Long MDS Codes for Optimal Repair Bandwidth

Zhiying Wang†, Itzhak Tamo‡, and Jehoshua Bruck∗
†Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125, USA
‡Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 84105, Israel
{zhiying, tamo, bruck}@caltech.edu

Abstract—This paper is eligible for the student paper award. MDS codes are erasure-correcting codes that can correct the maximum number of erasures given the number of redundancy or parity symbols. If an MDS code has r parities and no more than r erasures occur, then by transmitting all the remaining data in the code one can recover the original information. However, how much information is needed in order to recover the code if only one symbol is erased? The amount of information needed is called the repair bandwidth. It was shown that only a fraction of 1/r of the information is needed to recover a single erasure and explicit code constructions were given in previous works. If we view each symbol in the code as a vector or a column, then the code forms a 2D array and such codes are especially widely used in storage systems. In this paper, we ask the following question: given the length of the column l, can we construct MDS array codes with optimal repair bandwidth of 1/r, whose code length is as long as possible? In this paper, we give code constructions such that the code length is (r + 1) log l.

I. INTRODUCTION

MDS (maximum distance separable) codes are optimal error-correcting codes in the sense that they have the largest minimum distance given the number of parity symbols. If each symbol is a vector or a column, we call such a code an MDS array code (e.g. [1], [6], [20], [21]). In (distributed) storage systems, each column is usually stored in a different disk, and MDS array codes are widely used to protect data against erasures due to their error correction ability and low computational complexity. In this paper, we call each symbol a column or a node, and the column length is denoted by l.

If an MDS code has r parities, then it can correct up to r erasures of entire columns. In this paper, we not only would like to recover any e erasures, e ≤ r, but also care about the efficiency in recovery: what is the fraction of the remaining data transmitted in order to correct e erasures? We call this fraction repair bandwidth (fraction). For example, if e = r erasures happen, it is obvious that we have to transmit all of the remaining information, therefore, the fraction is 1. If only e < r erasures happen, then can we transmit less information? The answer is yes, and it was shown in [7] (which also formulated the repair problem) that this fraction is actually upper bounded by e/r. If this bound is achieved for some code, we say it has optimal repair. Since the repair of information is much more crucial than redundancy, and we study mainly high-rate codes, we will focus on the optimal repair of information or systematic nodes. Moreover, since single erasure is the most common scenario in practice, we assume e = 1. For example, in Figure 1, we show an MDS code with 4 systematic nodes, r = 2 parity nodes, and column length l = 2. One can check that this code can correct any two erasures, therefore it is an MDS code. In order to repair any systematic node, only 1/r = 1/2 fraction of the remaining information is transmitted. Thus this code has optimal repair.

In [10]–[12], [19] codes achieving the repair bandwidth upper bound were studied where the number of systematic nodes is less than the number of parity nodes (low code rate). For arbitrary code rate, [5], [13] proved that the upper bound e/r is asymptotically achievable when the column length l goes to infinity. And [2]–[4], [8], [9], [14], [15], [17] studied codes with more systematic nodes than parity nodes (high code rate) and finite l, and achieved the upper bound of the repair bandwidth. However, if we are interested in the code length, i.e., the number of systematic nodes given l, these high-rate constructions are relatively short. For example, suppose that we have 2 parity nodes, then the number of systematic nodes is only log l in all of the constructions, except for [4] it is 2 log l. In [16] it is shown that an upper bound for the code length is k ≤ 1 + log l, but the tightness of this bound is not known. It is obvious that there is a big gap between this upper bound and the constructed codes.

The main contribution of this paper is to construct codes with 2 parity nodes and 3 log l systematic nodes. Moreover, we will give a general construction of high-rate codes with (r + 1) log l systematic nodes for arbitrary number of parities r. It turns out that this construction is a combination of the code in [4] and also [2], [9], [14].

The rest of the paper is organized as follows: in Section II we will formally introduce the repair bandwidth and the code length problem. In Section III codes with 2 parity nodes are constructed, and we show that the code length is 3 log l. Generalized code constructions for arbitrary number of parities are given in Section IV and finally we conclude in Section V.

II. PROBLEM SETTINGS

An (n, k, l) MDS array code is an (n − k)-erasure-correcting code such that each symbol is a column of length l. The number of systematic symbols is k and the number of parity symbols is r = n − k. We call each symbol a column or a node, and k the code length. We assume that the code is systematic, hence the first k nodes of the code are information or systematic nodes, and the last r nodes are parity or redundancy nodes.
Suppose the columns of the code are $C_1, C_2, \ldots, C_n$, each being a vector in $F^r$, for some finite field $F$. We assume that for parity node $k+i$, information node $j$, the coding matrix is $A_{ij}$ of size $1 \times l$, $i \in [r], j \in [k]$. And the parity columns are computed as

$$C_{k+i} = \sum_{j=1}^{k} A_{ij} C_j,$$

for all $i \in [r]$. For example, in Figure 1, the coding matrices are $A_{1,j} = 1$ for all $j \in [k]$ and $A_{2,j}, j = 1, 2, 3, 4$ are

$$\begin{pmatrix} 1 & 5 \\ 0 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 4 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 5 \end{pmatrix}.$$ 

In our constructions, we require that $A_{1,j} = 1$ for all $j \in [k]$. Hence the first parity is the row sum of the information array. Even though this assumption is not necessarily true for an arbitrary linear MDS array code, it can be shown that any linear code can be equivalently transformed into one with such coding matrices [16].

Suppose a code has optimal repair for any systematic node $i, i \in [k]$, meaning only a fraction of $1/r$ data is transmitted in order to repair it. When a systematic node $i$ is erased, we are going to use $l/r \times l$ matrices $S_{ij}$, $j \neq i, j \in [n]$, to repair the node: From a surviving node $j$, we are going to compute and transmit $S_{ij} C_j$, which is only $1/r$ of the information in this node.

**Notations:** In order to simplify the notations, we write $S_{ij}$ and $S_{i,j+k} A_{ij}$ both as matrices of size $l/r \times l$ and the subspaces of their row spans.

Optimal repair of a systematic node $i$ is equivalent to the following **subspace property**: There exist matrices $S_{ij}$, $j \neq i, j \in [n]$, all with size $l/r \times l$, such that for all $j \neq i, j \in [k], t \in [r]$,

$$S_{ij} = S_{i,j+k} A_{ij},$$

where the equality is defined on the row spans instead of the matrices. And

$$\sum_{t=1}^{r} S_{i,j+k} A_{ij} = \mathbb{F}^l.$$  \hspace{1cm} (2)

Here the sum of two subspaces $A, B$ of $\mathbb{F}^l$ is defined as $A + B = \{a + b : a \in A, b \in B\}$. Obviously, the dimension of each subspace $S_{i,j+k} A_{ij}$ is no more than $l/r$, and the sum of $r$ such subspaces has dimension no more than $l$. This means these subspaces intersect only on the zero vector. Therefore, the sum is actually the direct sum of vector spaces. Moreover, we know that each $S_{i,j+k}$ has full rank $l/r$.

We claim that (1) (2) are necessary and sufficient conditions for optimal repair. The sketch of the proof is as follows: suppose the code has optimal repair bandwidth, then we need to transmit $l/r$ elements from each surviving column. Suppose we transmit $S_{ij} C_j$ from a systematic node $j \neq i, j \in [k]$, and $S_{i,j+k} C_{j+k} = \sum_{t=1}^{r} S_{i,j+k} A_{ij} C_j$ from a parity node $j+k \in [k+1, k+r]$. Our goal is to recover $C_j$ and cancel out all $C_j$, $j \neq i, j \in [k]$. In order to cancel out $C_j$, (1) must be satisfied. In order to solve $C_j$, all equations related to $C_i$ must have full rank $l$, so (2) is satisfied. One the other hand, if (1) (2) are satisfied, one can transmit $S_{ij} C_j$ from each node $j, j \neq i, j \in [n]$ and optimally repair the node $i$. Similar interference alignment technique was first introduced in [5], [13] for the repair problem.

It is shown in [16] that we can further simplify our repair strategy of node $i$ and assume $S_{ij} = S_i$, for all $j \neq i, j \in [n]$ by equivalent transformation of the coding matrices (probably with an exception of the strategy of one node). Then the **subspace property** becomes for any $j \neq i, j \in [k], t \in [r]$,

$$S_i = S_i A_{ij}.$$  \hspace{1cm} (3)

Again the equality means equality of row spans. And the sum of subspaces satisfies

$$\sum_{i=1}^{r} S_i A_{ij} = \mathbb{F}^l.$$  \hspace{1cm} (4)

Notice that if (3) is satisfied, we can say that $S_i$ is an invariant subspace of $A_{ij}$ (multiplied on the left) for all parity nodes $k+t$ and all information nodes $j \neq i$. If $A_{ij}$ is diagonalizable and has $l$ linearly independent left eigenvectors, an invariant subspace has a set of basis which are all eigenvectors of $A_{ij}$. As a result, our goal is to find matrices $A_{ij}$ and their invariant subspaces. And by using sufficiently large finite field and varying the eigenvalues of the coding matrices, we are able to ensure that the codes are MDS. Therefore, we will focus on finding eigenvectors of the coding matrices and not the eigenvalues.

For example, in Figure 1, the matrices $S_i, i = 1, 2, 3$ are

$$(1, 0), (0, 1), (1, 1).$$

One can check that the subspace property (3)(4) is satisfied for $i \in [3]$. For instance, since $S_3 = (1, 1)$ is an eigenvector for $A_{1,j}, t = 1, 2, j = 1, 2, 4$, we have $S_2 = S_3 A_{1,j}$. And it is easy to check that $S_3 \oplus S_3 A_{2,3} = \text{span}(1, 1) \oplus \text{span}(2, 4) = \mathbb{F}^2$. For the node $N_4$, the matrices $S_{4,j}$’s are not equal. In fact $S_4,j = (1, 4)$ for $j = 1, 2, 3, 5$ and $S_{4,6} = (1, 2)$.

### III. Code Constructions with 2 Parities

In this section, we are going to construct codes with column length $l = 2^m$, $k = 3m$ systematic nodes, and $r = 2$ parity
where $A_{1,i} = I$ and $A_{2,i} = A_i$ correspond to parity 1 and 2 respectively.

Now we only need to find coding matrices $A_i$’s, and subspaces $S_i$’s. Moreover, we only care about eigenvectors of $A_i$, not its eigenvalues. In the following construction, for any $i \in [k]$, $A_i$ has two different eigenvalues $\lambda_{i,0}, \lambda_{i,1}$, each corresponding to $1/2 = 2^{m-1}$ eigenvectors. Denote these eigenvectors as

$$V_{i,0} = \begin{pmatrix} v_{i,0} \\ v_{i,1} \\ \vdots \\ v_{i,l/2} \end{pmatrix}$$

for eigenvalue $\lambda_{i,0}$, and

$$V_{i,1} = \begin{pmatrix} v_{i,1/2+1} \\ v_{i,1/2+2} \\ \vdots \\ v_{i,1} \end{pmatrix}$$

for eigenvalue $\lambda_{i,1}$. Therefore, $A_j$ can be computed as

$$A_j = \left( V_{i,0} V_{i,1}^{-1} \right) \left( \begin{pmatrix} \lambda_{i,0} I_{l/2} \otimes I \\ \lambda_{i,1} I_{l/2} \otimes I \end{pmatrix} \right) \left( V_{i,0} V_{i,1}^{-1} \right)^{-1} \left( V_{i,0} V_{i,1}^{-1} \right).$$

By abuse of notations, we also use $V_{i,0}, V_{i,1}$ to represent the eigenspace corresponding to $\lambda_{i,0}, \lambda_{i,1}$, respectively. Namely, $V_{i,0} = \text{span}\{v_{i,0}, \ldots, v_{i,l/2}\}$ and $V_{i,1} = \text{span}\{v_{i,l/2+1}, \ldots, v_{i,1}\}$.

When a systematic node $i$ is erased, $i \in [k]$, we are going to use $S_i$ to rebuild it. The **subspace property** becomes

$$S_i = S_i A_j \quad \forall j \neq i, j \in [k],$$

$$S_i + S_j A_i = \mathbb{F}^m. \quad (5)$$

In the following construction, $e_a$, $a \in [0, l - 1]$, are some basis of $\mathbb{F}^l$, for example, one can think of them as the standard basis. The subscript $a$ is represented by its binary expansion, $a = (a_1, a_2, \ldots, a_m)$. For example, if $l = 16, m = 4$, $a = 5$, then $e_a = e_{(0,0,1,0)}$ and $a_1 = 0, a_2 = a_3 = 1, a_4 = 0$.

In order to construct the code, we first define 3 sets of vectors for $i \in [m]$:

$$P_{i,0} = \{ e_a : a_i = 0 \},$$

$$P_{i,1} = \{ e_a : a_i = 1 \},$$

$$Q_i = \{ e_a + e_b : a_i + b_i = 1, a_j = b_j, \forall j \neq i \}.$$

For example, if $m = 2$, $i = 1$, then $P_{i,0} = \{ e_{(0,0)}, e_{(0,1)} \} = \{ e_0, e_1 \}$, $P_{i,1} = \{ e_{(1,0)}, e_{(1,1)} \} = \{ e_2, e_3 \}$, and $Q_i = \{ e_{(0,0)} + e_{(1,0)}, e_{(0,1)} + e_{(1,1)} \} = \{ e_0 + e_2, e_1 + e_3 \}$. **Notation:** The subscript $i$ for sets $P_{i,u}, Q_i$ and $a_i$ (the $i$-th digit of vector $a$) is written modulo $m$. For example, if $i \in [tm + 1, (t + 1)m]$ for some integer $t$, then $P_{i,u} = P_{i-tm,u}$.

**Construction 1** The $(n = 3m + 2, k = 3m, l = 2m)$ code has coding matrices $A_i, i \in [k]$, each with two distinct eigenvalues, and eigenvectors $V_{i,0}, V_{i,1}$. When node $i$ is erased, we are going to use $S_i$ to rebuild. We construct the code as follows:

1. For $i \in [m]$, $V_{i,0} = \text{span}(Q_i)$, $V_{i,1} = \text{span}(P_{i,1})$. $S_i = \text{span}(P_{i,0})$.
2. For $i \in [m + 1, 2m]$, $V_{i,0} = \text{span}(P_{i,0})$, $V_{i,1} = \text{span}(Q_i)$, $S_i = \text{span}(P_{i,1})$.
3. For $i \in [2m + 1, 3m]$, $V_{i,0} = \text{span}(P_{i,0})$, $V_{i,1} = \text{span}(P_{i,1})$, $S_i = \text{span}(Q_i)$.

**Example 1** Deleting the node $N_4$, Figure 1 is a code using Construction 1 and $l = 2$. Another example of $l = 4$ is shown in Figure 2. One can check (5) holds. For instance, $S_1 = \text{span}\{e_0, e_1\} = \text{span}\{e_0 + e_1, e_1\}$ is an invariant subspace of $A_2$. So $S_1 = S_1 A_2$. If the two eigenvalues of $A_i$ are distinct, it is easy to show that $S_1 \oplus S_i A_i = \mathbb{F}^m, \forall i \in [6]$.

The above example shows that for $m = 1, 2$, the constructed code has optimal repair. It is true in general, as the following theorem suggests.

**Theorem 2** Construction 1 is a code with optimal repair bandwidth $1/2$ for rebuilding any systematic node.

**Proof:** By symmetry of the first two cases in the construction, we are only going to show that the rebuilding of node $i, i \in [m] \cup [2m + 1, 3m]$ is optimal. Namely, the subspace property (5)(6) is satisfied. Recall that $S_i A_j = S_j$ is equivalent to $S_j$ being an invariant subspace of $A_j$.

Case 1: $i \in [m]$.

- When $j \in [tm + 1, (t + 1)m], j - tm \neq i, t \in \{0,1\}$, define $B = \{ e_{a} : a_i = 1 - t, a_i = 0 \} \cup \{ e_{a} + e_{b} : a_i + b_i = 1, a_i = b_i = 0, a_j = b_j, \forall j \neq i, j \}$. Then it is easy to see that $S_i = \text{span}(P_{i,0}) = \text{span}(B)$. Moreover, each vector in set $B$ is an eigenvector of $A_j$, therefore $S_i$ is an invariant subspace of $A_j$.
- When $j - m = i, S_i = V_{i,0} = \text{span}(P_{i,0})$, so $S_i$ is an eigenspace of $A_j$.
- When $j \in [2m + 1, 3m]$, we can see that every vector in $P_{i,0}$ is a vector in $V_{i,0} = \text{span}(P_{i,0})$ or in $V_{i,1} = \text{span}(P_{i,1})$, hence it is an eigenvector of $A_j$.
- When $j = i$, consider a vector $e_a \in P_{i,0}$, then $a_i = 0$. And $e_a = (e_a + e_b) - e_b$ where $b_i = 1, b_j = a_j$ for all $j \neq i$. Here both $e_a + e_b$ and $e_b$ are eigenvectors of $A_i$.

$$e_a A_i = (e_a + e_b) A_i - e_b A_i = \lambda_{i,0} (e_a + e_b) - \lambda_{i,1} e_b = (\lambda_{i,0} - \lambda_{i,1}) e_b + \lambda_{i,0} e_a.$$  

Because $\lambda_{i,0} \neq \lambda_{i,1}$, we get $\text{span}\{e_a A_i, e_a\} = \text{span}(e_a, e_b)$. Hence $S_i A_j + S_j = \text{span}(e_a, e_b : a_i = 0, b_i = 1, a_j = b_j, \forall j \neq i) = \mathbb{F}^m$.

Case 2: $i \in [2m + 1, 3m]$.

- When $j = i - m$ or $j = i - 2m$, $S_i = \text{span}(Q_i)$ is an eigenspace of $A_j$.
- When $j \in [tm + 1, (t + 1)m]$, and $j \neq i - tm$ for $t \in \{0,1\}$, define $D = \{ e_a + e_b : a_j = b_j = 1 - t, a_i = b_i = 1 \} = \mathbb{F}^m$. Theorem 2 is satisfied.
1. a_2 = b_2, \forall z \neq i, j \} \cup \{ e_a + c_b + c_d + e_d : a_j = b_j = 0, c_j = d_j = 1, a_i + b_i = 1, c_i + d_i = 1, a_z = b_z = c_z = d_z, \forall z \neq i, j \}. We can see that S_i = \text{span}(Q_i) = \text{span}(D) and every vector in D is an eigenvector of A_i.

- When \( j = i \), consider any \( e_a + e_b \in Q_i \), where \( a_i = 1, b_i = 0, a_z = b_z, \forall z \neq i \). We have
  \[(e_a + e_b)A_i = \lambda_i e_a + \lambda_i e_b.\]

Because \( \lambda_i \neq \lambda_j \), we get \( \text{span}\{e_a + e_b : a_i = 1, b_i = 0, a_z = b_z, \forall z \neq i \} = \mathbb{F}^l.\)

It should be noted that if we shorten the code and keep only the first 2m systematic nodes in the code, then it is actually equivalent to the code in [4]. The repairing of the first 2m nodes does not require computation within each remaining node, since only standard bases are multiplied to the surviving columns (e.g. Figure 2). We call such repair optimal access.

It is shown in [16] that if a code has optimal access, then the code has no more than 2m nodes. On the other hand, the shortened code with the last m systematic nodes in the above construction is equivalent to that of [2], [9], [14]. Since the coding matrices A_i, i \in [2m + 1, 3m] are all diagonal, every information entry is included in only r + 1 entries in the code. We say such a code has optimal update. In [16] it is proven that an optimal-update code with diagonal coding matrices has no more than m nodes. Therefore, our code is a combination of the longest optimal-access code and the longest optimal-update code, which provides tradeoff among access, update, and the code length.

In addition, if we try to extend an optimal-access code \( C \) with length 2m to a code \( D \) with length k, so that \( C \) is a shortened code of \( D \), then the following theorem shows that \( k = 3m \) is largest code length. The proof is omitted and can be found in the long version of this paper [18]. Therefore, our construction is longest in the sense of extending \( C \).

**Theorem 3** Any extended code of an optimal-access code of length 2m will have no more than 3m systematic nodes.

**IV. CODES WITH ARBITRARY NUMBER OF PARITIES**

In this section, we will give constructions of codes with arbitrary number of parity nodes. Our code will have \( l = r^m \) rows, \( k = (r + 1)m \) systematic nodes, and r parity nodes, for any \( r \geq 2, m \geq 1 \).

Suppose \( A_{s,j} \) is the coding matrix for parity node \( k + s \) and information node \( i \). From Section II, we assume \( A_{1,j} = I \) for all \( i \). In our construction, we are going to add the following assumptions. Each \( A_{s,j} \) has \( r \) distinct eigenvalues, each corresponding to \( l/r = r^{m-1} \) linearly independent eigenvectors. Moreover, given an information node \( i \in [k] \), all matrices \( A_{s,j} \), \( s \in [2, r] \), share the same eigenspaces \( V_{i,0}, V_{i,1}, \ldots, V_{i,r−1} \). If these eigenspaces correspond to eigenvalues \( \lambda_{s,j}, \lambda_{s,j+1}, \ldots, \lambda_{s,j+r−1} \) for \( A_{s,j} \), then we assume they correspond to eigenvalues \( \lambda_{s,0}, \lambda_{s,1}, \ldots, \lambda_{s,r−1} \) for \( A_{s,j} \). By abuse of notations, \( V_{i,j} \) represents both the eigenspace and the \( 1/r \times 1 \) matrix containing \( 1/r \) independent eigenvectors. Under these assumptions, it is easy to see that if we write \( A_{s,j} \) as

\[
\begin{pmatrix}
V_{i,0} \\
\vdots \\
V_{i,r−1}
\end{pmatrix}^{-1}
\begin{pmatrix}
\lambda_{s,i,0} I \\
\vdots \\
\lambda_{s,i,r−1} I
\end{pmatrix}
\begin{pmatrix}
V_{i,0} \\
\vdots \\
V_{i,r−1}
\end{pmatrix},
\]

where the identity matrices are of size \( \frac{1}{r} \times \frac{1}{r} \), then \( A_{s,j} = A_{2s,j}^{s−1} \), for all \( s \in [r] \). Hence, we are going to write \( A_i = A_{2s,i} \), thus \( A_{s,j} = A_{s,i}^{s−1} \), and our construction will only focus on the matrix \( A_i \). As a result, the **subspace property** becomes

\[
S_i = S_i A_j, \quad (7)
\]

\[
S_i + S_i A_j + S_i A_j^2 + \cdots + S_i A_j^{r−1} = \mathbb{F}^l. \quad (8)
\]

Note that such choice of eigenvalues is not the unique way to construct the matrices, but it guarantees that the code has optimal repair bandwidth. Also, when the finite field size is large enough, we can find appropriate values of \( \lambda_{i,j} \)’s such that the code is MDS. At last, since each \( V_{i,j} \) has dimension \( l/r \) and corresponds to \( 1/r \) independent eigenvectors, we know that any vector in the subspace \( V_{i,j} \) is an eigenvector of \( A_i \).

Let \( e_0, e_1, \ldots, e_{m−1} \) be the standard basis of \( \mathbb{F}^l \). And we are going to use the \( r \)-ary expansion to represent the index of a base. An index \( a \in [0, r^m − 1] \) is written as \( a = (a_1, a_2, \ldots, a_m) \), where \( a_i \) is its \( i \)-th digit. For example, when \( r = 3, m = 4 \), we have \( e_5 = e_{(0,0,1,2)} \). Define for
and rebuilding subspaces are for digit \( i \) optimal repair bandwidth by-Theorem 2, and can be found in [18].

\[
\begin{align*}
    P_{i,u} &= \{ e_u : a_i = u \}, \\
    Q_i &= \{ \sum_{a_i=0}^{r-1} e_a : j \in [0,r-1], j \neq i \}.
\end{align*}
\]

So \( P_{i,u} \) is the set of bases whose index is \( u \) in the \( i \)-th digit. The sum in \( Q_i \) is over all \( e \) such that the \( j \)-th digit of \( a \) is some fixed value, \( \forall j \neq i \), and the \( i \)-th digit varies in \([0,r-1]\). In other words, a vector in \( Q_i \) is the summation of the corresponding bases in \( P_{i,u} \), \( \forall u \). For example, when \( r = 3, m = 2 \), \( P_{1,0} = \{ e_{(0,0)}, e_{(0,1)}, e_{(0,2)} \} = \{ e_0, e_2 \} \), \( P_{1,1} = \{ e_3, e_4, e_5 \} \), \( P_{2,1} = \{ e_6, e_7, e_8 \} \), and \( Q_1 = \{ e_0 + e_3 + e_4 + e_7, e_2 + e_5 + e_8 \} \).

In the following, all of the subscript \( i \) for sets \( P_{i,u}, Q_i \) and for digit \( a_i \) are computed modulo \( m \). For example, if \( i \in [tm+1, (t+1)m] \) for some integer \( t \), then \( Q_i = Q_{i-tm} \).

**Construction 2** The \((n = (r + 1)m + r, k = (r + 1)m, l = r^m)\) code is constructed as follows. For information node \( i \in [tm+1, (t+1)m], t \in [0,r-1] \), the \( u \)-th eigenspace \((u \in [0,r-1])\) of coding matrix \( A_i \) and the rebuilding subspace \( S_i \) are defined as

\[
\begin{align*}
    V_{i,u} &= \text{span}(P_{i,u}), \text{if } u \neq t, \\
    V_{i,t} &= \text{span}(Q_i), \\
    S_i &= \text{span}(P_{i,t}).
\end{align*}
\]

For information node \( i \in [rm+1, (r+1)m] \), the eigenspaces and rebuilding subspaces are

\[
\begin{align*}
    V_{i,u} &= \text{span}(P_{i,u}), \forall u \in [0,r-1] \\
    S_i &= \text{span}(Q_i).
\end{align*}
\]

The following theorem shows that the code indeed has optimal repair bandwidth \( 1/r \). The proof is similar to that of Theorem 2, and can be found in [18].

**Theorem 4** Construction 2 has optimal repair bandwidth \( 1/r \) when rebuilding one systematic node.

Again, this construction can be shortened to an optimal-access code of length \( rm \) [4] and an optimal-update code of length \( m \) [2, 9, 14].

V. Conclusions

In this paper, we presented a family of codes with parameters \((n = (r + 1)m + r, k = (r + 1)m, l = r^m)\) and they are so far the longest high-rate MDS code with optimal repair. The codes were constructed using eigenspaces of the coding matrices, such that they satisfy the subspace property. This property gives more insights on the structure of the codes, and simplifies the proof of optimal repair.

If we require that the code rate approaches 1, i.e., \( r \) being a constant and \( m \) goes to infinity, then the column length \( l \) is exponential in the code length \( k \). However, if we require the code rate to be roughly a constant fraction, i.e., \( m \) being a constant and \( r \) goes to infinity, then \( l \) is polynomial in \( k \).

Therefore, depending on the application, we can see a tradeoff between the code rate and the code length.

It is still an open problem what is the longest optimal-repair code one can build given the column length \( l \). Also, the sizes of the finite field used for the codes are not specified in the paper. Unlike the constructions in this paper, the field size may be reduced when we assume that the coding matrices do not have eigenvalues or eigenvectors (are not diagonalizable). These are our future work directions.

**References**


