Explicit Minimum Storage Regenerating Codes

Zhiying Wang, Itzhak Tamo, and Jehoshua Bruck

***Abstract*—In distributed storage, a file is stored in a set of nodes and protected by erasure-correcting codes. Regenerating code is a type of code with two properties: first, it can reconstruct the entire file in the presence of any r node erasures for some specified integer r; second, it can efficiently repair an erased node from any subset of remaining nodes with a given size. In the repair process, the amount of information transmitted from each node normalized by the storage size per node is termed *repair bandwidth (fraction)*.**

When the storage size per node is minimized, the repair bandwidth is lower bounded by $1/r$, where r is the number of parity nodes. A code attaining this lower bound is said to have optimal repair. We consider codes with minimum storage size per node and optimal repair, called MSR (minimum storage regenerating) codes. In particular, if an MSR code has r parities and any r erasures occur, then by transmitting all the information from the remaining nodes, the original file can be reconstructed. On the other hand, if only one erasure occurs, only a fraction of $1/r$ of the information in each remaining node needs to be transmitted.

If we view each node as a vector or a column over some field, then the code forms a 2D array. Given the length of the column l, the number of parities r, we construct explicitly high-rate MSR codes. The number of systematic nodes of our construction is $(r+1)\log_r l$, which is longer than previously known results. Besides, we construct MSR codes with other desirable properties: first codes with low complexity when the information is updated, and second, codes with low access or storage node I/O cost during repair.

I. Introduction

The family of regenerating codes defined by the parameters $n, k, d, \alpha, \beta, \mathcal{M}$, was proposed in [8] for the following distributed storage model. Consider a file of size $\mathcal{M}$ bits to be encoded into a length n codeword, where each symbol of the encoding is stored on a distinct (storage) node of size α bits. Consider codes satisfying the following two properties: (i) Reconstruction: the entire file can be decoded from the information stored on any k nodes; (ii) Repair: in case of a node erasure (failure) the information can be *functionally* repaired from any $d \leq n-1$ helper nodes, each transmitting β bits of information. Functional repair means that the newly repaired node does not need to be identical to the erased node, but only to satisfy the reconstruction condition.

It was shown [8] that by the above two properties, the parameters satisfy

$$\sum_{i=0}^{k-1} \min\{(d-i)\beta, \alpha\} \geq \mathcal{M}. \tag{1}$$

For fixed $n, k, d, \alpha, \mathcal{M}$, codes that can repair any node erausre with β bits of transmission satisfying (1) with equality are called optimal repair codes. Codes with reconstruction property and optimal repair property are called regenerating codes. We study in the paper Minimum Storage Regenerating (MSR) codes, which have the minimal possible storage size, i.e., $\alpha = \frac{\mathcal{M}}{k}$. Furthermore, we consider the case with $d = n-1$ helper nodes, and assume that the codes are systematic, i.e., the first k nodes store the information itself, and the last $r = n-k$ are the parity nodes. The code is assumed to have $k > r$, namely, to be high rate.

Typically the information is viewed as an array of n columns (also referred to as nodes or symbols), where each column stores a vector of information over some finite field and it corresponds to one of the n codeword symbols. The length of each vector is denoted by l and its elements are referred to as sub-symbols. These type of codes are called array codes (e.g. [3], [7], [12], [23], [24]), and they are widely used in data storage to protect information against erasures due to their error correction capability and low computational complexity.

As typically required from storage systems, we restrict our focus to codes that perform *exact* repair, i.e., the erased information is exactly recovered during the repair process. Note that the lower bound (1) still holds even if we consider exact repair, since it is a special case of functional repair.

The efficiency of the repair is measured by the *repair bandwidth*, β/α, which is the amount of transmitted information from each helper node, normalized by the amount of information it stores. For MSR codes with $d = n-1$, constraint (1) becomes $\beta \geq \mathcal{M}/k(n-k)$, and thus the repair bandwidth for a single erasure satisfies

$$\frac{\beta}{\alpha} \geq \frac{1}{n-k} = \frac{1}{r}.$$

Since systematic nodes account for the majority of the nodes, and repairing them is much more crucial than repairing the parity nodes, we focus on the optimal repair of only the systematic part. We note here that from the derivation in [8], (1) still holds even if we restrict to optimal repair of only the systematic nodes[1]. Finally, in this paper an MSR code is referred to a code that can optimally repair any of its systematic nodes, i.e., the amount of transmitted information is equal to $1/r$ amount of stored information for each of the $n-1$ helper nodes.

Z. Wang is with Center for Pervasive Communications and Computing, Univeristy of California, Irvine, Irvine, USA (email: zhiying@uci.edu).

I. Tamo is with Department of Electrical Engineering - Systems, Tel Aviv University, Tel Aviv, Israel (email: tamo@post.tau.ac.il).

J. Bruck is with Department of Electrical Engineering, California Institute of Technology, Pasadena, USA (email: bruck@caltech.edu).

The paper was partially presented in ISIT, 2012 [20].

[1] In fact, we can obtain (1) from the information flow graph as in [8] considering the following sequence of repairs: repair Node 1 from Nodes $n+1-d, \ldots, n$, index the new node by $n+1$; repair Node 2 from Nodes $n+2-d, \ldots, n+1$, index the new node by $n+2$; and so on; repair Node k from Nodes $n+k-d, \ldots, n+k-1$, index the new node by $n+k$. If Nodes $1, \ldots, k$ are systematic, and every repaired node is exactly the same as the erased node, then (1) is a bound on codes that repair any systematic node.

For example, Figure 1 shows an MSR code with $k = 4$ systematic nodes, $r = 2$ parity nodes, and column length $l = 2$ over the finite field $\mathbb{F}_4$. One can check that this code decode the entire information from 4 nodes, therefore it satisfies the reconstruction property. The code can optimally repair any systematic node by transmitting from each helper node a fraction of $1/r = 1/2$ of the information it stores. We conclude that this is an $(n = 6, k = 4, l = 2)$ MSR code.

Following [8], constructions of MSR codes were studied by a series of work. In [13], [15], [16], [21], [22] MSR codes with more parity nodes than systematic nodes, i.e., low rate codes were studied. For arbitrary code rate, [6] proved that (1) is asymptotically tight when the column length l tends to infinity. Finally the question of explicitly constructing high rate MSR codes for a fixed l was resolved in [4], [5], [11], [17], [19], and recently studied in [1], [14].

An interesting question is what the relations among the parameters, n, k, l, of an MSR code are. In the low rate MSR code constructions [15], [16], the number of systematic nodes k scales linearly with l, more precisely $k = l + 1$; on the other hand, for a high rate MSR code k is much shorter compared to l. For example, the longest known MSR code with 2 parity nodes has only $k = 2\log_2 l$ systematic nodes [5]. Furthermore [9] provided an upper bound on k for such codes, where it showed that $k \leq 2(\log_2 l)(\log_2 l + 1) + 1$. Currently, it is unknown what the exact maximum value of k is.

Besides bandwidth which corresponds to the amount of transmission incurred during repair, we study two other metrics of a code. Firstly, the repair *access* is defined as the fraction of data read from the surviving nodes in order to repair an erasure. Access is an important metric because it affects the storage node I/O operations and hence the speed and complexity in repair. Since any transmitted sub-symbol needs to be first accessed and read prior to transmission, the repair access is always at least the repair bandwidth, and therefore an MSR code with r parity nodes satisfies

$$\frac{1}{r} \leq \text{repair bandwidth} \leq \text{repair access}.$$

For example, in Figure 1 the repair of node $N1$ reads and transmits only the first row, so the repair bandwidth and access are both $1/2$. However, the repair of node $N3$ requires reading both rows, so the access is 1.

Secondly, we define the *update* parameter of an information sub-symbol to be equal to the total number of sub-symbols in the code that are a function of the updated sub-symbol. This metric is important when we frequently update some portion of the information. In the code in Figure 1 there are 3 sub-symbols that are a function of the information sub-symbol a (the sub-symbol a itself and one sub-symbol in each parity node), therefore the update parameter of a equals 3. Similarly, the update parameter of sub-symbol w equals 4. By the reconstruction property it can be seen that any parity node of a systematic regenerating code is a function of all the information sub-symbols. Therefore the update parameter in a code with r parity nodes is at least $r + 1$ for any information sub-symbol. A code is said to be optimal update if for any information sub-symbol it achieves the bound with equality, i.e., there are exactly $r + 1$ sub-symbols in the code that are a function of it.

The main contribution of this paper is as follows:

1) For any two positive integers l, r, we construct a high-rate MSR code with r parity nodes, column length l, and $k = (r+1)\log_r l$ systematic nodes. In particular, for $r = 2$ parity nodes we obtain an MSR code with $k = 3\log_2 l$, moreover, the code is defined over a finite field of size $1 + 2\log_2 l$.
2) We rigorously state the necessary and sufficient conditions of linear optimal repair codes (similar results also seen in [6], [15], [16]), and thus enable explicit code constructions and simplify proofs of repair optimality.
3) We design optimal update codes with 2 parities and $k = 2\log_2 l$ systematic nodes. This construction exceeds the upper bound of $k \leq \log_2 l$ given by [18] for optimal update and diagonal encoding matrices. Diagonal encoding matrices means that the encoding is done only within each row in the information array. However our construction allows mixing of different rows in encoding. As a result, we can see a fundamental difference between these two types of codes.
4) We construct a family of codes that further reduces the access. We introduce a technique that equivalently transforms a linear code through block-diagonal matrix. This technique can be applied to an arbitrary MSR code and therefore can be a useful tool for transforming regenerating codes.

Even though our construction with $(r+1)\log_r l$ systematic nodes is incremental improvement for k compared to the code in [5] (and the recent work [1], [14]), which has $k = r\log_r l$, we point out here a few advantages of our work. Through the necessary and sufficient conditions of optimal repair codes, we are then able to explicitly write the code generating matrix in terms of eigenspaces and eigenvalues, whereas [5] constructed codes recursively by Kronecker product of matrices and multiplication of permutation matrices. Moreover, our technique of eigenspaces inspired new code constructions in recent work [10], where the eigenspaces are constructed from the partition of a set of basis vectors similar to our work, and a variety of codes with $r = 2$, $k = 2\log_2 l$, or $k = 3\log_2 l$ are constructed. Also in [5] the required finite field size of the code is not specified. But in our construction the finite field size is given explicitly for codes with 2 parities, and therefore can be practical for distributed storage applications.

From our code constructions and the upper bound of $k \leq 2(\log_2 l)(\log_{r/(r-1)} l + 1) + 1$ in [9], we can see that when r is fixed and l grows, the multiplicative gap for k between the converse and the achievability is in the order of $\log l$.

The rest of the paper is organized as follows: in Section II we formally introduce the MSR code problem. In Section III codes with r parity nodes are constructed, and we show that the code has $k = (r+1)\log_r l$. We show an optimal update code with $2\log_2 l$ systematic nodes and 2 parity nodes in Section IV, and discuss how to reduce the access ratio in Section V. Finally we conclude in Section VI.

II. Problem Settings

N1	N2	N3	N4	N5	N6
a	b	c	d	$a+b+c+d$	$2a+w+2b+3c+d$
w	x	y	z	$w+x+y+z$	$3w+b+3x+2y+z$

Figure 1. An $(n=6,k=4,l=2)$ MSR code over $\mathbb{F}_4$ constructed by the irreducible polynomial t^2+t+1. For simplicity the field elements $t, t+1$ are represented by the integers $2,3$, respectively. The first 4 nodes are systematic and the last 2 are parities. To repair $N1$ transmit the first row from every remaining node. To repair $N2$ transmit the second row. To repair $N3$ transmit the sum of both rows. And to repair $N4$ transmit the sum of the first row and 2 times the second row from nodes $N1, N2, N3, N5$, and the sum of the first row and 3 times the second row from node $N6$.

For positive integers $i \leq j$, $[i,j]$ and $[i]$ are used to represent the set $\{i, i+1, \ldots, j\}$ and the set $\{1,2,\ldots,i\}$, respectively. We use capital letters to represent matrices and subspaces of vector spaces. For an integer l and a finite field $\mathbb{F}$, let S, T be two subspaces of $\mathbb{F}^l$. Denote the sum of the subspaces by $S+T=\{s+t : s \in S, t \in T\}$ and by $S \oplus T$ if it is a direct sum. Let A be a matrix over the field $\mathbb{F}$ and denote the subspace spanned by its rows by $\mathrm{span}(A)$. If A is a square matrix of size $l \times l$ then $SA=\{vA : v \in S\}$ is the image of the subspace S under the action of A.

An (n,k,l) MSR code over the finite field $\mathbb{F}$ is an erasure-correcting code stored in an array of size $l \times n$ where each of the array entry is an element of the field $\mathbb{F}$. The information is stored in the first k columns (symbols, or nodes) which makes the code systematic and the remaining $r=n-k$ symbols form the parity symbols. Note here that the third parameter l in our notation is the length of the column instead of the minimum distance, as typically used in coding theory. The l entries in each column are called sub-symbols, which are elements in $\mathbb{F}$.

The MSR code has two properties. First, the optimal repair property indicates the efficiency in the repair during a failure of any of the k systematic nodes. More precisely, if systematic node $i \in [k]$ fails, then each of the $n-1$ remaining nodes transmits in the course of the repair process l/r sub-symbols, which equals to a fraction of $1/r$ of the information it stores. Second, the code satisfies the reconstruction property, i.e., the entire information can be decoded from the information stored on any k nodes.

Next in this section we discuss the encoding, the repair, and the reconstruction for an (n,k,l) MSR codes.

A. Encoding

For $i=1,...,n$, let $C_i \in \mathbb{F}^l$ be the i-th column of the code. Let the first k nodes store the information itself. Each parity node $C_{k+i}, i=1,...,r$, is a linear function of the information nodes, i.e., there exist invertible *encoding matrices* of size $l \times l$, $A_{i,j}$, $j=1,...,k$ such that

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

Define the *coding matrix* A as

$$A=\begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ A_{r,1} & A_{r,2} & \cdots & A_{r,k} \end{pmatrix}. \tag{2}$$

For example, in Figure 1, the encoding matrices are $A_{1,j}=I$ for all $j=1,..,4$, and

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

In our constructions, we require that $A_{1,j}=I$ 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 encoding matrices [18].

B. Repair

Consider a code with optimal repair. Assume that systematic node i is erased and is repaired by transmitting from each of the $n-1$ helper nodes a quantity of information equal to a fraction of $1/r$ of the information it stores. The information transmitted from helper node $j \in [n]\backslash\{i\}$ is computed using the subspace $S_{i,j} \subseteq \mathbb{F}^l$ of dimension l/r as follows. Let $\hat{S}_{i,j}$ be an $l/r \times l$ matrix whose rows form a basis for $S_{i,j}$, i.e., $\mathrm{span}(\hat{S}_{i,j})=S_{i,j}$. We transmit the vector $\hat{S}_{i,j}C_j$ of length l/r. $S_{i,j}$ is called the *repairing subspace* for erased node i and helper node j.

In the following we show necessary and sufficient conditions for optimal repair.

Claim 1 *[18] The existence of an (n,k,l) optimal repair code is equivalent to the following **subspace property**: For any $i \in [k]$ there exist subspaces $S_{i,j} \subseteq \mathbb{F}^l$, $j \neq i, j \in [n]$, all of dimension l/r, such that for all $j \in [k]\backslash\{i\}, s \in [r]$, the following holds*

$$S_{i,j}=S_{i,k+s}A_{s,j}, \tag{3}$$
$$\sum_{s=1}^{r} S_{i,k+s}A_{s,i}=\mathbb{F}^l. \tag{4}$$

Obviously, in (4) the dimension of each subspace $S_{i,j}A_{s,i}$ is no more than l/r, and the sum of r such subspaces has dimension no more than l. This means that these subspaces intersect only on the zero vector. Therefore, the sum is actually a direct sum of the subspaces, and every matrix $A_{s,i}$ is of full rank l.

Sketch of proof: Since it is an optimal repair code, each surviving node transmits exactly l/r sub-symbols during the repair process. Let $\hat{S}_{i,j}$ be a $l/r \times l$ matrix whose rows are a set of basis of $S_{i,j}$, $i \in [k], j \neq i, j \in [n]$. For repair we transmit $\hat{S}_{i,j}C_j$ from a systematic node $j \neq i, j \in [k]$, and $\hat{S}_{i,j}C_{k+s}=\sum_{z=1}^{k}\hat{S}_{i,j}A_{s,z}C_z$ from a parity node $k+s \in [k+1,k+r]$. Our

goal is to recover C_i and cancel out all C_j, $j \neq i, j \in [k]$. In order to cancel out C_j, (3) must be satisfied. In order to solve C_i, all equations related to C_i must have full rank l, so (4) is satisfied. One the other hand, if (3) (4) are satisfied, one can transmit $\hat{S}_{i,j}C_j$ from each node j, $j \neq i, j \in [n]$ and optimally repair the node i. ∎

Similar interference alignment technique was first introduced in [6] for regenerating codes. Also, [15] was the first to formally prove similar conditions.

It was shown in [18] that we can further simplify our repair strategy of node i and assume by equivalent transformation of the encoding matrices that

$$S_{i,j} = S_i, \text{ for all } j \neq i, j \in [n]. \tag{5}$$

Furthermore, [18] showed that such simplification can be applied to all systematic nodes w.l.o.g. except for at most one special node. For instance see $N4$ in Figure 1, and more details in Example 1. However in Sections III and IV, we will shorten the code by one node if such exception exists and assume identical $S_{i,j} = S_i$ for all $i \in [k]$.

Under simplification (5), the subspace property for optimal repair of a systematic node i becomes: There exist a subspace $S_i \subseteq \mathbb{F}^l$ of dimension l/r, such that for all $j \neq i, j \in [k], s \in [r]$,

$$S_i = S_i A_{s,j}, \tag{6}$$

$$\sum_{s=1}^{r} S_i A_{s,i} = \mathbb{F}^l. \tag{7}$$

Notice that if (6) is satisfied then S_i is an invariant subspace of $A_{s,j}$ for any $s = 1, ..., r$ and $j \neq i$. If $A_{s,j}$ is diagonalizable then it is uniquely defined by its eigenspaces and eigenvalues. Moreover each of the invariant subspaces of $A_{s,j}$ has a set of basis composed of eigenvectors of $A_{s,j}$. In our proposed constructions, we first focus on finding the proper eigenspaces of the encoding matrices. These eigenspaces uniquely define the set of invariant subspaces for each encoding matrix. Then we choose the eigenvalue that corresponds to each eigenspace, in order to ensure the reconstruction property discussed in the next subsection.

Example 1 *In Figure* 1, *the subspaces* $S_{i,j} = S_i$ *for* $i = 1, 2, 3$ *are*

$$S_1 = \text{span}(1, 0), S_2 = \text{span}(0, 1), S_3 = \text{span}(1, 1).$$

One can check that the subspace property (6), (7) *is satisfied for* $i = 1, 2, 3$. *For instance, in order to repair systematic node* $N3$, *we need to transmit the sum of the sub-symbols from each node, which is equivalent to multiplying each column by the matrix* $\hat{S}_3 = (1, 1)$. *Note that* $(1, 1)$ *is an eigenvector for* $A_{s,j}$, $s = 1, 2, j = 1, 2, 4$, *hence we have* $S_3 = S_3 A_{s,j}$, *where the equality is between the subspaces. Furthermore, it is easy to check that*

$$S_3 \oplus S_3 A_{2,3} = \text{span}(1, 1) \oplus \text{span}(t + 1, t) = \mathbb{F}_4^2.$$

Node $N4$ *is an exception, where the subspaces* $S_{4,j}$*'s are not equal. In fact* $S_{4,j} = \text{span}(1, t)$ *for* $j = 1, 2, 3, 5$, *and* $S_{4,6} = \text{span}(1, t + 1)$.

C. Reconstruction

If some r of the nodes are erased, the reconstruction property ensures that the entire information can be decoded from the remaining k nodes. More precisely, any $lk \times lk$ submatrix composed of k block rows of the generator matrix

$$\begin{pmatrix} I & & & \\ & I & & \\ & & \ddots & \\ & & & I \\ A_{1,1} & A_{1,2} & \cdots & A_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ A_{r,1} & A_{r,2} & \cdots & A_{r,k} \end{pmatrix}$$

is of full rank, where I is the identity matrix of order l. Equivalently any $t \times t$ block sub-matrix of the coding matrix A in (2) is invertible, for all $t \in [r]$. For diagonalizable matrices, we can use a finite field large enough and choose its eigenvalues such that the reconstruction property is satisfied.

For example, in Figure 1, any 1×1 and any 2×2 block sub-matrix of the coding matrix

$$A = \begin{pmatrix} I & I & I & I \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \end{pmatrix}$$

is invertible over $\mathbb{F}_4$, and thus the code satisfies the reconstruction property.

III. A Long MSR Code Construction

In this section, we construct an $(n = k + r, k = (r + 1)m, l = r^m)$ MSR code for any positive integers r, m. For given r, l, this code has the larger k compared to previously known results. We begin with the construction description and the proof of the optimal repair, continue by arguing that the entire information is reconstructible from any k nodes, and end with a discussion on the update and access complexity.

A. Construction

Let m be a positive integer, and fix $l = r^m$. The code is uniquely defined by the encoding matrices $A_{s,i}, s \in [r], i \in [k]$, in (2). Fix k diagonalizable matrices A_i, $i \in [k]$, of size $l \times l$. We define the encoding matrix $A_{s,i}$ for parity node $k + s$ and systematic node i as

$$A_{s,i} = A_i^{s-1}, s \in [r], i \in [k]. \tag{8}$$

Each matrix A_i is assumed to have r distinct nonzero eigenvalues $\lambda_{i,j}$ that correspond to r eigenspaces $V_{i,j}$ of dimension $l/r = r^{m-1}$, for $j = 0, ..., r - 1$.

Remarks:

1) Since $A_{1,i} = A_i^0 = I$, the first parity node is a simply the row sum of the array.
2) By the definition of $A_{s,i}$ it is clear that it has eigenvalues $\lambda_{i,0}^{s-1}, \lambda_{i,1}^{s-1}, \ldots, \lambda_{i,r-1}^{s-1}$.
3) Let V be a subspace of the eigenspace $V_{i,j}$ then the action of the linear transformation A_i on V results in

$$VA_i = \{vA_i : v \in V\} = \{\lambda_{i,j} v : v \in V\} = V, \tag{9}$$

where the last equality follows since $\lambda_{i,j} \neq 0$. In other words, any subspace of $V_{i,j}$ is an invariant subspace of A_i.

4) The repairing subspace of node $i \in [k]$ will be S_i, and therefore by (8) the **subspace property** (6)-(7) becomes

$$S_i = S_i A_j, \forall j \neq i, j \in [k] \tag{10}$$

$$S_i + S_i A_i + S_i A_i^2 + \cdots + S_i A_i^{r-1} = \mathbb{F}^l. \tag{11}$$

In the first step of our construction we define the eigenspaces of each matrix A_i without specifying the eigenvalues. This will be enough to show the optimal repair property of the code regardless of the exact value of the eigenvalues. In the second step we will show that over a large finite field, there exists an assignment for the eigenvalues that guarantees the reconstruction property as well.

Let $\{e_a : a = 0, ..., l-1\}$ be some basis of $\mathbb{F}^l$, for example, one can think of them as the standard basis vectors. We introduce some notations on the subscript a, $a \in [0, r^m - 1]$. By abuse of notation, we write $a = (a_1, a_2, \ldots, a_m)$, where $a = \sum_i a_i r^{m-i}$ and each $a_i \in [0, r-1]$. Clearly this is the r-ary expansion of a and a_i is referred as the i-th digit of a. Define $M_{a,i}$ to be the r indices in $[0, r^m - 1]$ whose digits equal to the digits of a maybe except the i-th digit:

$$\begin{aligned} M_{a,i} =& \{a' = (a'_1, \ldots, a'_m) \in [0, r^m - 1] : \\ & a'_j = a_j, j \in [m] \backslash \{i\}\}. \end{aligned}$$

For example, when $r = 3, m = 4$, we have $e_5 = e_{(0,0,1,2)}$, and $M_{5,3} = \{(0,0,0,2) = 2, (0,0,1,2) = 5, (0,0,2,2) = 8\}$. Similarly, for two indices i, j we define $M_{a,\{i,j\}}$ as the r^2 indices in $[0, r^m - 1]$ whose digits equal to the digits of a except maybe the i-th and the j-th digits:

$$\begin{aligned} M_{a,\{i,j\}} =& \{a' = (a'_1, \ldots, a'_m) \in [0, r^m - 1] : \\ & a'_u = a_u, u \in [m] \backslash \{i, j\}\}. \end{aligned}$$

Next we define $(r+1)m$ subspaces that will be used for the encoding matrices and repairing subspaces. Let $i \in [m], u \in [0, r]$, and define

$$\begin{aligned} P_{i,u} &= \text{span}(e_a : a_i = u), \\ P_{i,r} &= \text{span}(\sum_{a' \in M_{a,i}} e_{a'} : a \in [0, r^m - 1]). \end{aligned} \tag{12}$$

For $u \neq r, P_{i,u}$ is spanned by the set of basis vectors whose i-th digit equals u, and therefore $P_{i,u}$ is of dimension l/r. Similarly, one can check that also $P_{i,r}$ is a subspace of dimension l/r.

For example, when $r = 3, m = 2$,

$$\begin{aligned} P_{1,0} &= \text{span}(e_{(0,0)}, e_{(0,1)}, e_{(0,2)}) = \text{span}(e_0, e_1, e_2), \\ P_{1,1} &= \text{span}(e_3, e_4, e_5), \\ P_{1,2} &= \text{span}(e_6, e_7, e_8), \\ P_{1,3} &= \text{span}(e_0 + e_3 + e_6, e_1 + e_4 + e_7, e_2 + e_5 + e_8). \end{aligned}$$

Using these $k = (r+1)m$ subspaces, we have a code construction as below which will be proven to be optimal repair.

Construction 1 *We construct an $(n = (r+1)m + r, k = (r+1)m, l = r^m)$ code. Let $u \in [0, r], i \in [m]$. For each $um + i \in [k]$, let the matrix A_{um+i} have eigenspaces $P_{i,u'}$, $u' \neq u$, that correspond to distinct nonzero eigenvalues. Let the encoding matrices be defined as in* (8)*. Furthermore, let $S_{um+i} = P_{i,u}$ be the repairing subspace of node $um + i$.*

Example 2 *Removing node N4 of the code in Figure* 1 *yields an $(n = 5, k = 3, l = 2)$ code constructed using Construction* 1 *with $r = 2, m = 1$.*

In Figure 2 *we list the subspaces $\{P_{i,u}\}$ for $r = 2, m = 2$, and in Figure* 3 *we list the corresponding $(n = 8, k = 6, l = 4)$ code. One can check the subspace property 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 $S_i \oplus S_i A_i = \mathbb{F}^4$, for $i \in [6]$.*

Figure 4 *illustrates the subspaces $\{P_{i,u}\}$ for $r = 3$, $m = 2$. Figure* 5 *is an $(n = 11, k = 8, l = 9)$ code constructed from these subspaces with* 8 *systematic nodes. One can see that the subspace property holds. Namely, if a systematic node is erased, one can transmit only a subspace of dimension* 3 *to repair, which corresponds to the optimal* 1/3 *repair bandwidth. Recall that the three encoding matrices for systematic node i are I, A_i, A_i^2, for $i \in [8]$.*

The following lemma shows that intersections between the subspaces $\{P_{i,u}\}$ have a certain structure that will be useful to prove the repair optimality of Construction 1.

Lemma 2 *Let $i, j \in [m]$ and $u, u' \in [0, r]$, such that $(i, u) \neq (j, u')$, i.e., the vectors differ in at least one coordinate, then the following holds,*

$$\oplus_{s \neq u'} (P_{i,u} \cap P_{j,s}) = P_{i,u}. \tag{13}$$

Proof: If $i = j$ then $u \neq u'$ and it is easy to verify (13). Assume that i and j are distinct, then it is also easy to check that

$$P_{i,u} \cap P_{j,s} = \begin{cases} \text{span}(e_a : a \in [0, r^m - 1], a_i = u, a_j = s), & u < r, s < r, \\ \text{span}(\sum_{a' \in M_{a,j}} e_{a'} : a \in [0, r^m - 1], a_i = u), & u < r, s = r, \\ \text{span}(\sum_{a' \in M_{a,i}} e_{a'} : a \in [0, r^m - 1], a_j = s), & u = r, s < r, \\ \text{span}(\sum_{a' \in M_{a,\{i,j\}}} e_{a'} : a \in [0, r^m - 1]), & u = r, s = r, \end{cases} \tag{14}$$

and each of these subspaces is of dimension $l/r^2 = r^{m-2}$. Notice that for distinct s, s' the subspaces $P_{j,s}, P_{j,s'}$ intersect trivially and therefore the vector space $\oplus_{s \neq u'} (P_{i,u} \cap P_{j,s})$ is indeed a direct sum of vector spaces. Furthermore, each vector space is a subspace of $P_{i,u}$, hence by dimensionality $\oplus_{s \neq u'} (P_{i,u} \cap P_{j,s}) = P_{i,u}$ as needed. ∎

	$P_{1,0}$	$P_{1,1}$	$P_{1,2}$	$P_{2,0}$	$P_{2,1}$	$P_{2,2}$
Basis for the subspace	e_0 e_1	e_2 e_3	e_0+e_2 e_1+e_3	e_0 e_2	e_1 e_3	e_0+e_1 e_2+e_3

Figure 2. Basis vectors of the subspaces $P_{i,u}$ constructed in Construction 1 with parameters $r=2, m=2, n=8, k=6, l=4$.

Node index i	1	2	3	4	5	6
Basis for 1st eigenspace of A_i	e_0+e_2 e_1+e_3	e_0+e_1 e_2+e_3	e_0 e_1	e_0 e_2	e_0 e_1	e_0 e_2
Basis for 2nd eigenspace of A_i	e_2 e_3	e_1 e_3	e_0+e_2 e_1+e_3	e_0+e_1 e_2+e_3	e_2 e_3	e_1 e_3
Basis for repairing subspace S_i	e_0 e_1	e_0 e_2	e_2 e_3	e_1 e_3	e_0+e_2 e_1+e_3	e_0+e_1 e_2+e_3

Figure 3. (n=8,k=6,l=4) MSR code. The first parity node is the row sum of the array, and the second parity is computed using the encoding matrices A_i. Each encoding matrix is defined by its two eigenspaces of dimension 2. In order to repair node i, each surviving node projects its information on the repairing subspace S_i, namely node $j \neq i$ transmits the multiplication $\hat{S}_i C_j$ E.g., node 5 has two distinct eigenspaces $\text{span}(e_0, e_1), \text{span}(e_2, e_3)$. Furthermore, if this node is lost, each surviving node projects its information on the subspace $S_5 = \text{span}(e_0+e_2, e_1+e_3)$.

	$P_{1,0}$	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	$P_{2,0}$	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$
Basis for the subspace	e_0 e_1 e_2	e_3 e_4 e_5	e_6 e_7 e_8	$e_0+e_3+e_6$ $e_1+e_4+e_7$ $e_2+e_5+e_8$	e_0 e_3 e_6	e_1 e_4 e_7	e_2 e_5 e_8	$e_0+e_1+e_2$ $e_3+e_4+e_5$ $e_6+e_7+e_8$

Figure 4. Subspaces $P_{i,u}$ used to construct a code with $r=3, m=2, n=11, k=8, l=9$.

Node index i	1	2	3	4	5	6	7	8
The 3 eigenspaces	$P_{1,3}$ $P_{1,1}$ $P_{1,2}$	$P_{2,3}$ $P_{2,1}$ $P_{2,2}$	$P_{1,0}$ $P_{1,3}$ $P_{1,2}$	$P_{2,0}$ $P_{2,3}$ $P_{2,2}$	$P_{1,0}$ $P_{1,1}$ $P_{1,3}$	$P_{2,0}$ $P_{2,1}$ $P_{2,3}$	$P_{1,0}$ $P_{1,1}$ $P_{1,2}$	$P_{2,0}$ $P_{2,1}$ $P_{2,2}$
Repairing subspace	$P_{1,0}$	$P_{2,0}$	$P_{1,1}$	$P_{2,1}$	$P_{1,2}$	$P_{2,2}$	$P_{1,3}$	$P_{2,3}$

Figure 5. An $(n=11, k=8, l=9)$ code. The subspaces $P_{i,u}$ are listed in Figure 4.

For example, when $r=2, m=2$, it is easy to check from Figure 2 that

$$\begin{aligned}
P_{1,0} \cap P_{2,1} &= \text{span}(e_a : a_1 = 0, a_2 = 1) = \text{span}(e_1), \\
P_{1,0} \cap P_{2,2} &= \text{span}(\sum_{a' \in M_{a,2}} e_{a'} : a_1 = 0) = \text{span}(e_0 + e_1), \\
P_{1,2} \cap P_{2,1} &= \text{span}(\sum_{a' \in M_{a,1}} e_{a'} : a_2 = 1) = \text{span}(e_1 + e_3), \\
P_{1,2} \cap P_{2,2} &= \text{span}(\sum_{a' \in M_{a,\{1,2\}}} e_{a'} : a' \in [0,3]) \\
&= \text{span}(e_0 + e_1 + e_2 + e_3).
\end{aligned}$$

As a result,

$$\begin{aligned}
P_{1,0} \cap P_{2,1} + P_{1,0} \cap P_{2,2} &= \text{span}(e_1, e_0 + e_1) = P_{1,0}, \\
P_{1,2} \cap P_{2,1} + P_{1,2} \cap P_{2,2} &= \text{span}(e_1 + e_3, e_0 + e_1 + e_2 + e_3) \\
&= P_{1,2}.
\end{aligned}$$

We note here that the property in Lemma 2 is a consequence of our special choice of the subspaces. In particular, let $P, T_1, ..., T_r$ be some subspaces of $\mathbb{F}^l$, each with dimension l/r that satisfy $\oplus_{i=1}^r T_i = \mathbb{F}^l$. Then the equality $\sum_{i=1}^r (P \cap T_i) = P$ does not necessarily hold for arbitrarily chosen subspaces T_i.

The following theorem shows that the code indeed has optimal repair.

Theorem 3 *Construction* 1 *has optimal repair of* $1/r$ *when repairing one systematic node.*

Proof: We need to show that the subspace property (10) and (11) holds. For distinct integers $um+i, u'm+j \in [k]$, where $u, u' \in [0, r-1]$, and $i, j \in [m]$, we will show that (10) is satisfied, namely

$$S_{um+i} A_{u'm+jx} = S_{um+i}.$$

The r eigenspaces of $A_{u'm+i'}$ are $P_{j,s}$ for $s \neq u'$. The repairing subspace is $S_{um+i} = P_{i,u}$. Then,

$$\begin{aligned}
S_{um+i} A_{u'm+j} &= P_{i,u} A_{u'm+j} \\
&= (\oplus_{s \neq u'} P_{i,u} \cap P_{j,s}) A_{u'm+j} \qquad (15) \\
&= \oplus_{s \neq u'} P_{i,u} \cap P_{j,s} \qquad (16) \\
&= P_{i,u} \qquad (17) \\
&= S_{um+i}.
\end{aligned}$$

Here (15) and (17) follow by Lemma 2, and (16) follows from (9) and the fact that $P_{j,s}$ is an eigenspace of $A_{u'm+j}$.

Next we show that (11) holds. Denote $A = A_{um+i}, S = S_{um+i}$, and (11) becomes

$$S + SA + .. + SA^{r-1} = \mathbb{F}^l.$$

Assume that $u = 0$, for the other cases the result follows in a similar way. Let $\lambda_0, \lambda_1, \ldots, \lambda_{r-1}$ be the r distinct eigenvalues

of A that correspond to the eigenspaces $P_{i,r}, P_{i,1}, \ldots, P_{i,r-1}$, respectively.

For a vector $a = (a_1, a_2, \ldots, a_m)$ or equivalently an integer $a \in [0, l-1]$, denote by $a_i(u) = (a_1, \ldots, a_{i-1}, u, a_{i+1}, \ldots, a_m)$ the vector that is the same as a except the i-th digit, which is u. Notice that $S = P_{i,0} = \text{span}(e_{a_i(0)}, \forall a \in [0, l-1])$ and for $e_{a_i(0)} \in S$,

$$\begin{aligned} e_{a_i(0)}A^s =& (\sum_{u=0}^{r-1} e_{a_i(u)} - \sum_{u=1}^{r-1} e_{a_i(u)})A^s \\ =& (\lambda_0^s \sum_{u=0}^{r-1} e_{a_i(u)} - \sum_{u=1}^{r-1} \lambda_u^s e_{a_i(u)})A^s \\ =& \lambda_0^s e_{a_i(0)} + \sum_{u=1}^{r-1} (\lambda_0^s - \lambda_u^s) e_{a_i(u)}. \end{aligned}$$

Writing the equations for all $s \in [0, r-1]$ in a matrix, we get

$$\begin{pmatrix} e_{a_i(0)} \\ e_{a_i(0)}A \\ e_{a_i(0)}A^2 \\ \vdots \\ e_{a_i(0)}A^{r-1} \end{pmatrix} = M \begin{pmatrix} e_{a_i(0)} \\ e_{a_{i(1)}} \\ \vdots \\ e_{a_i(r-1)} \end{pmatrix},$$

with

$$M = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \lambda_0 & \lambda_0 - \lambda_1 & \cdots & \lambda_0 - \lambda_{r-1} \\ \lambda_0^2 & \lambda_0^2 - \lambda_1^2 & \cdots & \lambda_0^2 - \lambda_{r-1}^2 \\ \vdots & \vdots & & \vdots \\ \lambda_0^{r-1} & \lambda_0^{r-1} - \lambda_1^{r-1} & \cdots & \lambda_0^{r-1} - \lambda_{r-1}^{r-1} \end{pmatrix}.$$

After a sequence of elementary column operations, M becomes the following Vandermonde matrix

$$M' = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_0 & \lambda_1 & \cdots & \lambda_{r-1} \\ \lambda_0^2 & \lambda_1^2 & \cdots & \lambda_{r-1}^2 \\ \vdots & \vdots & & \vdots \\ \lambda_0^{r-1} & \lambda_1^{r-1} & \cdots & \lambda_{r-1}^{r-1} \end{pmatrix}.$$

Since λ_i's are distinct, we know M' and hence M is non-singular. Therefore, $\text{span}(e_{a_i(0)}, e_{a_i(0)}A, \ldots, e_{a_i(0)}A^{r-1}) = \text{span}(e_{a_i(0)}, e_{a_i(1)}, \ldots, e_{a_i(r-1)})$. Since S contains $e_{a_i(0)}$ for all r-ary vector $a \in [0, l-1]$, we get that

$$S + SA + \cdots + SA^{r-1} = \text{span}(e_a : a \in [0, l-1]) = \mathbb{F}^l.$$

Hence the code has optimal repair. ∎

B. Reconstruction and finite field size

Next we show that any code satisfying the subspace property can be modified to satisfy the reconstruction property over a large finite field.

Theorem 4 *Consider a systematic code over $\mathbb{F}$ satisfying the subspace property (6) (7). If $\mathbb{F}$ is a field large enough then one can modify the code to ensure it is an MSR code.*

Proof: Multiply the encoding matrices $A_{s,i}$ by distinct nonzero variables $x_{(s-1)k+i}$, $i \in [k], s \in [r]$, to get a new systematic code defined by the coding matrix

$$\begin{bmatrix} x_1 A_{1,1} & \cdots & x_k A_{1,k} \\ \vdots & \ddots & \vdots \\ x_{(r-1)k+1} A_{r,1} & \cdots & x_{rk} A_{r,k} \end{bmatrix}. \quad (18)$$

Notice that since the subspace property (6)-(7) is satisfied for the code defined by the encoding matrices $A_{s,j}$, it is also satisfied for the new code (18) with the same repairing subspaces S_i. This follows since

$$S_i x_{(s-1)k+j} A_{s,j} = \{x_{(s-1)k+j} v A_{s,j} : v \in S_i\} = S_i A_{s,j} = S_i.$$

We conclude that the code defined in (18) has optimal repair.

We next show the reconstruction property. Clearly the new code satisfies the reconstruction property iff any $t \times t$ block sub-matrix in (18) is invertible, for any $t \in [r]$. Define the multivariate polynomial P in the variables $x_{(s-1)k+i}$'s, which is the product of the determinants of all the $t \times t$ block submatrices, for all $t = 1, ..., r$. Hence, the code can be made to satisfy the reconstruction property if there is an assignment to the variables that does not evaluate P to zero. More precisely, fix $t \in [r]$ and let $a = \{a_1, a_2, ..., a_t\} \subseteq [r]$ and $b = \{b_1, b_2, ..., b_t\} \subseteq [k]$ be two sets of indices of size t. Define $P_{a,b}$ to be the determinant of the sub-matrix of (18) restricted block rows a and block columns b. Clearly it is a polynomial of degree lt. Let $x = (x_1, ..., x_{rk})$ be the vector of variables and define the polynomial $P(x)$ as

$$P(x) = \prod_{t=1}^{r} \prod_{\substack{a \subseteq [r], |a|=t \\ b \subseteq [k], |b|=t}} P_{a,b}.$$

Denote by x^d the term $\prod_i x_i^{d_i}$ of degree $\sum_i d_i$. Furthermore, define the usual ordering on the terms x^d according to the lexicographic order, i.e., $x^d > x^{d'}$ iff $d_i > d'_i$ and i is minimum index j such that $d_j \neq d'_j$. The leading coefficient of a multivariate polynomial is the coefficient of the maximal nonzero term. For example, the leading coefficient of the polynomial $2x_1^2 x_3 + x_1^2 x_4$ is 2.

Notice that the leading coefficient of the polynomial $P_{a,b}$ equals $\prod_{i=1}^{t} \det(A_{a_i,b_i})$, which is nonzero, since by (7) each matrix A_{a_i,b_i} is invertible. Moreover if $P_{a,b}, P_{a',b'}$ are the determinants of different submatrices, then the leading coefficient of their product $P_{a,b} \cdot P_{a',b'}$, is the product of their leading coefficients. If the two leading coefficients are nonzero, so is their product. P is a product of such polynomials $P_{a,b}$, each corresponds to a $t \times t$ sub-matrix of (18), therefore its leading coefficient is also nonzero. Moreover, each $P_{a,b}$ is a homogeneous polynomial, hence so is P. We conclude that P is a nonzero polynomial since it has a nonzero term, say x^d. By the Combinatorial Nullstellensatz [2] we get that over a field of size greater than $\max_i\{d_i : d = (d_1, ..., d_{rk})\}$ there is an assignment for the variables x_i such that $P(x) \neq 0$, and that concludes the proof. ∎

From the above proof, one can see that a code can be constructed by firstly finding encoding matrices and subspaces $A_{s,j}, S_i$ that satisfy the subspace property, and secondly finding

coefficients $x_{(s-1)k+j}$ for the encoding matrices to guarantee the reconstruction property.

For the case of $r = 2$ parities, we can construct a $(3m + 2, 3m, 2^m)$ MSR code by explicitly specifying the eigenvalues of the encoding matrices and the finite field size.

Construction 2 *Consider the code in Construction* 1 *with* $r = 2$. *Let* $\{\lambda_{i,j}\}_{i\in[m],j=1,2}$ *be arbitrary* $2m$ *distinct nonzero elements of the field* $\mathbb{F}_q$, $q \geq 2m + 1$. *Recall that for* $u \in \{0, 1, 2\}$ *and* $i \in [m]$, *the eigenspaces of the encoding matrix* A_{um+i} *are* $P_{i,(u+1) \bmod 3}$ *and* $P_{i,(u+2) \bmod 3}$. *Assign to these two eigenspaces the eigenvalues* $\lambda_{i,1}$ *and* $\lambda_{i,2}$, *respectively.*

In Figure 6 we show the eigenvalues of the encoding matrices A_i, A_{m+i}, A_{2m+i}. Take the code of $m = 2$ in Figure 3, we can use finite field $\mathbb{F}_5$ and assign the eigenvalues as in Figure 7.

Theorem 5 *Construction* 2 *constructs a* $(3m + 2, 3m, 2^m)$ *MSR code over any field of size at least* $2m + 1$.

Proof: From Theorem 3 and Theorem 4 we know that Construction 2 is optimal repair. We only need to show that it satisfies the reconstruction property, namely, any two erasures can be repaired. This is equivalent to that (i) all the encoding matrices A_x's are invertible, and (ii) any 2×2 block sub-matrix

$$\begin{bmatrix} I & I \\ A_x & A_y \end{bmatrix}$$

is invertible, for any distinct $x, y \in [k]$. Since the eigenvalues are nonzero the first condition is satisfied. The second condition is equivalent to that $A_x - A_y$ is invertible.

Let $x = um + i, y = vm + j$, with $i, j \in [u], u, v \in \{0, 1, 2\}$. We claim that if a is an eigenvector of both A_x and A_y with eigenvalues μ_x and μ_y, respectively, then $\mu_x \neq \mu_y$. This follows since if $i \neq j$ then the eigenvalues of A_{um+i} are $\lambda_{i,1}, \lambda_{i,2}$ which by construction are distinct from $\lambda_{j,1}, \lambda_{j,2}$, the eigenvalues of A_{vm+j}. If $i = j, u \neq v$, then A_x and A_y have a common eigenspace and a belongs to it. More precisely, the eigenspaces of A_{um+i} and A_{vm+i} are $P_{i,(u+1) \bmod 3}, P_{i,(u+2) \bmod 3}$ and $P_{i,(v+1) \bmod 3}, P_{i,(v+2) \bmod 3}$, respectively. W.l.o.g. assume that $P_{i,(u+1) \bmod 3} = P_{i,(v+2) \bmod 3}$. By construction $P_{i,(u+1) \bmod 3}$ and $P_{i,(v+2) \bmod 3}$ have eigenvalues $\lambda_{i,1}$ and $\lambda_{i,2}$, respectively, which are again distinct by construction.

Let the two eigenspaces of A_{um+i}, A_{vm+j} be V_1, V_2 and U_1, U_2, respectively, which correspond to eigenvalues λ_1, λ_2, and μ_1, μ_2. Clearly

$$V_1 \oplus V_2 = U_1 \oplus U_2 = \mathbb{F}^l.$$

By Lemma 2

$$\oplus_{i=1}^2 \oplus_{j=1}^2 (V_i \cap U_j) = \oplus_{i=1}^2 (V_i) = \mathbb{F}^l.$$

Thus the space $\mathbb{F}^l$ is the direct sum of 4 subspaces, $V_i \cap U_j$, $i, j = 1, 2$. Assume to the contrary that $A_x - A_y$ is not invertible. Then there exists a nonzero vector a such that

$$a(A_{um+i} - A_{vm+j}) = 0,$$

where $a = \sum_{i,j=1}^2 a_{i,j}$, and $a_{i,j} \in V_i \cap U_j$. Then,

$$\begin{aligned} 0 =& a(A_{um+i} - A_{vm+j}) \\ =& (\lambda_1 - \mu_1)a_{1,1} + (\lambda_1 - \mu_2)a_{1,2} \\ &+ (\lambda_2 - \mu_1)a_{2,1} + (\lambda_2 - \mu_2)a_{2,2}. \end{aligned} \tag{19}$$

Since a is nonzero, at least one of the $a_{i,j}$'s is nonzero, say $a_{1,1}$. Note that $a_{1,1}$ is a common eigenvector of A_{um+i} and A_{vm+j} therefore $\lambda_1 - \mu_1 \neq 0$ and the RHS of (19) is a non-trivial linear combination of linearly independent vectors which equals to zero, and we get a contradiction. ■

One can observe that the proposed code construction has parameters $n = 3m + 2, k = 3m, l = 2^m$, and a field size that scales linearly with the number of systematic nodes. On the other hand, the $(m + 3, m + 1, 2^m)$ MSR code in [17] requires only a field of size 3. Thus, Construction 2 constructs a longer code, but has longer (actual) column length since it has to store 2^m symbols of a *larger* field. Nonetheless, it may be possible to modify the structure of the encoding matrices a bit , to obtain a code that requires only a constant field size. This remains as a future research direction.

C. Update and access complexity

Next we make some remarks on the shortened codes of Construction 1. The shortening technique was also used in [15] [16] in order to get optimal repair code with different code rates. We discuss the update and access complexity of the shortened codes.

1) The code restricted to the systematic nodes $i \in [m], u = r$ is equivalent to that of [4], [17] (e.g., Nodes 5,6 in Figure 3). Since the encoding matrices A_i^{s-1} (when represented under the standard basis vectors) are all diagonal, every information sub-symbol appears exactly once in each of the parities, and therefore it appears $r + 1$ times in the code, once in each of the parities and once in its systematic node. This is an *optimal update* code (see more discussions in Section IV). The upper bound of $k \leq \log_2 l$ for optimal update codes with 2 parities and diagonal encoding matrices was proved in [18]. Therefore, the shortened $(n = \log_2 l + 2, k = \log_2 l, l)$ MSR code has the largest possible number of systematic nodes among such codes. However, when the encoding matrices are not diagonal, we will show in Section IV a construction of an optimal update MSR code with $k = 2 \log_2 l$.
2) Shortening the code to contain only the rm systematic nodes $i \in [m], u \in [0, r - 1]$ results in a code $\mathcal{C}$ that is equivalent to the code in [5] (e.g. Nodes $1, 2, 3, 4$ in Figure 3). Each repairing subspace $P_{i,u}$ can be represented by an $l/r \times l$ matrix, whose rows are $\{e_a : a_i = u\}$, such that each row has exactly one nonzero entry. Therefore when repairing a node, only l/r sub-symbols from each surviving node are read and transmitted, with no need of any computations within the surviving node . Such a code is termed to have *optimal access*. We note here that the $(n = 2 + 2 \log_2 l, k = 2 \log_2 l, l)$ shortened code has the largest possible k over all optimal access

encoding matrix	A_i	A_{m+i}	A_{2m+i}
1st (eigenspace,eigenvalue)	$(P_{i,1}, \lambda_{i,1})$	$(P_{i,2}, \lambda_{i,1})$	$(P_{i,0}, \lambda_{i,1})$
2nd (eigenspace,eigenvalue)	$(P_{i,2}, \lambda_{i,2})$	$(P_{i,0}, \lambda_{i,2})$	$(P_{i,1}, \lambda_{i,2})$

Figure 6. Pairs of (eigenspace, eigenvalue) assignment for nodes $i, m+i, 2m+i$ by Construction 2.

Node index	1	2	3	4	5	6
(i,u)	$(1,0)$	$(2,0)$	$(1,1)$	$(2,1)$	$(1,2)$	$(2,2)$
Eigenvalues	$\{1, 4\}$	$\{2, 3\}$	$\{1, 4\}$	$\{2, 3\}$	$\{1, 4\}$	$\{2, 3\}$

Figure 7. Eigenvalue assignment for $m = 2, r = 2, k = 6, l = 4$. The eigenvalues are elements in the field $\mathbb{F}_5$.

codes with 2 parities. In particular, [18] showed that a $(k+2, k, l)$ optimal access MSR code has the upper bound $k \leq 2\log_2 l$.

3) We conclude that the code construction is a combination of the longest optimal access code and the longest optimal update code (with diagonal encoding matrices).

Optimal access implies low complexity for repair, and therefore is very attractive in practice. As mentioned above, in [18] it was shown that an optimal access code with 2 parities have at most $2\log_2 l$ systematic nodes. For codes with $k > 2\log_2 l$, it is desirable that the code contains a subset of $2\log_2 l$ systematic nodes such that optimal access is achieved when repairing these nodes. What is the largest k satisfying the above condition? This question can be phrased equivalently as follows. Consider a $(2+2\log_2 l, 2\log_2 l, l)$ optimal access code $\mathcal{C}$. How much can we extend $\mathcal{C}$ by adding more systematic nodes, while still maintaining the optimal repair property? In Theorem 7 we resolve this question by showing that $\mathcal{C}$ can be extended by at most $\log_2 l$ systematic nodes, which would lead to a code with $k \leq 3\log_2 l$ systematic nodes. Notice that Construction 1 attains this bound with equality.

Before proving the theorem we will need the following lemma.

Lemma 6 *[18, Lemma 8] Consider a $(k+2, k, l)$ MSR code with repairing subspaces S_i for $i \in [k]$, then for any subset of indices $J \subseteq [k]$,*

$$\dim(\cap_{i\in J} S_i) \leq \frac{l}{2^{|J|}}.$$

Theorem 7 *Let $\mathcal{C}$ be an $(2m+2, 2m, 2^m)$ optimal access code, and let $\mathcal{D}$ be an $(k+2, k, 2^m)$ optimal repair code that is an extension of $\mathcal{C}$, then the number of systematic nodes in $\mathcal{D}$ satisfies $k \leq 3m$.*

Proof: Let $\mathcal{C}$ be an optimal access code of length $2m$ with 2 parities. Let $\mathcal{D}$ be an extended code of $\mathcal{C}$. By transforming the coding matrix of $\mathcal{D}$ (see [18], Section II), we can always assume the encoding matrices of the parities in $\mathcal{D}$ are

$$\begin{pmatrix} I & \cdots & I & I & \cdots & I \\ A_1 & \cdots & A_{2m} & A_{2m+1} & \cdots & A_k \end{pmatrix}.$$

Here the first $2m$ column blocks correspond to the encoding matrices of $\mathcal{C}$. First consider the code $\mathcal{C}$ that corresponds to the first $2m$ systematic nodes. If $\mathcal{C}$ has optimal access, then each repairing subspace is spanned by $l/2$ standard basis vectors. Since $\mathcal{C}$ contains $2m$ systematic nodes, on average each standard basis vector appears in $2m \times \frac{l}{2} \times \frac{1}{l} = m$ repairing subspaces. Fix $i \in [0, l-1]$ and let $J = \{j : e_i \in S_j\}$ be the set of indices of the repairing subspaces that contain the standard basis vector e_i. We claim that the size of J is m. By Lemma 6

$$1 \leq \dim(\cap_{j\in J} S_j) \leq \frac{2^m}{2^{|J|}},$$

hence $|J| \leq m$. Moreover, if J is of size less than m, then by a simple counting argument we get that there exists $e_{j'}$ and its set of indices J' of size greater than m, which is a contradiction. Hence, we conclude that for each i the size of J is exactly m and,

$$\text{span}(e_i) = \cap_{i\in J} S_i.$$

Now consider a systematic node $j \in [2m+1, k]$ that is added to the code $\mathcal{C}$. Since $\mathcal{D}$ is an optimal repair code, each repairing subspace of the nodes in $\mathcal{C}$ is an invariant subspace of A_j. Since the intersection of invariant subspaces is again an invariant subspace we get that $\cap_{i\in J} S_i = \text{span}(e_i)$ is an invariant subspace of A_j, for any $j \neq i$, $i = 0, ..., l-1$. Namely, each standard basis vector is an eigenvector of A_j, and therefore A_j is a diagonal matrix. We conclude that restricting the code $\mathcal{D}$ to its last $k-2m$ systematic nodes will yield an optimal update code. By Theorem 9 in [18], there are only m nodes that are all optimal update with diagonal encoding matrices, hence $k - 2m \leq m$. ■

IV. Long optimal Update Code

Recall each entry in the $l \times k$ information array is called a sub-symbol. When an information sub-symbol updates its value, all the corresponding parity sub-symbols that are functions of it need also to be updated. Since update is one of the most frequent operation in the system, one would like to minimize the amount of sub-symbol updates incurred by one information sub-symbol update. By the reconstruction property each parity node (symbol) is a function of the *entire* information, hence at least one parity sub-symbol needs to be updated in any update of an information sub-symbol. An *optimal update* MSR code attains this lower bound, namely each parity node updates exactly one of its sub-symbols for each information sub-symbol update. In linear codes the condition for optimal update property is equivalent to each encoding matrix is a generalized permutation matrix, i.e. it has exactly one nonzero entry in each row and each column.

Consider codes with two parities. For diagonal encoding matrices, which are a special case of generalized permutation matrices, a $(k+2,k,l)$ MSR code satisfies $k \leq \log_2 l$ [18]. In this section we will show that by dropping the constraint on the encoding matrices being diagonal, one can improve the number of systematic nodes in an optimal update MSR code by a factor of two.

Similar to the previous section, we first define a set of subspaces that will be used as encoding matrix eigenspaces and repairing subspaces. Let $l = 2^m$ for some integer m and consider a field $\mathbb{F}$ where 2 is not the characteristic. Let x and y be nonzero elements of $\mathbb{F}$. For $i = 1, ..., m$ define the following four subspaces of $\mathbb{F}^l$ of dimension $l/2$:

$$\begin{aligned}
P_{i,0} =& \text{span}(e_a : a \in [0, l-1], a_i = 0),\\
P_{i,1} =& \text{span}(ye_a + xe_b : a, b \in [0, l-1], a_i = 0, b_i = 1,\\
&\text{and for all } j \neq i, a_j = b_j),\\
P_{i,2} =& \text{span}(e_a : a \in [0, l-1], a_i = 1),\\
P_{i,3} =& \text{span}(-ye_a + xe_b : a, b \in [0, l-1], a_i = 0, b_i = 1,\\
&\text{and for all } j \neq i, a_j = b_j),
\end{aligned} \tag{20}$$

For the following construction, we let the encoding matrices for the 1st parity be $A_{1,j} = I$, and denote by $A_j = A_{2,j}$ the encoding matrix for the 2nd parity, $j \in [k]$.

Construction 3 *Construct the $(n = 2m+2, k = 2m, l = 2^m)$ code over $\mathbb{F}$ by the following $2m$ encoding matrices A_{um+i} and repairing subspaces S_{um+i}, $i \in [m]$ and $u \in \{0,1\}$.*

- *Define the matrix A_i, $i \in [m]$, to have eigenspaces $P_{i,1}, P_{i,3}$ that correspond to eigenvalues $xy, -xy$ respectively.*
- *Define the matrix A_{m+i}, $i \in [m]$ to have eigenspaces $P_{i,0}, P_{i,2}$ that correspond to distinct nonzero eigenvalues λ, μ respectively.*
- *Let the repairing subspace that correspond to the matrix A_{um+i} be $S_{um+i} = P_{i,u}$, $i \in [m]$ and $u \in \{0,1\}$.*

Example 3 *When $m = 1$, we obtain a $(4,2,2)$ code with 2 encoding matrices represented with respect to the standard basis*

$$A_1 = \begin{bmatrix} & x^2 \\ y^2 & \end{bmatrix}, A_2 = \begin{bmatrix} \lambda & \\ & \mu \end{bmatrix}, \tag{21}$$

and repairing subspaces

$$S_1 = P_{1,0} = \text{span}(1,0), S_2 = P_{1,1} = \text{span}(y,x).$$

When $m = 2$, the subspaces $P_{i,u}, S_i$ are listed in Figure 8. The encoding matrices represented with respect to the standard basis are

$$A_1 = \begin{bmatrix} & & x^2 & \\ & & & x^2 \\ y^2 & & & \\ & y^2 & & \end{bmatrix}, A_2 = \begin{bmatrix} & x^2 & & \\ y^2 & & & \\ & & & x^2 \\ & & y^2 & \end{bmatrix},$$

$$A_3 = \begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \mu & \\ & & & \mu \end{bmatrix}, A_4 = \begin{bmatrix} \lambda & & & \\ & \mu & & \\ & & \lambda & \\ & & & \mu \end{bmatrix}.$$

The repairing subspaces are

$$\begin{aligned}
S_1 = P_{1,0} &= \text{span}\begin{bmatrix} 1 & & 0 & 0 \\ & 1 & 0 & 0 \end{bmatrix},\\
S_2 = P_{2,0} &= \text{span}\begin{bmatrix} 1 & 0 & & \\ & & 1 & 0 \end{bmatrix},\\
S_3 = P_{1,1} &= \text{span}\begin{bmatrix} y & & x & \\ & y & & x \end{bmatrix},\\
S_4 = P_{2,1} &= \text{span}\begin{bmatrix} y & x & & \\ & & y & x \end{bmatrix}.
\end{aligned}$$

In both examples it is not difficult to check that the subspace property is satisfied, hence the code has optimal repair. And since the encoding matrices are generalized permutation matrices, the code has optimal update.

The next lemma is similar to Lemma 2 and it will be used later to prove the optimal repair property.

Lemma 8 *The intersection of the subspaces $P_{i,u}$ defined in (20) satisfy for $i \neq j \in [m], u \in \{0,1\}$:*

$$\begin{aligned}
P_{i,u} &= (P_{i,u} \cap P_{j,0}) \oplus (P_{i,u} \cap P_{j,2}) \text{ and}\\
P_{i,u} &= (P_{i,u} \cap P_{j,1}) \oplus (P_{i,u} \cap P_{j,3}).
\end{aligned} \tag{22}$$

Proof: It is easy to verify that $\mathbb{F}^l = P_{j,0} \oplus P_{j,2}$ and that each of the following four subspaces is of dimension $l/4$.

$$\begin{aligned}
P_{i,0} \cap P_{j,0} =& \text{span}(e_a : a \in [0, l-1], a_i = 0, a_j = 0),\\
P_{i,0} \cap P_{j,2} =& \text{span}(e_a : a \in [0, l-1], a_i = 0, a_j = 1),\\
P_{i,1} \cap P_{j,0} =& \text{span}(ye_a + xe_b : a_j = b_j = 0, a_i = 0, b_i = 1,\\
&\text{and for all } s \neq i, j, a_s = b_s),\\
P_{i,1} \cap P_{j,2} =& \text{span}(ye_a + xe_b : a_j = b_j = 1, a_i = 0, b_i = 1,\\
&\text{and for all } s \neq i, j, a_s = b_s).
\end{aligned}$$

Note that the subspaces $P_{j,0}$ and $P_{j,2}$ intersect trivially, and therefore the $(P_{i,u} \cap P_{j,0}) \oplus (P_{i,u} \cap P_{j,2})$ is indeed a direct sum of vector spaces. Furthermore, it is a direct sum of two subspaces of $P_{i,u}$ of dimension $l/4$, hence by dimensionality it equals to $P_{i,u}$ as needed. The proof of $P_{i,u} = (P_{i,u} \cap P_{j,1}) \oplus (P_{i,u} \cap P_{j,3})$ follows in a similar way. ∎

For example, from Figure 8 we see that

$$\begin{aligned}
P_{1,1} \cap P_{2,1} &= \text{span}(y^2 e_0 + xye_1 + xye_2 + x^2 e_3), \text{ and}\\
P_{1,1} \cap P_{2,3} &= \text{span}(-y^2 e_0 + xye_1 - xye_2 + x^2 e_3).
\end{aligned}$$

Therefore, their direct sum is

$$\begin{aligned}
&(P_{1,1} \cap P_{2,1}) \oplus (P_{1,1} \cap P_{2,3})\\
=&\text{span}(y^2 e_0 + xye_2, xye_1 + x^2 e_3)\\
=&P_{1,1}.
\end{aligned}$$

Theorem 9 *Construction 3 has optimal repair and optimal update.*

Proof: It is easy to check that the encoding matrices are all generalized permutation matrices. In particular, A_{m+i}, $i \in [m]$, has diaganoal entries that equal to λ if the row index $a \in [0, l-1]$ has $a_i = 0$, and equal to λ if $a_i = 1$. And A_i,

	$P_{1,0}$ $= S_1$	$P_{1,1}$ $= S_3$	$P_{1,2}$	$P_{1,3}$	$P_{2,0}$ $= S_2$	$P_{2,1}$ $= S_4$	$P_{2,2}$	$P_{2,3}$
basis for	e_0	$ye_0 + xe_2$	e_2	$-ye_0 + xe_2$	e_0	$ye_0 + xe_1$	e_1	$-ye_0 + xe_1$
the subspace	e_1	$ye_1 + xe_3$	e_3	$-ye_1 + xe_3$	e_2	$ye_2 + xe_3$	e_3	$-ye_2 + xe_3$

Figure 8. Subspaces $P_{i,u}$ used to construct optimal update code with 2 parities, $m = 2$. The repairing subspace $S_{um+i} = P_{i,u}$, $u \in \{0,1\}$, are also labeled.

$i \in [m]$, has nonzero entries if the (row, column) index pair is (a,b) with $a_i = 0, b_i = 1, a_j = b_j$ for all $j \neq i$. So the code has optimal update. We need to show that the subspace property (10), (11) holds. Fix $i, j \in [m]$ and $u, t \in \{0,1\}$ such that $um + i \neq tm + j$. First we show that (10) holds:

$$S_{um+i}A_{tm+j} = S_{um+i}, \text{ i.e., } P_{i,u}A_{tm+j} = P_{i,u}. \tag{23}$$

- Case $i \neq j$: Recall that the eigenspaces of A_{tm+j} are $P_{j,1}, P_{j,3}$ or $P_{j,0}, P_{j,2}$ depending on the value of t. In either case, by Lemma 8, for $u \in \{0,1\}$,

$$P_{i,u} = (P_{i,u} \cap P_{j,0}) \oplus (P_{i,u} \cap P_{j,2}) \text{ and}$$
$$P_{i,u} = (P_{i,u} \cap P_{j,1}) \oplus (P_{i,u} \cap P_{j,3}).$$

  In other words, $P_{i,u}$ is a direct sum of two invariant subspaces of A_{tm+j}. We can proceed the proof the same as in Theorem 3 equation (17).
- Case $i = j$, and $u \neq t$: In this case (23) follows since $P_{i,u}$ is an eigenspace of A_{tm+i}.

Next we show that (11) holds, i.e.,

$$S_{um+i}A_{um+i} + S_{um+i} = \mathbb{F}^l.$$

Assume that $u = 0$, and we will show that for $i \in [m]$, the transformation A_i maps the subspace $S_i = P_{i,0}$ to the subspace $P_{i,2}$, namely,

$$S_iA_i = P_{i,2}. \tag{24}$$

Then

$$S_iA_i + S_i = P_{i,2} + P_{i,0} = \mathbb{F}^l$$

and the result will follow. let $e_a \in P_{i,0}$ and b be the integer that is identical to a except the i-th digit. Then for $i \in [m]$,

$$\begin{aligned} e_aA_i &= \frac{1}{2y}[(ye_a + xe_b) - (-ye_a + xe_b)]A_i \\ &= \frac{xy}{2y}(ye_a + xe_b) - \frac{-xy}{2y}(-ye_a + xe_b) \\ &= x^2e_b \in P_{i,2}. \end{aligned}$$

Here we used the fact that 2 is not the characteristic of the field, and y is nonzero. Since A_i is chosen to be of full rank, the dimension of S_iA_i equals the dimension of S_i, and therefore (24) is satisfied. When $u = 1$ the result similarly follows. ∎

Using Theorem 4 the code can be made to satisfy the reconstruction property over a large enough finite field. Moreover the update complexity does not change when the encoding matrices are modified using Theorem 4. To summarize the result of this section, we gave an optimal update MSR construction that doubled the number of systematic nodes compared to the bound in [18]. The reason for the violation of the bound is by using non-diagonal encoding matrices.

V. Codes with Small Access Ratio

Repairing a failed node is a computationally heavy task that requires a large amount of the system's resources. One way to reduce the repair complexity is by reducing the amount of sub-symbols needed to be accessed and read during the repair process. This parameter is quantified by the *access ratio* of the system. In this section we use explicit linear transformations performed on the code in Construction 1 that yields a code with a lower access ratio, while maintaining the reconstruction and the optimal repair properties. Unlike in previous sections, the construction in this section uses different repairing subspaces $S_{i,j}$ for distinct helper nodes j when repairing node i.

Given an (n,k,l) MSR code $\mathcal{C}$ (and a repairing strategy), let $R(i,j)$ denote the number of sub-symbols accessed in helper node j during the repair of systematic node i. The *access ratio* of $\mathcal{C}$ is defined as

$$R = \frac{\sum_{i=1}^{k}\sum_{j=1,j\neq i}^{n} R(i,j)}{k(n-1)l}.$$

Recall that l is the number of sub-symbols stored in each node, then $(n-1)l$ is the amount of surviving sub-symbols in the system in the event of one node erasure, hence R is the average fraction of the sub-symbols in the system being accessed during a repair process. Since the amount of accessed sub-symbols must be no less than that of the transmitted sub-symbols, we know $R \geq 1/r$ for an MSR code. We say the pair of erased node i and helper node j has *optimal access repair* if $R(i,j)/l = 1/r$. Similarly, we say the erased node i has *optimal access repair* if

$$\frac{\sum_{j=1,j\neq i}^{n} R(i,j)}{(n-1)l} = \frac{1}{r}.$$

The **optimal access condition** for the node pair (i,j) is: The repairing subspace $S_{i,j}$ equals the row span of an $l \times l/r$ matrix with only l/r nonzero columns.

The $((r+1)m + r, (r+1)m, r^m)$ MSR code constructed in Construction 1 has $(r+1)m$ systematic nodes, where rm of them have optimal access repair, i.e., only l/r sub-symbols are accessed from each node during the repair process. Thus, the total access cost for repairing these nodes equals $rm \cdot (n-1)l/r$ sub-symbols. However, repairing any of the remaining m systematic nodes, one has to access *all* the surviving sub-symbols in the system. In this case, although the repair bandwidth is optimal, in order to generate the transmitted data one has to access the entire information in the helper node. Repairing these nodes costs accessing $m \cdot (n-1)l$ sub-symbols, and the access ratio of the code is

$$R = \frac{rm \cdot (n-1)l/r + m \cdot (n-1)l}{(r+1)m \cdot (n-1)l} = \frac{2}{r+1}. \tag{25}$$

This value of the access ratio $R = 2/(r+1)$ is our baseline. Next we describe how to apply linear transformation on the encoding matrices, such that R is reduced, while the optimal repair and reconstruction properties are maintained.

Let $A = (A_{i,j})_{i\in[r],j\in[k]}$ be the coding matrix of a $(k+r,k,l)$ MSR code, with repairing subspaces $S_i, i = 1,...,k$. We will apply a linear transformation on the code by multiplying on the right the coding matrix A by a block diagonal matrix B, to get the new coding matrix C as follows,

$$\begin{aligned} C &= \begin{bmatrix} C_{1,1} & \cdots & C_{1,k} \\ \vdots & \ddots & \vdots \\ C_{r,1} & \cdots & C_{r,k} \end{bmatrix} \\ &= AB \\ &= \begin{bmatrix} A_{1,1} & \cdots & A_{1,k} \\ \vdots & \ddots & \vdots \\ A_{r,1} & \cdots & A_{r,k} \end{bmatrix} \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_k \end{bmatrix}. \end{aligned}$$

Namely, for $i \in [r], j \in [k]$

$$C_{i,j} = A_{i,j}B_j, \tag{26}$$

where B_j is an invertible matrix of size $l \times l$. After applying the linear transformation B on the coding matrix, the repairing subspaces should be changed accordingly. Recall that $S_{i,j}$ denotes the repairing subspace for surviving node j during the repair of node i. Define the new repairing subspaces for the code defined by the coding matrix C as follows:

$$S_{i,j} = \begin{cases} S_iB_j, & j \in [k], \\ S_i, & j \in [k+1,k+r]. \end{cases} \tag{27}$$

Notice that compared to the original code, the repairing subspaces are changed only for the systematic nodes. Next we show that the optimal repair and reconstruction properties and kept under the above transformation.

Theorem 10 *Consider the linear transformation* (26), (27) *applied to an MSR code satisfying* (6)(7)*, then the resulting code with encoding matrices $C_{i,j}$ and repairing subspaces $S_{i,j}$ is also an MSR code.*

Proof: Given the coding matrix A of an MSR code, by the subspace property (6) we have for any distinct $i, j \in [k]$, and $t \in [r]$,

$$S_i = S_iA_{t,j}.$$

Therefore,

$$S_{i,j} = S_iB_j = S_iA_{t,j}B_j = S_{i,k+t}C_{t,j},$$

and (3) is satisfied. Moreover, by (7) the sum of subspaces satisfies

$$\sum_{t=1}^{r} S_iA_{t,i} = \mathbb{F}^l.$$

Since the B_i's are of full rank,

$$\sum_{t=1}^{r} S_{i,k+t}C_{t,i} = \sum_{t=1}^{r} S_iA_{t,i}B_i = (\sum_{t=1}^{r} S_iA_{t,i})B_i = \mathbb{F}^l.$$

Therefore (4) is satisfied, and the equivalent code with coding matrix C has optimal repair. Since the code with coding matrix A satisfies the reconstruction property, every $t \times t$ block sub-matrix of A is invertible, for $t \in [r]$. Consider any such invertible sub-matrix:

$$\begin{pmatrix} A_{a_1,b_1} & \cdots & A_{a_1,b_t} \\ \vdots & \ddots & \vdots \\ A_{a_1,b_t} & \cdots & A_{a_t,b_t} \end{pmatrix},$$

for t indices $a_i \in [r]$ and $b_i \in [k]$, $i \in [t]$. Then the corresponding sub-matrix of C is

$$\begin{pmatrix} A_{a_1,b_1} & \cdots & A_{a_1,b_t} \\ \vdots & \ddots & \vdots \\ A_{a_1,b_t} & \cdots & A_{a_t,b_t} \end{pmatrix} \begin{pmatrix} B_{b_1} & & \\ & \ddots & \\ & & B_{b_t} \end{pmatrix},$$

which is also invertible. Therefore, the code with coding matrix C also satisfies the reconstruction property. ■

In order to reduce the access ratio of the transformed code compared to the baseline (25), we need to find linear transformations B_j such that the optimal access condition is satisfied by $S_{i,j} = S_iB_j$ for many pairs (i,j), $i \in [k], j \in [n]$. Consider the code in Construction 1, we next construct B_j from the eigenspaces of the encoding matrices, and then show that the access ratio is reduced.

Let V_j be the $l \times l$ matrix whose rows form a basis of left eigenvectors of the encoding matrix A_j and call it the *eigenspace matrix*. Namely, $V_jA = \Lambda V_j$, and Λ is the diagonal matrix:

$$\begin{pmatrix} \lambda_{j,0}I & & \\ & \ddots & \\ & & \lambda_{j,r-1}I \end{pmatrix}.$$

Here I is the identity matrix of size $l/r \times l/r$. When $j = vm + y$, for $v \in [0,r], y \in [m]$, we have

$$V_j = \begin{pmatrix} \hat{P}_{y,0} \\ \vdots \\ \hat{P}_{y,v-1} \\ \hat{P}_{y,v+1} \\ \vdots \\ \hat{P}_{y,r} \end{pmatrix},$$

where the rows of the matrix $\hat{P}_{y,u}$ are the vectors that span the subspace $P_{y,u}$ defined in (12), $u \neq v$. For example, for the code in Figure 3 if $j = 1$, using standard basis $\{e_0, e_1, e_2, e_3\}$ we can write

$$V_1 = \begin{pmatrix} e_2 \\ e_3 \\ e_0 + e_2 \\ e_1 + e_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Finally, define the transformation matrix

$$B_j = V_j^{-1}. \tag{28}$$

Construction 4 *Consider the coding matrix A and the repairing subspaces S_i, $i \in [k]$ defined in Construction* 1*. Construct an $((r+1)m + r, (r+1)m, r^m)$ MSR code according to* (26)*,* (27)*, and* (28)*, with coding matrix C and repairing subspaces $S_{i,j}$, $i \in [k], j \in [n]$.*

Theorem 11 *The access ratio of Construction* 4 *equals*

$$R = \frac{2}{r+1} - \frac{r-1}{(n-1)(r+1)}.$$

Proof: Suppose node $i = um + x$ was erased. Fix a helper systematic node $j = vm + y$, where $u, v \in [0, r]$ and $x, y \in [m]$. Recall that $P_{y,s}$ for $s \in [0, r]\backslash v$ are the eigenspaces of the encoding matrix A_j.

By (27) for $j \in [k]\backslash\{i\}$ we use the repairing subspace:

$$S_{i,j} = S_i B_j = S_i V_j^{-1}.$$

Here $S_i = P_{x,u}$ as in Construction 1, and B_j is defined in (28). We will show that in a lot of cases S_i can be rewritten as the span of the product of a matrix M and the eigenspace matrix V_j:

$$S_i = \text{span}(MV_j), \tag{29}$$

where M is of size $l/r \times l$ and contains only l/r nonzero columns. This will lead to

$$S_{i,j} = \text{span}(MV_j)V_j^{-1} = \text{span}(MV_jV_j^{-1}) = \text{span}(M)$$

and therefore the code will have optimal access for the node pair (i, j).

- Case $x = y$, $u \neq v$. In this case, $S_i = P_{x,u}$ is one of the eigenspaces in V_j. We can choose $M = [0, \ldots, 0, I, 0, \ldots, 0]$, where 0 and I are of size $l/r \times l/r$, and I is in the u-th block. Thus (29) is satisfied.
- Case $x \neq y$, $u \neq r$. We have observed in Lemma 2 that the subspaces satisfy $P_{x,u} = \oplus_{u' \neq v}(P_{x,u} \cap P_{y,u'})$, where $P_{y,u'}, u' \neq v$ are all the eigenspaces of A_j. Moreover, each $P_{x,u} \cap P_{y,u'}$ only contains linear combinations of l/r^2 vectors in $P_{y,u'}$. Hence (29) holds.
- Case $x \neq y$, $u = r$. We need to access all remaining sub-symbols.

Recall the code has $k = (r+1)m$. From above we see that for each systematic node j as a helper node, it has optimal access for $r + (m-1)r = mr$ erased nodes (the first two cases), and accesses all sub-symbols for $m-1$ erased nodes (the last case).

For each parity node j as a helper node, the repairing subspace is still S_i. By remark 2 in section III-C it has optimal access for rm erased nodes ($i \in [rm]$), and accesses all sub-symbols for m erased nodes ($i \in [rm+1, (r+1)m]$). Therefore, the access ratio is

$$\begin{aligned} R =& \frac{k(rm\frac{l}{r} + (m-1)l) + r(rm\frac{l}{r} + ml)}{k(n-1)l} \\ =& \frac{2}{r+1} - \frac{r-1}{(n-1)(r+1)}. \end{aligned}$$

Hence the proof is completed. ∎

We note here that this transformation lowers the access ratio compared to the original code (25), but in the mean time increases the average updates for each systematic sub-symbol. According to different system requirements, one can choose one code over the other.

The transformation (26)(27) provides a general method to tune the access ratio. Given *any* MSR code, one can define such transformations and manipulate the encoding matrices to lower the access ratio.

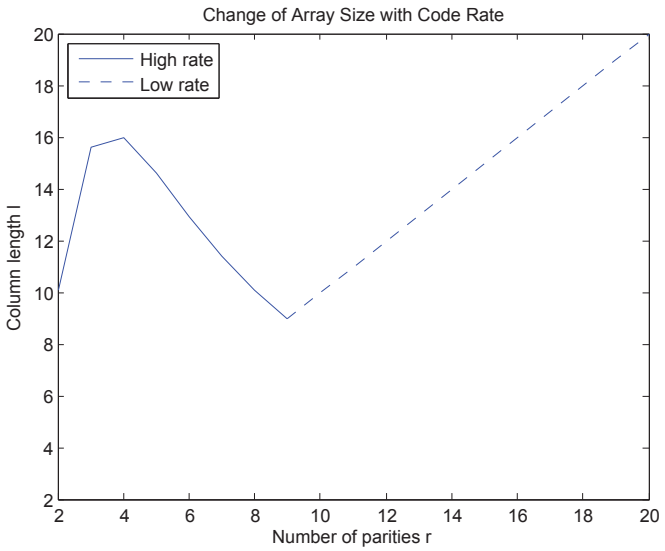

Figure 9. Change of array size with code rate. $k = 10$. For high code rate or $r \leq 9$, the column length is shown in the solid line. For low code rate or $r \geq 9$, the column length is shown in the dashed line.

VI. Concluding Remarks

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 the longest known high-rate MSR codes. The codes were constructed using eigenspaces of the encoding 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.

Next we make some observations on the code paraemters, and then point out open problems for MSR codes. 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 number of systematic nodes 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 and therefore the different codes rate, one can obtain different asymptotic characteristics of the number of systematic nodes.

For $n \geq 2k$ or $k \leq r$ (low code rate), constructions in [13], [16] give the column length $l = r$. With some modifications, this column length is feasible for all $k \leq r+1$. In our construction (high code rate), the column length is $l = r^{\frac{k}{r+1}}$. Fix the value of k, we can draw the graph of the column length with respect to the number of parities. Even though we need integer values for k, r, l, this graph still shows the trend of the code parameters. For example, this relationship is shown in Figure 9 for $k = 10$. These two regimes coincide when $r = k-1 = 9$. Actually, we can see that these two constructions are identical for $r = k-1$. Note that our construction only considers the repair of systematic nodes, so is only practical when $k >> r+1$. It is interesting to investigate the actual shape of this curve, and to understand for fixed k how the column length l changes with the number of parities r.

Moreover, it is still an open problem what the largest k

is given the column length l for an optimal repair code. Moreover, the bound of the finite field size used for the codes may not be tight. Unlike the constructions in this paper, the field size may be reduced when we assume that the encoding matrices do not have eigenvalues or eigenvectors, namely, they are not diagonalizable.

In this paper, we addressed the repair of systematic nodes only. See [19] for a construction of codes that optimally repair systematic and parity codes. In that work, the encoding matrices are constructed directly instead of from their eigenspaces. Regardless of this difference, the subspace property need to be satisfied to ensure optimal repair. It is an interesting problem to construct codes with large k for a fixed column length l, such that they repair any node optimally.

At last, one possible application of the codes is to store hot/cold data. We notice that in our construction, some of the nodes have lower access ratio than others during repair. Since hot data is more commonly requested, we can put the hot data in the low-access nodes, and cold data in the others. More generally, appropriate measurement of repair cost should be defined, and outer bounds and code constructions can be considered for hot/cold data storage.

REFERENCES

[1] G. K. Agarwal, B. Sasidharan, and P. Vijay Kumar, "An alternate construction of an access-optimal regenerating code with optimal sub-packetization level," in *Communications (NCC), 2015 Twenty First National Conference on*. IEEE, 2015, pp. 1–6.

[2] N. Alon and M. Tarsi, "Combinatorial nullstellensatz," *Combinatorics Probability and Computing*, vol. 8, no. 1, pp. 7–30, 1999.

[3] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *Computers, IEEE Transactions on*, vol. 44, no. 2, pp. 192–202, 1995.

[4] V. R. Cadambe, C. Huang, and J. Li, "Permutation code: Optimal exact-repair of a single failed node in MDS code based distributed storage systems," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 1225–1229.

[5] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra, "Polynomial length MDS codes with optimal repair in distributed storage," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*. IEEE, 2011, pp. 1850–1854.

[6] 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," *Information Theory, IEEE Transactions on*, vol. 59, no. 5, pp. 2974–2987, 2013.

[7] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," in *FAST-2004: 3rd Usenix Conference on File and Storage Technologies*, 2004.

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

[9] S. Goparaju, I. Tamo, and R. Calderbank, "An improved sub-packetization bound for minimum storage regenerating codes," *Information Theory, IEEE Transactions on*, vol. 60, no. 5, pp. 2770–2779, 2014.

[10] J. Li, X. Tang, and U. Parampalli, "A framework of constructions of minimal storage regenerating codes with the optimal access/update property," *Information Theory, IEEE Transactions on*, vol. 61, no. 4, pp. 1920–1932, 2015.

[11] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through hadamard designs," *Information Theory, IEEE Transactions on*, vol. 59, no. 5, pp. 3021–3037, 2013.

[12] J. S. Plank, "The raid-6 liber8tion code," *International Journal of High Performance Computing Applications*, 2009.

[13] 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," *Information Theory, IEEE Transactions on*, vol. 57, no. 8, pp. 5227–5239, 2011.

[14] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "A high-rate MSR code with polynomial sub-packetization level," in *Information Theory (ISIT), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 2051–2055.

[15] N. B. Shah, K. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *Information Theory, IEEE Transactions on*, vol. 58, no. 4, pp. 2134–2158, 2012.

[16] C. Suh and K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *Information Theory, IEEE Transactions on*, vol. 57, no. 3, pp. 1425–1442, 2011.

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

[18] ——, "Access versus bandwidth in codes for storage," *Information Theory, IEEE Transactions on*, vol. 60, no. 4, pp. 2028–2037, 2014.

[19] Z. Wang, I. Tamo, and J. Bruck, "On codes for optimal rebuilding access," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. IEEE, 2011, pp. 1374–1381.

[20] ——, "Long MDS codes for optimal repair bandwidth," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 1182–1186.

[21] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. IEEE, 2009, pp. 2276–2280.

[22] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Allerton Conference on Control, Computing, and Communication*. Citeseer, 2007, pp. 1–5.

[23] L. Xu, V. Bohossian, J. Bruck, and D. G. Wagner, "Low-density MDS codes and factors of complete graphs," *Information Theory, IEEE Transactions on*, vol. 45, no. 6, pp. 1817–1826, 1999.

[24] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *Information Theory, IEEE Transactions on*, vol. 45, no. 1, pp. 272–276, 1999.

Zhiying Wang received the B.Sc. degree in Information Electronics and Engineering from Tsinghua University in 2007, M. Sc. and Ph.D degrees in Electrical Engineering from California Institute of Technology in 2009 and 2013, respectively. She was a postdoctoral fellow in Department of Electrical Engineering, Stanford University. She is currently Assistant Professor at Center for Pervasive Communications and Computing, Univeristy of California, Irvine. Dr. Wang received NSF Center for Science of Information (CSoI) Postdoctoral Research Fellow, 2013. She was the recipient of IEEE Communication Society Data Storage Best Paper Award for 2013. Her research focuses on information theory, coding theory, with an emphasis on coding for storage devices and systems and compression for genomic information.

Itzhak Tamo was born in Israel in 1981. He received a B.A. in Mathematics and a B.Sc. in Electrical Engineering in 2008, and a Ph.D. in Electrical Engineering in 2012, all from Ben-Gurion University, Israel. During 2012-2014 he was a postdoctoral researcher at the Institute for Systems Research, University of Maryland, College Park. Since 2015 he has been a senior lecturer in the Electrical Engineering Department, Tel Aviv University, Israel. Itzhak Tamo received the 2015 IEEE Information Theory Society Paper Award and the IEEE Communication Society Data Storage Technical Committee 2013 Best Paper Award. His research interests include storage systems and devices, coding, information theory, and combinatorics.

Jehoshua Bruck (S'86-M'89-SM'93-F'01) is the Gordon and Betty Moore Professor of computation and neural systems and electrical engineering at the California Institute of Technology (Caltech). His current research interests include information theory and systems and the theory of computation in nature.

Dr. Bruck received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University, in 1989.

His industrial and entrepreneurial experiences include working with IBM Research where he participated in the design and implementation of the first IBM parallel computer; cofounding and serving as Chairman of Rainfinity (acquired in 2005 by EMC), a spin-off company from Caltech that created the first virtualization solution for Network Attached Storage; as well as cofounding and serving as Chairman of XtremIO (acquired in 2012 by EMC), a start-up company that created the first scalable all-flash enterprise storage system.

Dr. Bruck is a recipient of the Feynman Prize for Excellence in Teaching, the Sloan Research Fellowship, the National Science Foundation Young Investigator Award, the IBM Outstanding Innovation Award and the IBM Outstanding Technical Achievement Award.