n, k, d]$ MSR codes for all possible values of parameters $n, k, d$. We also introduce a generalized construction for $[n, k]$ MSR codes to achieve the optimal repair bandwidth for all values of $d$ simultaneously.

Index Terms—distributed storage systems, regenerating codes, interference alignment

I. INTRODUCTION

Distributed storage systems form the backbone for modern cloud computing, large-scale data servers, and peer–to-peer systems. The data in these systems is stored in a redundant fashion — typically via replication (for instance, Hadoop [2] and Google file systems [4] adopt a triple replication policy) — to safeguard data against not-so–infrequently occurring disk failures. An alternative approach to storing data on these systems, which highly reduces the redundancy involved in replication, is to use maximum distance separable (MDS) codes such as Reed–Solomon codes. Though MDS codes are the most space–efficient for a targeted worst–case number of simultaneous node failures, they, unlike repetition codes, incur a high repair bandwidth\(^1\) when the system undergoes the repair of a single node failure. A new class of erasure codes, called regenerating codes, was recently defined by Dimakis et al. [5] over a set of $n$ nodes, which simultaneously optimizes storage efficiency, worst–case resilience and repair bandwidth for single node failures. These codes follow a trade–off curve which is intuitively evidenced by the contrast between repetition codes and MDS codes: the repair bandwidth decreases as the storage redundancy per node increases.

Formally, a file $F$ of size $M$, is said to be stored on a DSS consisting of $n$ nodes, each with a storage capacity of $a$, using an $[n, k, d]_{\alpha}(a)$ (or, in short, $[n, k, d]$) regenerating code, if it satisfies two properties:

(a) data recovery: the file $F$ can be recovered using the contents of any $k$ of the $n$ nodes (this property will also be referred to as the MDS property); and

(b) repair property: the contents of any node can be recovered using the contents of a helper set of any $d$ other helper nodes, where each node transmits $\beta$ number of symbols to the replacement node.

An optimal $[n, k, d]$ regenerating code achieves the optimal value of total repair bandwidth $\gamma = d \beta$ (minimum repair bandwidth) for a given storage capacity $a$ and $M$. This is given implicitly by the following trade–off:

$$M = \sum_{i=0}^{k-1} \min \{ a, (d - i) \beta \}.$$ \hspace{1cm} (1)

Most of the regenerating codes research (e.g. [6]–[15]) is focussed on the extremal points of this trade–off: MBR and MSR codes. Minimum bandwidth regenerating (MBR) codes achieve the optimal $\alpha$ when the repair bandwidth equals that of a repetition code. This paper concerns minimum storage regenerating (MSR) codes, often dubbed as optimal bandwidth MDS codes, because they are optimal regenerating codes that are also MDS codes. For these codes, $\alpha = M/k$, and the optimal repair bandwidth is given by:

$$\beta = \frac{\alpha}{d - k + 1}.$$ \hspace{1cm} (2)

\(^1\)A recent work [4] revisits this for the case of Reed–Solomon codes.

\(^2\)To be precise, these are vector MDS codes, i.e., MDS codes over $\mathbb{F}_q$.\]
It is easy to see that the total repair bandwidth $d\beta$ is optimized when the number of helper nodes $d = n - 1$. However, it is not always practical to connect to all the remaining nodes to aid the repair of a failed node. We therefore consider the following question: Are there constructions of $[n, k, d]$ MSR codes, for $d < n - 1$?

A. Previous Work

This question had not been wholly unanswered before this paper. The first MSR code constructions appeared in [7], [10], which roughly correspond to the family of parameters $\{n, k, d\}$ with rate $k/n \leq 1/2$. The asymptotic existence of MSR codes for all triples $\{n, k, d\}$ was eventually shown in [11] using interference alignment techniques developed for a wireless interference channel; these codes achieve optimality as a regenerating code (as well as approach the MSR point) only when $\alpha \to \infty$, i.e., $\beta/\alpha \to 1/(d - k + 1)$, as $M \to \infty$.

MSR codes, being MDS vector codes, can be expressed as a set of $k$ systematic vectors and $n - k$ parity vectors (the corresponding nodes are referred to as systematic and parity nodes, respectively). For the high-rate ($k/n \geq 1/2$) regime, code constructions were discovered independently in [12–14], [17] for the specific case of $d = n - 1$. Of these, the constructions in [12], [13], [17] focus on the relaxation of restricting optimal repair to systematic nodes in the system; we call the corresponding codes systematic–repair MSR codes. Practical systems usually store information in a systematic format. Parity nodes may fail, but as in the above works, we do not require optimal bandwidth repair for such nodes (maybe they are less urgent or critical). Clearly, any node can be repaired by reconstructing the whole file, so this covers the node repairability (even if suboptimally). Wang et al. construct an MSR code for $d = n - 1$ in [14] that achieves the optimal repair bandwidth also for parity nodes, albeit at the cost of some other metrics such as the number of symbols read from a node (access-optimality) and the complexity of updating parities when systematic data changes (update-optimality). Agarwal et al. in [18], and Raviv et al. in [19] have also presented alternative constructions of access-optimal MSR codes for $d = n - 1$ while achieving the corresponding optimal sub-packetization.

B. Contribution & Outline

We present the first high-rate finite-\(\alpha\) constructions for systematic–repair MSR codes for $d < n - 1$. We start by describing in Section II the representative code construction that contains the ideas behind those in [12], [13], [17]. Leveraging on this, we present our construction in Section III but restrict to the case when the helper nodes contain the remaining $k - 1$ systematic nodes. This restriction is removed in Section IV, thus rounding out the code construction. In Section V, we further generalize the construction to systematic–repair $[n, k]$ MSR codes that universally achieve the optimal repair bandwidth for any given $d$ helper nodes ($k \leq d \leq n - 1$). We conclude with some remarks in Section VI.

C. Subsequent Work

Since this work was first presented, a flurry of recent results have essentially completely solved the problem of constructing $[n, k, d]$ MSR codes. Rawat et al. [21] construct MSR codes which optimally repair all nodes. However, the flavor of their construction, which is not systematic in nature, differs from ours. Most recently, Ye and Barg [22], [23] show that $[n, k, d]$ MSR codes can be explicitly constructed over a small finite field and with a near optimal sub-packetization $\alpha$. Birenjith et al. [24] also construct explicit $[n, k, d = n - 1]$ MSR codes with these properties. An interesting and related direction has been covered in [25] and [26] where MDS codes have been constructed that have a significantly reduced sub-packetization $\alpha$ at the expense of achieving only a near-optimal (and not the optimal) repair bandwidth.

II. Primer: Code Construction for $d = n - 1$

Let $n = k + r$ denote the number of nodes in the distributed storage system, where each node has the capacity to store a vector of size $\alpha$ over $\mathbb{F}_q$. Throughout this paper, we discuss systematic constructions and assume that the first $k$ nodes are systematic nodes and store raw information, while the remaining $r$ nodes correspond to the parities. We use the notation $x_i, i \in [k]$, for the raw systematic vectors stored in the systematic nodes. The parity nodes are defined by

$$x_{k+i} = \sum_{j=1}^{k} A_{ij} x_{j}, \quad i \in [r],$$

where $A_{ij}$'s are $\alpha \times \alpha$ encoding matrices. The generator matrix of the code is then given by

$$G = \begin{bmatrix} I & 0 & \cdots & \cdots & \cdots \\ 0 & I & \cdots & \cdots & \cdots \\ A_{1,1} & \cdots & A_{1,k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{r,1} & \cdots & A_{r,k} \end{bmatrix}. \quad (4)$$

In this section, we consider MSR codes where $d = n - 1$. In other words, when a single node failure occurs, all the remaining nodes aid in its repair. We also restrict our attention to codes that consider failures only of the systematic nodes, and discuss in this section, a construction that underlies the ideas in [13], [17] and [12]. Note

3This work was first presented as an invited talk at the 53rd Annual Allerton Conference on Communication, Control, and Computing. A simultaneous result was presented at the same venue by Tamo and En Gad [20].

4Explicitness implies that the generator or the parity check matrix of the code can be explicitly specified, not merely proved to exist.
that the construction is optimal neither in terms of sub-
packetization for access-optimal MSR codes nor in terms of
field size. However, its simplicity will inform our gen-
eralization for the general parameter triple \( \{n, k, d\} \) in
Section \[III\]

A commonly adopted strategy in constructing an MSR
code is to first guarantee the optimal repair bandwidth
property for a single failure (in this case, for a single sys-
tematic node failure), and then transform the construc-
tion to ensure the MDS property. This is illustrated in
Example \[1\] below.

Example 1. Assume \((n, k, d) = (4, 2, 3)\) and \(\alpha = 4\). Let
the first two nodes \(x_1\) and \(x_2\) be the systematic nodes,
and let the parity nodes \(x_3\) and \(x_4\) be defined as

\[
x_3 = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
x_1 + \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
x_2,
\]

\[
x_4 = \begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}
x_1 + \begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0
\end{pmatrix}
x_2.
\]

Fig. \[1\] (a) depicts the component-wise storage in each
element. It can be observed that a single failure in either
\(x_1\) or \(x_2\) can be reconstructed by downloading \(\alpha/2 = 2\)
elements from each of the remaining \(d = 3\) nodes. How-
ever, the data is not recoverable if both \(x_1\) and \(x_3\) fail and
hence, the code is not MDS. To overcome this problem,
we associate a coefficient \(\lambda\) with \(P_2\) such that \(\begin{pmatrix}
1 & 1 \\
P_1 & \lambda P_2
\end{pmatrix}\),
which is the sub-matrix of the generator matrix \(G\) cor-
responding to the nodes \(x_3\) and \(x_4\), is non-singular. Note
that,

\[
\begin{vmatrix}
I & 1 \\
P_1 & \lambda P_2
\end{vmatrix} = \det(\lambda P_2 - P_1) = \begin{vmatrix}
0 & \lambda & -1 & 0 \\
\lambda & 0 & 0 & -1 \\
-1 & 0 & \lambda & 0 \\
0 & -1 & \lambda & 0
\end{vmatrix} = (\lambda^2 - 1)^2,
\]

which is non-zero\(^5\) if the field size \(q = 5\) and \(\lambda = 2\). Fig. \[1\]
(b) shows the component-wise storage for the resulting
MSR code.

Construction \[2\] generalizes the construction given in
Example \[1\] for an \(\{n, k, n - 1\}\) MSR code. Note that any
MSR code construction must both specify the generator

\[\text{matrix of the code as well as the optimal bandwidth repair strategy that is implemented on the code.}\]

Construction 1. Let \(\alpha = r^k\) and label the \(\alpha\) elements \(\{0 : r^k - 1\}\) by \(r\)-ary vectors in \(\mathbb{Z}_r^k\). Define permutation \(f_j^\ell\) on
\(\{0 : r^k - 1\}\) as follows:

\[
f_j^\ell : \mathbb{Z}_r^k \to \mathbb{Z}_r^k \\
v \mapsto v + \ell e_j,
\]

for \(j \in [k]\) and \(\ell \in [0 : r - 1] =: \{0, 1, \ldots, r - 1\}\), where
\(\{e_1, e_2, \ldots, e_k\}\) is the standard vector basis for \(\mathbb{Z}_r^k\). The mapping \(f_j^\ell\) is bijective, and therefore, corresponds to a
permutation on \(\{0 : r^k - 1\}\). Let \(P_{i,j}\) be the \(\alpha \times \alpha\) matrix corresponding to the permutation \(f_j^\ell\), that is, \(P_{i,j} x = y\),
where \(x, y \in \mathbb{F}_q^r\), and \(x(v) = y(f_j^\ell(v))\). Here, \(x(v)\)
corresponds to the \(v\)th coordinate of vector \(x\), where \(v\) is the
\(k\)-ary representation of a number in \(\{0 : \alpha - 1\}\). In other
words, \(P_{i,j}\) scrambles the elements of a vector according to
the permutation \(f_j^\ell\). (Notice that \(P_{i,j} = I_{\alpha}\))

1) MSR Code: The generator matrix of the code is given by \(G\),
where \(A_{i,j} = \lambda j_i P_{i-1,j_i} i \in [r] \) and \(j \in [k]\). The non-zero coefficients \(\lambda j_i \in \mathbb{F}_q\) will be defined in
Section II-B to ensure the MDS property.

2) Repair Strategy: Let \(Y_j = \{v \in [0 : r^k - 1] : v \cdot e_j = 0\}\)
denote a subset of \(\{0 : r^k - 1\}\). \(Y_j\) can be interpreted as those elements in \(\{0 : \alpha - 1\}\) whose label
representation in \(\mathbb{Z}_r^k\) have a 0 in their \(j\)th coordinate.
If systematic node \(j\) fails, it is repaired by accessing
the elements corresponding to \(Y_j\) from each of the
remaining nodes, i.e., by accessing \(x_i(v)\), where \(v \in Y_j\) and
\(j \neq i \in [n]\).

Construction \[3\] is obtained by first constructing an
\(\{n, k\}\) array code (Section II-A) which guarantees
the optimal bandwidth repair for a single systematic node fail-
ure. The array code is then transformed (Section II-B) to
an MDS array code (and thereby, a systematic-repair
MSR code) by transforming the encoding matrices of the
parity nodes, while retaining the repair property.

\[\text{by an \{n,k\} array code, we mean a set of k systematic vectors, and}
\]
\(n - k\) parity vectors defined according to \(G\), which may or may not
satisfy any properties.

\[
\begin{array}{cccc}
C_1 & C_2 & C_3 & C_4 \\
C_1 & C_2 & C_3 & C_4 \\
\hline \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
\hline \\
\end{array}
\]

\[
\begin{array}{cccc}
C_1 & C_2 & C_3 & C_4 \\
C_1 & C_2 & C_3 & C_4 \\
\hline \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
\hline \\
\end{array}
\]

\[
\begin{array}{cccc}
C_1 & C_2 & C_3 & C_4 \\
C_1 & C_2 & C_3 & C_4 \\
\hline \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
0 & 1 & 1 & + x_1 + x_2 + x_3 + x_4 \\
\hline \\
\end{array}
\]
A. Repair Property: Interference Alignment

The optimal repair bandwidth property of an \([n, k, n-1]\) MSR code can be viewed as a signal interference problem: the objective is to retrieve the desired signal — the contents of the failed systematic node, say, \(x_i\) — which, in the repair data downloaded from the remaining nodes, is interfered by partial contents of the remaining systematic nodes, \(x_j\), for \(i \neq j \in [n]\). The solution, turns out to be an interference alignment strategy, where the repair data associated with the interfering systematic data is aligned, so as to minimize the interference data. This is crystallized in the following lemma.

**Lemma 1.** Let \(x_j, i \in [k]\), be the failed systematic node. For an \([n, k, n-1]\) MSR code, the set of \(d = n-1\) helper nodes is given by \(D = \{x_j | j \in [n] \setminus \{i\}\}\). To recover the contents of the failed systematic node with the optimal repair bandwidth, it is necessary and sufficient to find \(n-1\) (repair) matrices denoted by \(\{S_j\}_{j \in [n]}\), where \(r = n-k\), such that, for \(j \in [k], j \neq i\), the following two conditions are satisfied:

(a) signal recovery:
\[
\text{rank} \left( \begin{pmatrix} S_{i+1}^j A_{1,j} \\ S_{i+2}^j A_{2,j} \\ \vdots \\ S_{i+r}^j A_{r,j} \end{pmatrix} \right) = \alpha, \quad (5)
\]

(b) interference alignment:
\[
\text{rank} \left( \begin{pmatrix} S_i^j \\ S_{i+1}^j A_{1,j} \\ \vdots \\ S_{i+r}^j A_{r,j} \end{pmatrix} \right) = \alpha/r. \quad (6)
\]

Stated otherwise, to optimally repair \(x_i\), it is necessary and sufficient to find \(n-1\) (repair) subspaces of dimension \(\alpha/r\), denoted by \(\{S_j\}_{j \in [n]}\), where \(r = n-k\), such that, for \(j \in [k], j \neq i\), the following two conditions are satisfied:

(a) signal recovery:
\[
S_{i+1}^j A_{1,j} \oplus \cdots \oplus S_{i+r}^j A_{r,j} \subseteq \mathbb{F}_q^\alpha, \quad (7)
\]

(b) interference alignment:
\[
S_i^j \subseteq S_{i+s}^j A_{sj}, \quad \forall s \in [r], \quad (8)
\]

where \(\subseteq\) denotes equality of subspaces, \(S\) is the subspace obtained by operating the subspace \(S\) by the matrix \(A\), and \(\oplus\) denotes the subspace sum.

For completeness, we provide a proof for Lemma 1. Lemma 2 generalizes Lemma 1 when the number of helper nodes \(d < n-1\). This will be used later in Section III.

**Lemma 2.** (Corollary of Lemma 1) In general, for an \([n, k, d]\) MSR code, if the set of \(d = (k-1)+t < n-1\) helper nodes is given by \(D = \{x_j | j \in [k] \setminus \{i\}\} \cup \{b_1, \ldots, b_t\}\) (where \(b_t \in [k+1], \ldots, n\) denote the parity nodes in the helper set), it is necessary and sufficient to find \(d\) (repair) subspaces of dimension \(\alpha/t\) denoted by \(\{S_j\}_{j \in [k]}\), such that, for \(j \in [k], j \neq i\), the following two conditions are satisfied:

\[
S_{i+1}^j A_{1,j} \oplus \cdots \oplus S_{i+r}^j A_{r,j} \subseteq \mathbb{F}_q^\alpha, \quad (9)
\]

\[
S_i^j \subseteq S_{i+s}^j A_{sj}, \quad \forall s \in [t]. \quad (10)
\]

The optimal repair property of Construction 1 can now be justified.

**Lemma 3.** The repair strategy in Construction 1 is optimal with respect to repair bandwidth.

Proof: Define \(S_i^j \triangleq S_i^j \triangleq Y_i, j \neq i\). Notice that the rank of subspace of \(S_i^j\) is \(r^{k-1} = \alpha/r\). Per definition, the permutation \(P_{ij}\) maps \(Y_i\) to \(Y_i + \ell e_i = \{v \in [0, r^{k-1}] : v \cdot e_i = \ell\}\). This implies that for any distinct \(\ell, \ell' \in [0 : r-1]\), the intersection \(S_i^j \cap S_i^{j'}\) contains only the all-zero vector. Thus the subspaces \(S_i^j, S_i^{j'} P_{ij}, \ldots, S_i^{j'} P_{i-1,j'}\) span the space \(\mathbb{F}_q^\alpha (a = r^t)\) and the signal recovery condition(s) in Lemma 1 are satisfied; see Fig. 2 Furthermore, applying a permutation \(P_{ij}\) corresponding to a different coordinate \(j \neq i\) maps \(Y_i\) to itself. This validates the interference alignment condition(s) in Lemma 1. Finally, note that the two conditions continue to be satisfied when replacing the permutations \(P_{i-1,j}\) with any scaled versions \(A_{ij} = \lambda_{ij} P_{i-1,j}\), because the scaling of the basis vectors does not change the relevant subspaces and thereby does not affect the conditions in Lemma 1. \(\Box\)

B. MDS Property

This second step relies on the following two lemmas, the proofs of which are left to the reader.

**Lemma 4.** Let \(B\) denote the parity part of the generator matrix for an \([n, k]\) array code denoted by \(C\), where

\[
B = \begin{bmatrix}
B_{1,1} & \cdots & B_{1,k} \\
\vdots & \ddots & \vdots \\
B_{r,1} & \cdots & B_{r,k}
\end{bmatrix}
\]
Given that $B_{i,j}$ is non-singular for all $i,j$, $C$ is an MDS array code if and only if any square sub-block-matrix $B'$ of $B$ is also non-singular, where

$$B' = \begin{bmatrix} B_{i,j} & \cdots & B_{i,j} \\ \vdots & \ddots & \vdots \\ B_{i,j} & \cdots & B_{i,j} \end{bmatrix},$$

for some $\{i_1, \cdots, i_t\} \subset [r], \{j_1, \cdots, j_t\} \subset [k]$. 

**Lemma 5.** Let $B$ denote the $r \times k\alpha$ matrix associated with the parity part of the generator matrix for an $[n,k]$ array code, as defined in Lemma 4. If $B_{i,j}$ is non-singular for all $i \in [r], j \in [k]$, and the field size $q$ is large enough, there exist coefficients $\lambda_{i,j} \in \mathbb{F}_q$, such that all square sub-block-matrices of $A$ are non-singular, where

$$A = \begin{bmatrix} \lambda_{1,1} B_{1,1} & \cdots & \lambda_{1,k} B_{1,k} \\ \vdots & \ddots & \vdots \\ \lambda_{r,1} B_{r,1} & \cdots & \lambda_{r,k} B_{r,k} \end{bmatrix}.$$

In other words, any parity generator matrix $B$ for an $[n,k]$ array code with non-singular encoding matrices can be transformed into a parity generator matrix $A$ for an $[n,k]$ MDS array code by multiplying the encoding matrices with appropriate scalar coefficients.

**Proof Sketch:** To obtain a valid set of $\lambda_{i,j}$’s, one may first sort the pairs $(i,j)$ with respect to $i+j$ increasingly, and then recursively choose a value for each $\lambda_{i,j}$ such that all sub-block-matrices with $\lambda_{i,j} A_{i,j}$ on their bottom right corner become non-singular. It suffices to have the field size $q$ greater than the number of such sub-block-matrices at any step multiplied by $\alpha$;

$$|\mathbb{F}| > q_{\text{min}} = \alpha \max \left\{ \binom{n-k-1}{t} \times \binom{k-1}{t} : t \in [k] \right\}. \quad (11)$$

**Remark:** Notice that this construction is slightly different than that described in [17], where the authors choose different permutation matrices with a potentially smaller value of $m$ in $\alpha = \alpha^m$. This is achieved by choosing a different set of vectors than $\{v_i, i \in [k]\}$ when defining the permutations in Construction [11]. However, we forego the analysis of reducing $m$ for the sake of simplicity and readability, and take the current construction as a starting point for the generalized construction below.

### III. Code Construction for Restricted Helper Set

We now move to the construction of $[n,k,d]$ systematic-repair MSR codes for any $n$, $k$, and $d$, where $k+1 \leq d \leq n-1$. In this section, we start with the restricted case when the helper set $D$ includes all remaining $k-1$ systematic nodes. Let us begin with an example.

**Example 2.** Let us look at the case when $[n, k, d] = [k+3, k, k+1]$ for $k \in \mathbb{N}$. Given a failure at the systematic node $i$, we are interested in repairing it by downloading $\binom{k}{k-1} = \binom{k}{1} = \frac{k}{2}$ symbols from each node in the helper set $D$. Let us assume that $D_i$ includes all of the remaining $k-1$ systematic nodes. Hence, there are $\binom{k}{1} = 3$ different ways to choose $D_i$ depending on which two parity nodes are included in it. Let us use an indicator $\alpha \in [3]$ to differentiate between these scenarios, and denote the helper set for each scenario by $D_{i,\alpha}$.

**Construction.** Let $\alpha = 2^k$ and label the $\alpha$ elements $[0 : 2^k - 1]$ by binary vectors in $\mathbb{Z}_2^k$. Define permutation $f^j$ on $[0 : 2^k - 1]$ as follows:

$$f^j : \mathbb{Z}_2^k \rightarrow \mathbb{Z}_2^k \quad \nu \mapsto \nu + \ell \epsilon_j,$$

for $j \in [3k]$ and $\ell \in (0,1)$, where $\{\epsilon_1, \epsilon_2, \cdots, \epsilon_{3k}\}$ is the standard vector basis for $\mathbb{Z}_2^k$. The mapping $f^j$ is again bijective and therefore corresponds to a permutation on $[0 : 2^k - 1]$. As before, let $P_{i,j}$ be the $\alpha \times \alpha$ matrix corresponding to the permutation $f^j$, that is, $P_{i,j} x = y$, where $x, y \in \mathbb{F}_q^k$, and $x(\nu) = y(f^j(\nu))$. As before, let $x(\nu)$ be the $v$th coordinate of $x \in \mathbb{F}_q^k$, where $v$ is the k-ary vector representation of the coordinate, and $P_{0,j} = I_{k \alpha}$.

1) MSR Code: The generator matrix of the code is given by

$$G = \begin{bmatrix} I & 0 \\ 0 & I \\ A_{1,1} & \cdots & A_{1,k} \\ A_{2,1} & \cdots & A_{2,k} \\ A_{3,1} & \cdots & A_{3,k} \end{bmatrix},$$

where

$$A_{1,j} = \lambda_{1,j} P_{0,3j-2} P_{0,3j-1},$$

$$A_{2,j} = \lambda_{2,j} P_{1,3j-2} P_{1,3j},$$

$$A_{3,j} = \lambda_{3,j} P_{1,3j-1} P_{1,3j}, \quad (12)$$

for $j \in [k]$. The non-zero coefficients $\lambda_{i,j} \in \mathbb{F}_q$ are again selected according to the discussion in Section II-B to establish the MDS property.

2) Repair Strategy via $D_{i,1} = \{x_j | j \in [k+3], j \neq i, k+3\}$: Let $Y_{i,1} = \{v \in [0 : 2^k - 1] : v \cdot \epsilon_{3i-2} = 0\}$ denote a subset of $[0 : 2^k - 1]$. $Y_{i,1}$ can be interpreted as those elements in $[0 : 2^k - 1]$ whose label representation in $\mathbb{Z}_2^k$ has a 0 in its $(3i-2)$th coordinate. If systematic node $i$ fails, it can be repaired by accessing the elements corresponding to $Y_{i,1}$ from each of the helper nodes, i.e., by accessing $x_j(\nu)$, where $\nu \in Y_{i,1}$ and $j \in \{1, 2, \cdots, i-1, i+1, \cdots, k, k+1, k+2\}$.

3) Repair Strategy via $D_{i,2} = \{x_j | j \in [k+3], j \neq i, k+2\}$: Similarly, let $Y_{i,2} = \{v \in [0 : 2^k - 1] : v \cdot \epsilon_{3i-1} = 0\}$. If systematic node $i$ fails, it can be repaired by accessing $x_j(\nu)$, where $\nu \in Y_{i,2}$ and $j \in \{1, 2, \cdots, i-1, i+1, \cdots, k, k+1, k+3\}$.

4) Repair Strategy via $D_{i,3} = \{x_j | j \in [k+3], j \neq i, k+1\}$: Finally, let $Y_{i,3} = \{v \in [0 : 2^k - 1] : v \cdot \epsilon_{3i} = 0\}$ denote the location of the elements that are accessed if systematic node $i$ fails, i.e., node $i$ can
be the failed systematic node. Define $Y_{i,j}$ to define the proper subspaces pair. Let us label these cases with numbers.

The key element in the construction is to satisfy the two interference alignment conditions in Lemma 2.

Let us for simplicity assume $\rho = 1$, $\alpha = 0$.

Given the code construction in (12), we can verify that

$$Y_{i,j} = \begin{cases} Y_{i,j} & \text{if } j \neq i, \\ Y_{i,j} & \text{if } j = i. \end{cases}$$

Now we define subspaces $S_{i,j}^a \triangleq S_i^a \triangleq Y_{i,j}, j \neq 1, a \in [3]$. Let us for simplicity assume $a = 1$. The other scenarios can be justified similarly. Based on (13), we observe that the permutation $Q_{i,j}$ maps the basis $Y_{i,j}$ to its complementary subset $U_{i,j}$ and vice versa, while $Q_{i,j}$ preserves both of them. Hence,

$$\text{rank} \left( \begin{array}{c} S_{i,j}^1 \\ S_{i,j}^2 \end{array} \right) = \text{rank} \left( \begin{array}{c} Y_{i,j} \\ U_{i,j} \end{array} \right) = \alpha.$$

Furthermore, $Y_{i,j}$ remains unchanged under any other permutation $Q_{i,j}, j \neq i$, and hence

$$\text{rank} \left( \begin{array}{c} S_{i,j}^1 \\ S_{i,j'}^1 \end{array} \right) = \text{rank} \left( \begin{array}{c} Y_{i,j} \\ Y_{i,j'} \end{array} \right) = \alpha / 2.$$

The key element in the construction is to satisfy the two requirements in Lemma 2 for any systematic failure and any such helper set $\mathcal{D}$. Let $r = d - k + 1$ denote the number of parity nodes in the helper set of size $d$. There are $\binom{r}{k}$ different ways to choose $p$ parity nodes during the repair. Let us label these cases with numbers $a \in \binom{[r]}{k}$, and set $\mathcal{R}_a$ to be the subset of parity nodes corresponding to case $a$.

Assume that $\mathcal{R}_a = \{x_{k+d(a)}, x_{k+d_2(a)}, \ldots, x_{k+d_\rho(a)} \}$, where $d_1(a), \ldots, d_\rho(a) \in [r]$. Finally, define $r$-ary vectors $\omega_a$ for $a \in \binom{[r]}{k}$ as

$$\omega_a(i) = \begin{cases} t - 1 & \text{if } \exists t : i = d_t(a), \\ 0 & \text{otherwise}. \end{cases}$$

The non-zero coefficients $\lambda_{i,j}$ are defined according to Section II-B to ensure the MDS property; and later will be modified again in Section IV.

2) Repair Strategy: Let $\mathcal{R}_a$ correspond to the parity subset of the helper set $\mathcal{D}$. Define $Y_{i,j} \subset [0 : \alpha - 1]$ as $\{x \in [0, \alpha - 1] : x \cdot e_{a+(j-1)} \} = 0$. If systematic node $j$ fails, it is repaired by accessing the elements corresponding to $Y_{i,j}$ from helper nodes, i.e., by accessing $x_i(v)$, where $i \in D$, and $v \in Y_{i,j}$.

Lemma 6. The repair strategy in Construction 2 is optimal with respect to repair bandwidth.

Proof: Let us first explain the role of $\omega_a$ by revisiting Example 2 via (12). Note that the matrix $P_{0,x}$ for any $x$ is also equal to the identity matrix $I_a$. Here we assumed that

$a = 1 \rightarrow \mathcal{R}_1 = \{x_{k+1}, x_{k+2} \} \rightarrow \omega_1 = (0, 1, 0)^t,$

$a = 2 \rightarrow \mathcal{R}_2 = \{x_{k+1}, x_{k+3} \} \rightarrow \omega_2 = (0, 0, 1)^t,$

$a = 3 \rightarrow \mathcal{R}_3 = \{x_{k+2}, x_{k+3} \} \rightarrow \omega_3 = (0, 0, 1)^t.$

In general, the matrix $\Omega = \{\omega_1 | \omega_2 | \cdots | \omega_{\binom{r}{k}}\}$ is designed in a way that for any choice of $a \in \binom{[r]}{k}$ we can always find a column in $\Omega$, denoted by $\omega_a$, such that its intersection with $\rho$ rows associated with scenario $a$, forms $\{0, 1, \ldots, \rho - 1 \}$.

Now assume that systematic node $i$ fails and we are to perform an optimal repair given that the parity nodes in the helper set are in $\mathcal{R}_a = \{x_{k+d(a)}, x_{k+d_2(a)}, \ldots, x_{k+d_\rho(a)} \}$. 

$$\begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} P_{0,3j-2} & P_{0,3j-1} & I_a \end{bmatrix}$$

Fig. 3. Relation between $\alpha$ and $\omega_a$. 

Construction 2. Let $a = \rho \binom{r}{k}$ and label the $\alpha$ elements of $[0: \alpha - 1]$ by $\rho$-ary vectors in $\mathbb{Z}_\rho^k$. Define permutation $f_j$ on $[0: \alpha - 1]$ as follows:

$$f_j : \mathbb{Z}_\rho^k \rightarrow \mathbb{Z}_\rho^k,$$

for $j \in [\binom{r}{k}]$, and $\ell \in [0: \rho - 1]$, where $\{e_{1}, \cdots, e_{\binom{r}{k}}\}$ is the standard vector basis of $\mathbb{Z}_\rho^k$. Let $P_{i,j}$ be the $\alpha \times \alpha$ matrix corresponding to the permutation $f_j$.

1) MSR Code: The generator matrix of the [n, k, d] code is given by (4), where

$$A_{i,j} = \lambda_{i,j} \prod_{a \in [\binom{r}{k}]} P_{wa(i,a+(j-1))}, \text{ for } j \in [k], i \in [r].$$

Lemma 5 and 6.
Define $Q_{\alpha,\beta} = \lambda_{\alpha,\beta}^{-1}A_{\alpha,\beta}$; this is a product of multiple permutation matrices, and hence can be viewed as a permutation matrix itself. It again suffices to define the proper subspaces $S^i_j$ that fulfill the two interference alignment conditions in Lemma 2. Let us select our subspaces as $S^i_j = S^i_{\alpha} \triangleq \lambda_{\alpha}^{-1}Y_{i,\alpha} = \{x| x \cdot e_{\alpha + (i-1)(\alpha)} = 0\}$, we have

$$Y_{i,\alpha}Q_{d_1,\alpha} = \{x| x \cdot e_{\alpha + (i-1)(\alpha)} = 0\},$$

$$Y_{i,\alpha}Q_{d_2,\alpha} = \{x| x \cdot e_{\alpha + (i-1)(\alpha)} = 1\},$$

...

$$Y_{i,\alpha}Q_{d_\rho,\alpha} = \{x| x \cdot e_{\alpha + (i-1)(\alpha)} = \rho - 1\},$$

and hence,

$$\operatorname{rank} \begin{pmatrix} S_{i,\alpha}^1Q_{d_1,\alpha}^1 \\ S_{i,\alpha}^2Q_{d_2,\alpha}^1 \\ \vdots \\ S_{i,\alpha}^\rho Q_{d_\rho,\alpha}^1 \end{pmatrix} = \rho \times \operatorname{rank}(Y_{i,\alpha}) = \frac{\rho}{\rho} = \alpha.$$

The second condition in Lemma 2 is also automatically satisfied since

$$Y_{i,\alpha} \simeq Y_{i,\beta}Q_{1,i'} \simeq Y_{i,\beta}Q_{2,i'} \simeq \cdots \simeq Y_{i,\beta}Q_{\rho,i'} \quad \text{for } i' \neq i.$$

Remark: Note that the sub-packetization $\alpha$ can be optimized further, but it is not the goal in this paper. Although Construction 2 suggests a fairly large value, i.e. $\alpha = \rho^k$, it is clear that we do not need $\rho^k$ columns in $\Omega$ to cover all the $\binom{\rho}{k}$ helper set selection scenarios. Indeed, $\alpha$ in Example 2 can be reduced to $2^k$, where $\Omega_{\text{new}} = \{\omega_1|\omega_2\}$. We leave the optimization strategies of this kind of future work. We also refer the reader to [22–24], where MSR constructions with near-optimal sub-packetization parameter, e.g. $\alpha = r\frac{n}{\sqrt{r}}$, are introduced. A different approach to solve this problem is presented in [27], which is a method to reduce the sub-packetization of the existing constructions dramatically in exchange for a slight increase in repair bandwidth.

IV. Code Construction for any Helper Set

In this section, we show that Construction 2 in fact holds, even when an arbitrary set of $d$ helper nodes is allowed to be chosen from the $(n-1)$ surviving nodes. This generality merely imposes some additional constraints on the selection of the scaling coefficients $\lambda_{i,j}$ of the encoding matrices $A_{i,j} = \lambda_{i,j}Q_{i,j}$, where $Q_{i,j}$ is the (product) permutation matrix corresponding to $A_{i,j}$ as defined in Construction 2. We now arrive at the main theorem.

Theorem 7. Construction 2 gives an $[n,k,d]$ systematic-repair MSR code for any set of $d$ helper nodes, for a large enough field size for the scaling coefficients $\lambda_{i,j}$ for the encoding matrices $A_{i,j}$.

Proof: Part 1: First, we illustrate the proof by fixing $d = k + 1$, and taking an example set of helper nodes for an example failure of node $x_1$ (or node 1). Let us denote the (indices of the) helper set by $D$, and let $D = \{h_1, h_2, \ldots, k, k+1, \ldots, k+h\}$, that is, there are $h$ parity nodes and $d - h = k + 1 - h$ systematic nodes in the helper set. Let $S(D)x_i$ denote the repair information that node $i$ sends to help in the repair of node $j$ when $D$ is the set of helper nodes. (Wherever clear, we ignore the $D$ in the notation and simply write $S_i$.) When node 1 fails, the information we therefore have at its replacement node can be written as:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_h \\ x_{h+1} \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} S_h^1 \\ S_{h+1}^1 \\ \vdots \\ S_k^1 \\ S_{k+1}^1A_{11} \\ \vdots \\ S_{k+1}^1A_{1k} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_h \\ x_{h+1} \\ \vdots \\ x_k \end{pmatrix}.$$  \hspace{1cm} (14)

Suppose all $S^1$’s in (14) be replaced by a repair subspace $S^1$ (corresponding to Lemma 2(b)) that we would have used if $D = \{2,3,\ldots,k,k+1,k+2\}$. Specifically, suppose $S^1A_{i,1}$ and $S^1$ complete the space $\mathbb{F}_q^n$. Since $S^1$ and $S^1A_{i,j}$ denote the same subspace, for $j \neq 1$, the components of $x_i, i \in \{h_1,h_2,\ldots,k\}$ can be easily subtracted from the information coming from the parity nodes, using that coming from the systematic nodes $h$ to $k$. Thus, in order to recover $x_1$, we can concentrate on the following information at the replacement node:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{h-2} \\ x_{h-1} \end{pmatrix} = \begin{pmatrix} S^1A_{11} \\ S^1A_{12} \\ \vdots \\ S^1A_{1,h-2} \\ S^1A_{1,h-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{h-2} \\ x_{h-1} \end{pmatrix}.$$  \hspace{1cm} (15)

Let $S^1A_{i,j} = \lambda_{i,j}S^1Q_{i,j} = \lambda_{i,j}\tilde{Q}_{i,j}S^1$, where $\tilde{Q}_{i,j}$ is an $\alpha/2 \times \alpha/2$ matrix, and $(i,j) \neq (1,1)$. It must be noted that not only is $S^1$ dependent on the choice of $D$, but so in turn is $\tilde{Q}_{i,j}$. Let us also denote $S^1x_i$ by $\tilde{x}_i$. Then, (15) can be rewritten as:

$$\begin{pmatrix} \lambda_{1,1}S^1Q_{1,1} \\ \lambda_{1,2}S^1Q_{1,2} \\ \vdots \\ \lambda_{1,h-2}S^1Q_{1,h-2} \\ \lambda_{1,h-1}S^1Q_{1,h-1} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{h-2} \\ x_{h-1} \end{pmatrix}.$$  \hspace{1cm} (16)

The matrix in (16) — call it $M$ — is a square matrix of dimensions $ha/2 \times ha/2$. A sufficient condition to recover $x_1$ is that $M$ is invertible. Notice that the determinant of $M$, det$(M)$, is a polynomial in the following variables: $\lambda_{i,j}, i \in [h], j \in [h-1]$. Hence, det$(M)$ is a nonzero
polynomial of degree $ha/2$ in the given variables. From Schwartz–Zippel–DeMilllo–Lipton lemma \[28\], if the finite field $\mathbb{F}_q$ over which the determinant is defined has cardinality $|\mathbb{F}_q| = q > ha/2$, there exist $\lambda_{i,j}$’s for which the determinant det(M) above is nonzero.

**Part 2:** Notice that $M$ above is defined for a particular example scenario. In general, let the number of helper nodes be $d$, the failed systematic node be $f \in [k]$, the set of helper nodes by $D \subseteq [n] \setminus \{f\}$, the set of systematic helper nodes be $D_s \subseteq [k] \setminus \{f\}$, and the set of parity helper nodes be $D_p \subseteq [k + 1 : k + r]$. Let the number of parity helper nodes be denoted by $h$, where $h$ ranges from $d - k + 1$ to $r$. Let us represent by $H_p$ the set of parity helper nodes but indexed within $[r]$, where $i$ corresponds to node $k+i$ of the system, that is, $H_p = \{i \mid k+i \in D_p\} \subseteq [r]$.

The matrix $M$ in \[16\], in general, can be seen to be a square matrix of dimensions $ha/(d-k+1) \times ha/(d-k+1)$. In particular, $M$ is a function of $f, D_s, \text{ and } H_p$, and the determinant polynomial has degree which is a function of $|D_p| = h$ and $d$. For each $f, D_s$ and $H_p$, we obtain a sufficiency condition that the corresponding $M$ is invertible. Therefore, the product of the corresponding determinant polynomials is a nonzero polynomial of degree

$$q_{\text{ANY}} = k \left( \sum_{h=d-k+1}^{r} \frac{r}{h} \frac{k-1}{d-h} \frac{h_a}{d-k+1} \right)$$

$$= \left( \sum_{h=d-k+1}^{r} \frac{r}{h} \frac{k-1}{d-h} \right) \frac{k_a}{d-k+1}
$$

(17)

consequently, there exist $\lambda_{i,j}$’s in $\mathbb{F}$ such that any systematic node is repairable with optimal repair bandwidth using any arbitrary set of $d$ helper nodes, as long as the field size $|\mathbb{F}| > q_{\text{ANY}}$.

**Part 3:** Finally, using Lemma \[4\], Lemma \[5\] and Lemma \[6\] we obtain an $[n,k,d]$ systematic–repair MSR code for any set of $d$ helper nodes, when the field size $q > q_{\text{ANY}} + q_{\text{MDS}}$.

**Remark:** As shown in \[17\], the sufficiency conditions on the field size such as equation (11) are not always necessary. Indeed, the authors in \[17\] show that for small values of $r$ there exist smart choices of coefficients $\lambda_{i,j}$ such that a field size of 3 or 4 would suffice. However for arbitrary values of $r$, their construction also follows a combinatorial Nullstellensatz argument to prove the existence of such codes. Note that for the case of $d = n - 1$, there is only one selection of helper sets, and hence condition \[17\] becomes unnecessary.

Table I provides a comparison of these sufficiency conditions for a select set of code parameters. The calculation of $q_{\text{MDS}}$ and $q_{\text{ANY}}$ assumes a large value for $n$. The field size is only given based on sufficiency conditions in equations \[11\] and \[17\], which is based on an existence argument. However as shown in \[17\], the explicit and optimized constructions of MSR codes may require much smaller field sizes. The last row in the table corresponds to the all-$d$ construction in Example \[5\].

### Table I

| $[n,k,d]$ | $\alpha$ | $q_{\text{MDS}}/\alpha$ | $q_{\text{ANY}}/\alpha$ | $|\mathbb{F}|/\alpha$ |
|-----------|----------|--------------------------|--------------------------|----------------------|
| $[n,n-2,n-1]$ | $2^{n-2}$ | $n-1$ | $n-2$ | $O(n)$ |
| $[n,n-3,n-1]$ | $2^{n-3}$ | $n-3$ | $n-3$ | $O(n^2)$ |
| $[n,n-3,n-2]$ | $2^{n(n-3)}$ | $3^{n-2}$ | $O(n^2)$ |
| $[n,n-4,n-1]$ | $3^{n-4}$ | $4^{n-4}$ | $n-4$ | $O(n^2)$ |
| $[n,n-4,n-2]$ | $3^{n(n-4)}$ | $4^{n-2}$ | $4(n-2)(n-4)$ | $O(n^2)$ |
| $[n,n-4,n-3]$ | $2^{2(n-4)}$ | $3^{n-3}$ | $(n-4)(2^{n-3}n+32)$ | $O(n^3)$ |
| $[n,n-3]$ all-$d$ | $6^{n(n-3)}$ | $3^{n-3}$ | $(n-3)(n-3)^{n-3}$ | $O(n^3)$ |

**V. Construction of $[n,k]$ MSR Codes**

This section is devoted to generalizing the construction of $[n,k,d]$ MSR codes from previous sections to $[n,k]$ MSR codes, where a fixed code construction achieves the optimal repair bandwidth for any given helper nodes ($k \leq d \leq n - 1$).

**Definition.** Let $\Omega$ be an $r \times \theta$ matrix, where its elements are chosen from $\mathbb{Z}_\tau$ and $\tau$ is the least common multiple $\text{lcm}{1,2,\ldots,r}$. We call $\Omega$ an $(r,\theta)$-Product Matrix if and only if for any subset of rows in $\Omega$ such as $\{i_1,\ldots,i_p\} \subset [r]$ (for $p \leq r$), there exists a column $c \in [\theta]$ which intersects with these rows in the set of all the elements modulo $\mathbb{Z}_p$.

We previously mentioned that the matrix $\Omega$ in Fig. 3 satisfies the property that for any selection of two rows, there exists a column such that it intersects with those rows in $\{0,1\}$, the set of all elements modulo $\mathbb{Z}_2$. Before proceeding with an example construction of a Product Matrix, recall the definition of $\omega_\rho(i)$, which was the key element in Construction 2. Consider an $r \times \binom{r}{\rho}$ matrix $\Omega$ where

$$\Omega_{i,a} = \omega_\rho(i) \quad \text{for} \quad i \in [r] \text{ and } a \in \binom{r}{\rho}.$$

It is easy to verify that for any selection of $\rho$ rows from $\Omega$, one can find a column whose intersection with these rows form all elements in $\{0,1,\ldots,\rho-1\}$.

**Example 3.** A simple way to construct a Product Matrix is to concatenate the $\Omega$ matrices from Construction 2 for different values of $\rho \leq r$ and arrive at an $r \times \theta$ matrix where $\theta = \binom{r}{\rho} + \binom{r-1}{\rho-1} + \cdots + \binom{2}{2}$. The $(3,4)$-Product Matrix defined as

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

where the elements are chosen from $\mathbb{Z}_6$, serves as such an example.

**Construction 3.** Let $\Omega$ be an $(r,\theta)$-Product Matrix. Define $\alpha = \tau^\theta$, where $\tau = \text{lcm}{1,2,\ldots,r}$, and label the
a elements \(0 : \alpha - 1\) by \(\tau\)-ary vectors in \(\mathbb{Z}_\tau\). Define permutation \(f_j^\ell\) on \([0 : \alpha - 1]\) as follows:

\[
f_j^\ell : \mathbb{Z}_\tau^{k\theta} \rightarrow \mathbb{Z}_\tau^{\theta}
\]

\[
v \mapsto v + \ell v_j^n
\]

for \(j \in [k\theta]\) and \(\ell \in [0 : \tau - 1]\), where \(\{e_1, \ldots, e_{k\theta}\}\) is the standard vector basis of \(\mathbb{Z}_\tau^{k\theta}\). Let \(P_j^\ell\) be the \(\alpha \times \alpha\) matrix corresponding to the permutation \(f_j^\ell\).

1) MSR Code: The generator matrix of the \([n, k]\) code is given by (4), where

\[
A_{i,j} = \lambda_{i,j} \prod_{a \in [\theta]} P_{\rho a}^{\theta a (j-1) \rho}
\]

for \(j \in [k\theta]\), where \(\lambda_{i,j} \in \mathbb{F}_q\) are defined according to the discussion in Section II-B and Section IV to ensure the MDS property and to allow arbitrary selections of helper nodes.

2) Repair Strategy: Assume that the helper set \(D\) consists of the remaining \(k - 1\) systematic nodes (an assumption that can be dropped by a clever selection of \(\lambda_{i,j} \in \mathbb{F}_q\) according to Section [V] and a subset of parity nodes denoted by \(R\)). Furthermore, assume \(a \in [\theta]\) corresponds to a column in \(\Omega\) whose intersection with the rows associated with these parities forms all elements mod \(\mathbb{Z}_\rho\), where \(\rho = |R|\). Define \(Y_{i,a} \subset [0 : \alpha - 1]\) as \(\{x \in [0, \alpha - 1] : x \cdot e_{a+(j-1)\rho} = 0\ (mod\ \rho)\}\). If systematic node \(j\) fails, it is repaired by downloading the elements corresponding to \(Y_{i,a}\) from helper nodes in \(D\), i.e., by accessing \(x_i(v)\), where \(i \in D\), and \(v \in Y_{i,a}\).

**Lemma 8.** The repair strategy in Construction 3 is optimal with respect to repair bandwidth.

**Proof:** The proof in general follows steps similar to those for that of Theorem 3 and is omitted to avoid repetition. We refer the reader to the previous sections for notation, definitions and the overall flow of the proof. However, it is helpful to mention that the matrix \(\Omega\) is designed in a way that for any repair scenario \(a\), we can always find a column in \(\Omega\), denoted by \(\Omega_a\), such that its intersection with \(\rho\) rows associated with parity nodes in scenario \(a\) is given by \(\{0, 1, \ldots, \rho - 1\}\).

Now assume that systematic node \(i\) fails and we are to perform an optimal repair given parity nodes in \(\mathcal{R}_a = \{x_{k+d_1^\rho}, x_{k+d_2^\rho}, \ldots, x_{k+d_r^\rho}\}\). It is now clear that if we select our subspaces as \(S_i^j = S_j^\rho = Y_{i,a}\) = \(\{x | x \cdot e_{a+(i-1)\rho} = 0\ (mod\ \rho)\}\), then \(\dim(S_j^\rho) = \frac{\alpha}{\rho}\), and therefore,

\[
\begin{pmatrix}
S_i^j Q_{d_1^\rho, i}\theta \\
S_i^j Q_{d_2^\rho, i}\theta \\
\vdots \\
S_i^j Q_{d_r^\rho, i}\theta
\end{pmatrix}
= \rho \times \text{rank}(Y_{i,a}) = \frac{\alpha}{\rho} = \alpha.
\]

Notice that arriving at a construction which guarantees the optimal repair bandwidth for any selection of helper nodes comes at the cost of a huge increase in the sub-packetization parameter \(\alpha\).

**Remark:** This increase in sub-packetization can be viewed also as a trade-off between \(\alpha\) and the flexibility of choosing the helper set size. For instance, assume that only a few choices of \(d\) such as \(\{d_1, d_2, \ldots, d_\delta\}\) \(\subset [r]\) are desired. A naive construction of the product matrix as in Example 3 would result in an \(r \times \theta\) matrix, where

\[
\theta = \left(\frac{r}{d_1 - k + 1}\right) + \left(\frac{r}{d_2 - k + 1}\right) + \cdots + \left(\frac{r}{d_\delta - k + 1}\right),
\]

and its elements are chosen from \(\mathbb{Z}_\tau\), with

\[
\tau = \text{lcm}(d_1 - k + 1, d_2 - k + 1, \ldots, d_\delta - k + 1).
\]

The sub-packetization of this MSR code — call it an \([n, k, \{d_1, d_2, \ldots, d_\delta\}]\) MSR code — from Construction 3 is then given by \(\alpha = \tau^{k\theta}\), which establishes a trade-off between \(\alpha\) and the flexibility \(\{d_1, d_2, \ldots, d_\delta\} \subset [r]\) of the helper set size.

For practical purposes one may seek to reduce \(\alpha\). One way to do so is to minimize the number of columns in a Product Matrix for a fixed number of rows. For example, it is easy to verify that the matrix \(\Omega\) defined as

\[
\Omega = \begin{bmatrix}
0 & 0 \\
1 & 0 \\
2 & 1
\end{bmatrix},
\]

is also a Product Matrix with \(r = 3\), and would be preferred over the \((3, 4)\)-Product Matrix introduced in Example 3. Although the trivial construction of Product Matrices guarantees \(\theta = O(2^r)\), we believe that the optimal value for \(\theta\) is much smaller. We finish this section by stating this as an interesting combinatorial problem, which will be addressed in future work.

**Open Problem.** Given a positive integer \(r \in \mathbb{Z}\), what is the smallest value for \(\theta\) such that there exists an \((r, \theta)\)-Product Matrix, i.e., an \(r \times \theta\) matrix \(\Omega\) with elements in \(\mathbb{Z}_\tau = \text{lcm}(\{1, 2, \ldots, r\})\), where for any selection of \(\rho\) \(\{1 \leq \rho \leq r\}\) rows from \(\Omega\), one can always find a column in \(\Omega\) which intersects with these \(\rho\) rows in all the elements in \(\mathbb{Z}_\rho\)?

**VI. Conclusion**

In this paper we first presented a new construction for systematic–repair \([n, k, d]\) MSR codes for all possible values of parameters \(n, k,\) and \(d\). We discussed how a
proper selection of coefficients in $F_q$ guarantees the MDS property of the code, which can also be utilized to drop the restriction of choosing all the remaining systematic nodes when selecting the helper nodes.

Next, we introduced a more generalized construction, where a single $[n,k]$ code simultaneously satisfies the optimal repair for all $d \in \{k+1, \ldots, n-1\}$. Both generalizations come at the cost of increasing $\alpha$. A lower bound on $\alpha$ is proved in [29] when $d = n-1$. Whether similar bounds exist for general $\{n,k,d\}$ or not is left for future work.

References


Appendix A

Interference Alignment

Proof of Lemma 1. We prove the result for the failure of systematic node $i=1$. The argument generalizes for the failure of other systematic nodes. Let us assume that node $x_1$ fails, and let each of the remaining $d=n-1$ nodes send $\beta=\alpha/r$ symbols to recover $x_1$. In other words, node $x_i$ (where $j \in |n|$, $j \neq i$) sends $S_j^i x_j$ for some $\alpha/r \times \alpha$ matrix $S_j$. We therefore need to recover $x_1$ from the following functions of $x_i, j \in [k]$:

\[
\begin{pmatrix}
S_1^1 \\
S_2^1 \\
S_3^1 \\
S_{k+1}^1 A_{11} \\
S_{k+1}^1 A_{12} \\
\ldots \\
S_{k+1}^1 A_{1k} \\
S_{k+2}^1 A_{21} \\
S_{k+2}^1 A_{22} \\
\ldots \\
S_{k+2}^1 A_{2k} \\
\vdots \\
\vdots \\
S_{k+r}^1 A_{r1} \\
S_{k+r}^1 A_{r2} \\
\ldots \\
S_{k+r}^1 A_{rk}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
x_k
\end{pmatrix}
\]
\[ Necessity: \text{ Suppose the systematic vectors } x_2 \text{ through } x_k \text{ be the zero vectors. Then, (18) simplifies to: } \\
\begin{pmatrix}
S_{k+1}^1 A_{1,1} & \cdots & \cdots & S_{k+1}^1 A_{1,k} \\
S_{k+2}^1 A_{2,1} & \cdots & \cdots & S_{k+2}^1 A_{2,k} \\
\vdots & \ddots & \ddots & \vdots \\
S_{k+r}^1 A_{r,1} & \cdots & \cdots & S_{k+r}^1 A_{r,k}
\end{pmatrix} x_1, \\
\text{where the matrix is an } \alpha \times \alpha \text{ square matrix. Since } x_1 \text{ is recoverable, it is necessary that the matrix be non-singular, thus proving the signal recovery conditions (5) and (7). Note that this also implies that all encoding matrices } A_{i,j}, i \in [r], j \in [k], \text{ are non-singular. } s \\
\text{Suppose now, without loss of generality, that the interference alignment condition (6) is not satisfied for } j = 2. \text{ Again, without loss of generality, let } S_1^1 \not\subset S_{k+1}^1 A_{1,2}. \text{ This implies that} \\
\text{rank} \left( \begin{pmatrix}
S_1^1 \\
S_{k+1}^1 A_{1,2}
\end{pmatrix} \right) = \alpha + \epsilon, \tag{19}
\] \\
\text{for some } \epsilon > 0. \text{ Since } x_1 \text{ is recoverable, from (18), we have access to the following information at the replacement node:} \\
\begin{pmatrix}
S_1^1 \\
S_{k+1}^1 A_{1,2}
\end{pmatrix} \begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_k
\end{pmatrix} = \begin{pmatrix}
I_r \\
S_2^1 \\
S_3^1 \\
\vdots \\
S_k^1
\end{pmatrix} \begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_k
\end{pmatrix}. \tag{20}
\] \\
\text{From (19), the rank of the matrix in (20) is at least} \\
\frac{\alpha + \alpha}{r} + \epsilon + (k-2)\frac{\alpha}{r} > (n-1)\frac{\alpha}{r}, \\
\text{the total number of symbols available at the replacement node. In other words, we are able to recover more number of linearly independent symbols that are functions of the systematic data vectors } x_1 \text{ through } x_p, \text{ than the number of repair symbols available at the replacement node — a contradiction! Thus, conditions (5) and (6) must be true.} \\
\text{Sufficiency: Suppose that we have the required repair matrices } S_j^1 \text{ that satisfy the signal recovery and interference alignment conditions (5) and (6). Using (5), we can eliminate the contribution of systematic vectors } x_2 \text{ through } x_k \text{ in the information transmitted by the parity nodes (that is, the last } r \text{ rows in (18)). For instance, } S_1^1 \not\subset S_{k+1}^1 A_{1,2} \text{ implies that } S_{k+1}^1 A_{1,2} = BS_2^1, \text{ for some } \alpha/r \times \alpha/r \text{ matrix } B, \text{ and therefore the contribution of } S_{k+1}^1 A_{1,2} x_2 \text{ can be removed from the repair information transmitted by the parity node } k+1 \text{ using the repair information } S_2^1 x_2 \text{ (or equivalently, } BS_2^1 x_2) \text{ transmitted by systematic node 2. Using (5), it is then easy to recover } x_1. \]