Codes for Distributed Storage

Vinayak Ramkumar[1], S. B. Balaji[2], Birenjith Sasidharan[3], Myna Vajha[4], M. Nikhil Krishnan[5] and P. Vijay Kumar[6]

[1]*Indian Institute of Science, Bengaluru, India; vinram93@gmail.com*
[2]*Qualcomm, Bengaluru, India; balaji.profess@gmail.com*
[3] *Govt. Engineering College, Barton Hill, Trivandrum, India; birenjith@gmail.com*
[4]*Qualcomm, Bengaluru, India; mynaramana@gmail.com*
[5]*International Institute of Information Technology Bangalore, India; nikhilkrishnan.m@gmail.com*
[6]*Indian Institute of Science, Bengaluru, India; pvk1729@gmail.com*

ABSTRACT

In distributed data storage, information pertaining to a given data file is stored across multiple storage units or nodes in redundant fashion to protect against the principal concern, namely, the possibility of data loss arising from the failure of individual nodes. The simplest form of such protection is replication. The explosive growth in the amount of data generated on a daily basis brought up a second major concern, namely minimization of the overhead associated with such redundant storage. This concern led to the adoption by the storage industry of erasure-recovery codes such as Reed-Solomon (RS) codes and more generally, maximum distance separable codes, as these codes offer the lowest-possible storage overhead for a given level of reliability.

In the setting of a large data center, where the amount of stored data can run into several exabytes, a third concern

Vinayak Ramkumar, S. B. Balaji, Birenjith Sasidharan, Myna Vajha, M. Nikhil Krishnan and P. Vijay Kumar (2022), "Codes for Distributed Storage", Foundations and Trends® in Communications and Information Theory: Vol. 19, No. 4, pp 547–813.
DOI: 10.1561/0100000115.

arises, namely the need for efficient recovery from a commonplace occurrence, the failure of a single storage unit. One measure of efficiency in node repair is how small one can make the amount of data download needed to repair a failed unit, termed the repair bandwidth. This was the subject of the seminal paper by Dimakis *et al.* [50] in which an entirely new class of codes called regenerating codes was introduced, that within a certain repair framework, had the minimum-possible repair bandwidth. A second measure relates to the number of helper nodes contacted for node repair, termed the repair degree. A low repair degree is desirable as this means that a smaller number of nodes are impacted by the failure of a given node. The landmark paper by Gopalan *et al.* [72] focuses on this second measure, leading to the development of the theory of locally recoverable codes. The two events also led to the creation of a third class of codes known as locally regenerating codes, where the aim is to simultaneously achieve reduced repair bandwidth and low repair degree. Research in a different direction led researchers to take a fresh look at the challenge of efficient RS-code repair, and led to the identification of improved repair schemes for RS codes that have significantly reduced repair bandwidth.

This monograph introduces the reader to these different approaches towards efficient node repair and presents many of the fundamental bounds and code constructions that have since emerged. Several open problems are identified, and many of the sections have a notes subsection at the end that provides additional background.

The version of record is available at: http://dx.doi.org/10.1561/0100000115

1

Introduction

Given the failure-prone nature of a storage device, reliability against data loss has always been of paramount importance in the storage industry. In the early days, this was achieved through simple replication of data, for example, triple replication was a commonplace selection within the Hadoop distributed file system (HDFS). However, the explosive growth in the amount of data stored over the past couple of decades encouraged the industry to look for other means of ensuring reliability and having less storage overhead. Here, the class of maximum distance separable (MDS) codes are a natural choice as they incur the least amount of storage overhead for a given level of protection, measured in terms of the maximum number of node failures that can be tolerated.

1.1 Conventional Repair of an MDS Code

Many of the schemes employed in redundant array of independent disks (RAID) technology make use of MDS codes. An $[n, k]$ MDS code is a block code of length n and dimension k over a suitably-defined finite field. To store data using an $[n, k]$ MDS code, the data file is first partitioned into k equal-sized fragments, that are then stored on k distinct storage units. An additional set of $r = (n - k)$ fragments

of redundant data are then created and stored on a further set of r storage units in such a manner that the contents of any k out of the n storage units suffice to recover the data. In this way, the contents of a file are efficiently stored in redundant fashion, across a set of n storage units. For example, RAID 6 makes use of a $[5, 3]$ MDS code. Other examples of MDS codes that appear in the erasure coded-version HDFS-EC of HDFS are a $[9, 6]$ MDS code as well as a $[14, 10]$ code, the latter employed by Facebook. Throughout the monograph, we will alternately refer to a storage unit as a node.

Today's data centers store massive amounts of information, amounts that can run into several exabytes, i.e., 10^{18} bytes. While protection against data loss and maintaining low values of storage overhead continue to be of primary importance, a third concern has recently surfaced. This has to do with the efficiency with which a failed storage unit can be repaired. We will view the repair process as one in which a new storage unit, which we will term as the replacement node, is brought in as a substitute for a failed storage unit. The replacement node then draws from the partial or entire contents of all or a subset of the remaining $(n-1)$ nodes, and uses the data so received to replicate the contents of the original failed node.

As is well known, an $[n, k, d_{\min}]$ code $\mathcal{C}$ is protected against data loss if the number of node failures does not exceed $(d_{\min}-1)$. For a given value of $(d_{\min}-1)$, MDS codes in general, and Reed-Solomon (RS) codes in particular, have the least possible value of storage overhead given by $\frac{n}{k} = \frac{n}{n-d_{\min}+1}$. This follows as the minimum distance $d_{\min}$ of an MDS code satisfies $d_{\min} = (n-k+1)$, which by the Singleton bound [156] is the largest value possible. In coding-theoretic terms, the problem of node repair is equivalent to recovery from erasure of a single code symbol. The most obvious approach is to invoke a parity-check (p-c) equation involving the erased code symbol. Let

$$\underline{c} = (c_1 \; c_2 \; \ldots \; c_n)$$

be a code word and let us assume without loss of generality, erasure of the first code symbol c_1. Any p-c equation involving c_1 of the form

$$\sum_{i=1}^{n} h_i c_i = 0, \qquad h_1 \neq 0,$$

is associated to a codeword

$$\underline{h} \;=\; (h_1\ h_2\ \ldots\ h_n)$$

belonging to the $[n, n-k]$ dual code $\mathcal{C}^{\perp}$. In the case of an MDS code $\mathcal{C}$, its dual $\mathcal{C}^{\perp}$ is also an MDS code and hence has parameters $[n, n-k, k+1]$. Thus any codeword $\underline{h}$ in $\mathcal{C}^{\perp}$ has Hamming weight $w_H(\underline{h}) \geq k+1$. Thus if a p-c equation

$$\sum_{i=1}^{n} h_i c_i = 0,$$

is used to recover the code symbol c_1, then we have

$$c_1 = \sum_{i=2}^{n} \Big(\frac{-h_i}{h_1}\Big) c_i, \tag{1.1}$$

with at least k terms of the form $\frac{-h_i}{h_1}$ on the right side being nonzero.

1.2 Regenerating Codes and Locally Recoverable Codes

For the operation of a data center, equation (1.1) has two implications. Firstly, that the replacement of the failed node must necessarily contact k "helper nodes", i.e., nodes that store the code symbols $\{c_i \mid \frac{h_i}{h_1} \neq 0\}$. Secondly, equation (1.1) suggests that each helper node must transfer its entire contents (represented by c_i) for repair of the failed node. The number of helper nodes contacted (at least k in the case of an MDS code) is called the repair degree of the code. The total amount of data downloaded for repair of the failed node is termed the repair bandwidth. In the case of an MDS code, it is clear that the repair bandwidth is at least k times the amount of data stored in the failed node.

This is illustrated below in the case of an $[14, 10]$ MDS code. Assume a data file of size equal to 1 GB. The data file is partitioned into 10 fragments, each of size 100 MB and each data fragment is stored in a different node. Four parity nodes are then created, corresponding to the four parity symbols of the MDS code. The contents of the 14 nodes can be regarded as the layering of 10^8 codewords, each belonging to the $[14, 10]$ MDS code over $\mathbb{F}_{2^8}$. Fig. 1.1 shows repair of a failed node. As

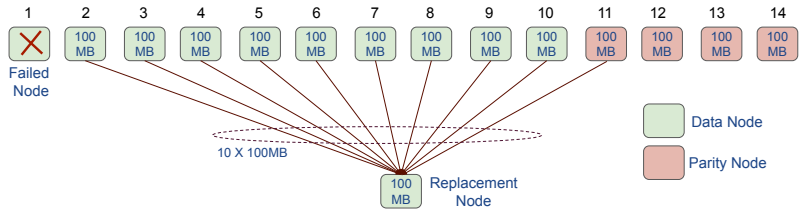

Figure 1.1: Illustrating the repair degree and repair bandwidth involved in the conventional repair of a failed node in a $[14, 10]$ MDS code

can be seen, there are $k = 10$ helper nodes corresponding to nodes 2 through 11 and each helper node passes on the 100 MB of data or parity stored in the respective node, to the repair center. Thus in this case the repair degree equals 10 and the repair bandwidth equals 10×100 MB$=$ 1 GB.

Seminal papers by Dimakis *et al.* [50] and Gopalan *et al.* [72] heralded the theory of two entirely new classes of erasure-recovery codes, termed as regenerating codes (RGCs) and locally recoverable codes (LRCs), that were designed with the express aim of lowering the repair bandwidth and repair degree respectively. The development of the theory of RGCs and LRCs also led to the creation of a class of codes termed as locally regenerating codes by Kamath *et al.* [117] and Rawat et el. [189], where the aim is to simultaneously achieve reduced repair bandwidth and low repair degree. Research in a slightly different direction, pioneered by Shanmugam *et al.* [215] and Guruswami and Wootters [85], led to a re-examination of the repair bandwidth of RS codes and the design of more efficient repair schemes that permitted node repair with reduced repair bandwidth.

As an indication of the kind of impact that research on the topics of RGCs and LRCs has had on the development of coding theory, we note that papers reporting research in this area have received many best paper awards over the years. The list includes [50], [62], [72], [103], [137], [185], [228], [229], [237], [255].

1.3 Overview of the Monograph

This monograph presents an overview of how research on the topic of codes for distributed storage has evolved in a certain direction (see Fig. 1.2 for an overview of topics covered here). There have been several excellent prior surveys on the topic, including those found in [46], [51], [136], [145]. Additionally, concise surveys by the authors of the present monograph can be found in [10], [178].

Given the vast nature of the literature on the topic of codes for distributed storage, we have undoubtedly missed many papers that have made a strong contribution. We apologize in advance to the authors of these papers for the inadvertent omission. Furthermore, as can be seen from the listing of topics in Fig. 1.2, our focus here is only on certain specific approaches to coding for distributed storage.

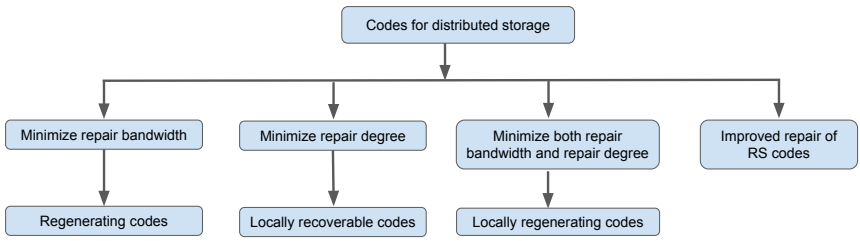

Figure 1.2: An overview of the coverage of codes for distributed storage in this monograph.

MDS Codes Section 2 provides background on MDS, RS codes and a generalization of RS codes known as generalized RS (GRS) codes.

Regenerating Codes The next seven sections deal with RGCs. The definition of an RGC along with a fundamental upper bound on file size is presented in Section 3. The bound reveals that there is a tradeoff between the storage overhead and the repair bandwidth. Sections 4 and 5 present constructions for the two main classes of RGCs, namely minimum bandwidth regenerating (MBR) codes and minimum storage regenerating (MSR) codes, that lie at the two ends of the storage-repair

bandwidth tradeoff. The tradeoff itself is explored in the following section, Section 6. Constructions for RGCs that lie on interior points of the tradeoff are presented in Section 7. The sub-packetization level of an RGC may be regarded as denoting the number of symbols stored per node. An alternate viewpoint is to regard a regenerating code as a code over a vector symbol alphabet of the form $\mathbb{F}_q^{\alpha}$, with α denoting the sub-packetization level. Lower values of sub-packetization are desirable in practice, as a large sub-packetization level, apart from increasing the complexity of implementation, also limits the smallest size of a file that can be stored. Section 8 presents lower bounds on the sub-packetization level of an MSR code.

Several variants of RGCs have been explored in the literature. Piggyback codes, ϵ-MSR codes and the codes of Li-Liu-Tang, are MDS codes that have reduced repair bandwidth and much smaller sub-packetization level. Cooperative RGCs explore the cooperative repair of a set of $t > 1$ failed nodes. Secure RGCs are designed to provide security in the presence of an eavesdropper or an active adversary. Rack-aware RGCs are designed to minimize the amount of cross-rack repair data that is transferred. An erasure-recovery code is said to possess the repair-by-transfer (RBT) property, if it enables repair of a failed node without need for computation at either helper or replacement node. Fractional repetition codes form a class of erasure-recovery codes that possesses the repair-by-transfer property and can be viewed as a generalization of a class of RBT MBR codes. The former codes potentially offer reduced storage overhead at the cost of reduced freedom in the selection of helper nodes. All these variants of RGCs can be found discussed in Section 9.

Locally Recoverable Codes As noted above, the need for repair of a failed node with low degree prompted the creation of LRCs. Section 10 introduces LRCs and presents an upper bound on the rate and minimum distance of an LRC as well as optimal code constructions.

One means of handling the simultaneous failure of several nodes with low repair degree is to make the local codes that are at the core of an LRC more powerful. There are other approaches however, each with its own advantages and disadvantages. The three sections that follow present these other approaches. Availability codes, discussed in

Section 11, represent one such example. This class of codes has the additional feature that in the case of a single erased node, there are multiple, node-disjoint means of recovering from the node failure. This can be a very useful feature to have in practice, particularly as a means of handling cases when there are multiple simultaneous demands for the data contained within a particular node.

Sequential-recovery LRCs place the least stringent conditions on an LRC for the local recovery from multiple erasures, and consequently, have smallest possible storage overhead. These are discussed in Section 12. If an LRC has large block length and small value of repair degree r, and a particular local code is overwhelmed by erasures, the only option is to fall back on the properties of the full-length block code to recover from the erasure pattern, leading to a sharp increase in the repair degree. Codes with hierarchical locality, discussed in Section 13, are designed to address this situation, provide layers of local codes having increasing block length as well as erasure-recovery capability, and permit a more graceful degradation in repair degree with an increasing number of erasures.

Maximally recoverable codes (MRCs), discussed in Section 14, may be regarded as the subclass of LRCs that are as MDS as possible in the sense that every set of k columns of the generator matrix of an MRC is a linearly independent set, unless the locality constraints imposed make it impossible for this to happen. An MRC is maximal in the sense that if an MRC is not able to recover from an erasure pattern, then no other code satisfying the same locality constraints can possibly recover from the same erasure pattern.

Locally Regenerating Codes Section 15 introduces a class of codes in which the local codes are themselves regenerating codes. As a result, these codes simultaneously offer both low repair degree as well as low repair bandwidth.

Improved Repair Schemes for RS Codes The evolution of RGCs and LRCs spurred researchers to take a fresh look at the challenge of efficient RS-code repair and led to the identification of improved repair

schemes for RS codes having significantly reduced repair bandwidth. These developments are described in Section 16.

Codes in Practice The final section, Section 17, discusses the impact that the theoretical developments discussed in this monograph have had in practice.

2

Maximum Distance Separable Codes

In this section, we will provide some background on maximum distance separable (MDS) codes, of which Reed-Solomon (RS) codes are the principal example. MDS codes are widely used in the storage industry, appearing for example in the guise of RAID codes. References to MDS and RS codes can be found scattered throughout the manuscript, as they are often an ingredient to a particular code construction or else are closely related in some manner.

2.1 Reed-Solomon Codes

Let $\mathbb{F}_q$ be a finite field of q elements. Let $\mathbb{F}_q[x]$ denote the set of all polynomials in x over $\mathbb{F}_q$:

$$\mathbb{F}_q[x] \quad = \quad \left\{ \sum_{i=0}^{d} u_i x^i \mid u_i \in \mathbb{F}_q, d \in \{0, 1, 2, \cdots\} \right\}.$$

If $f(x) = \sum_{i=0}^{d} u_i x^i$, with $u_d \neq 0$, then f is said to have degree d and monic if $u_d = 1$. Let $\{\theta_1, \theta_2, \ldots, \theta_n\} \subseteq \mathbb{F}_q$ be a set of n distinct elements and set

$$\mathcal{C}_{\mathrm{RS}} \quad = \quad \{ \ (f(\theta_1), f(\theta_2), \ldots, f(\theta_n)) \ \mid \ f \in \mathbb{F}_q[x], \ \deg(f) \leq k-1 \}.$$

We will refer to a code having the structure of $\mathcal{C}_{\rm RS}$ as an $[n, k]$ Reed-Solomon (RS) code [196]. Thus each codeword $\underline{c}$ in $\mathcal{C}_{\rm RS}$ is of the form

$$(c_1, c_2, \cdots, c_n) \quad = \quad (f(\theta_1), f(\theta_2), \ldots, f(\theta_n))$$

for some polynomial of the form $f(x) = \sum_{i=0}^{k-1} u_i x^i$. We can therefore write:

$$[c_1 \; c_2 \; \ldots \; c_n] \quad = \quad [u_0 \; u_1 \; \ldots \; u_{k-1}]G$$

where the $(k \times n)$ generator matrix G of the RS code $\mathcal{C}_{\rm RS}$ is the Vandermonde matrix given by

$$G \quad = \quad \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \theta_1 & \theta_2 & \ldots & \theta_n \\ \vdots & \vdots & \vdots & \vdots \\ \theta_1^{k-1} & \theta_2^{k-1} & \ldots & \theta_n^{k-1} \end{bmatrix}.$$

From the properties of a Vandermonde matrix, it follows that every $(k \times k)$ sub-matrix of G is non-singular. Thus $\mathcal{C}_{\rm RS}$ is a k-dimensional subspace of $\mathbb{F}_q^n$, i.e., $\mathcal{C}_{\rm RS}$ is an $[n, k]$ linear code. We will use the terminology (n, M) to denote a code $\mathcal{C}$ having block length n and of size $|\mathcal{C}| = M$, that is not necessarily linear. Thus $\mathcal{C}_{\rm RS}$ is simultaneously also an (n, q^k) code.

The minimum distance $d_{\min}$ of a code $\mathcal{C}$ is the minimum Hamming distance between a pair of distinct codewords in $\mathcal{C}$. The minimum weight $w_{\min}$ of a linear code is equal to the minimum Hamming weight of a nonzero codeword in $\mathcal{C}$. It is straightforward to show that in a linear code, we must have $d_{\min} = w_{\min}$. Since a polynomial of degree d can have at most d zeros, it follows that in the case of an RS code $\mathcal{C}_{\rm RS}$, $d_{\min} = w_{\min} \geq (n - k + 1)$. On the other hand, the polynomial

$$f(x) \quad = \quad \prod_{j=1}^{k-1} (x - \theta_j) \tag{2.1}$$

has exactly $(k - 1)$ zeros and it follows from this that $d_{\min} = w_{\min} = (n - k + 1)$ in the case of an $[n, k]$ RS code.

We use $[n, k, d_{\min}]$ to denote an $[n, k]$ linear code having minimum distance $d_{\min}$. Analogously, we will use $(n, M, d_{\min})$ to denote an (n, M) code having minimum distance $d_{\min}$. It follows that the RS code $\mathcal{C}_{\rm RS}$ is

an $[n, k, (n-k+1)]$ linear code over $\mathbb{F}_q$. The Singleton bound below will establish that an RS code $\mathcal{C}_{\text{RS}}$ has the largest possible size among all codes of block length n and $d_{\min} = (n-k+1)$.

2.2 Singleton Bound

Let $\mathcal{C}$ be an (n, M) code over an alphabet $\mathcal{A}$ of size $|\mathcal{A}| = q$. Thus $\mathcal{C} \subseteq \mathcal{A}^n$. Let $\mathcal{C}$ have minimum distance $d_{\min}$. We will now derive a bound on the maximum possible size M of $\mathcal{C}$.

Let A be the $(M \times n)$ matrix whose rows are precisely the M codewords in $\mathcal{C}$. Let B be the $(M \times (n - d_{\min} + 1))$ sub-matrix of A obtained by restricting attention to the last $(n - d_{\min} + 1)$ columns of A. Clearly all the rows of B must be distinct, else, $\mathcal{C}$ will have minimum distance $\leq d_{\min} - 1$, a contradiction. It follows that $M \leq |\mathcal{A}|^{n-d_{\min}+1}$. This upper bound on the size M of an (n, M) code having minimum distance $d_{\min}$ is called the Singleton bound [223].

Theorem 1. (Singleton Bound) The size M of an $(n, M, d_{\min})$ code $\mathcal{C}$ over an alphabet $\mathcal{A}$ must satisfy the upper bound:

$$M \leq |\mathcal{A}|^{n-d_{\min}+1}.$$

Definition 1. Codes achieving the Singleton bound with equality are called maximum distance separable (MDS) codes.

Remark 1. An $[n, k, d_{\min}]$ RS code is an MDS code since $d_{\min} = (n - k + 1)$ and the code size M equals $q^k = q^{n-d_{\min}+1}$.

It follows from the arguments used to establish the Singleton bound, that a codeword belonging to an $(n, M, d_{\min})$ code $\mathcal{C}$ can be uniquely identified given access to any set of $(n - d_{\min} + 1)$ code symbols. In particular, a codeword belonging to a linear $[n, k, d_{\min}]$ MDS code $\mathcal{C}$, can be uniquely identified given any set of k code symbols. It follows from this that if G is the generator matrix of an $[n, k]$ MDS code, then every $(k \times k)$ sub-matrix of G is non-singular. It is straightforward to establish the converse and hence, an $[n, k]$ code is an MDS code iff every $(k \times k)$ sub-matrix of a generator matrix G for the code is nonsingular.

2.2.1 Recovery from Erasures

Let $\mathcal{C}$ be an $[n, k, d_{\min}]$ code that is used for transmission over an erasure channel, i.e., a channel in which a subset of the code symbols transmitted over the channel are erased. Since each codeword in $\mathcal{C}$ is uniquely determined from a subset of $(n - d_{\min} + 1)$ or more code symbols, it follows that the transmitted codeword can be recovered if no more that $(d_{\min} - 1)$ code symbols are erased, i.e., if at least $(n - d_{\min} + 1)$ code symbols remain unerased.
If $\mathcal{C}$ is an RS code of the form:

$$\mathcal{C}_{\text{RS}} \quad = \quad \{(f(\theta_1), f(\theta_2), \ldots, f(\theta_n)) \mid f \in \mathbb{F}_q[x], deg(f) \leq k-1\}$$

and $\{\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_k}\}$ represent a subset of the unerased code symbols of size $k = (n - d_{\min} + 1)$, the remaining code symbols can be explicitly recovered using Lagrange interpolation:

$$f(x) \quad = \quad \sum_{\ell=1}^{k} f(\theta_{i_\ell}) \prod_{j=1, j\neq\ell}^{k} \frac{(x - \theta_{i_j})}{(\theta_{i_\ell} - \theta_{i_j})}.$$

2.3 Generalized Reed-Solomon Codes

We now present a generalization of RS codes under which the dual of a generalized RS (GRS) code is once again a generalized RS code [156]. Let $\mathcal{C}$ be an RS code as above, i.e.,

$$\mathcal{C}_{\text{RS}} \quad = \quad \{(f(\theta_1), \ldots, f(\theta_n)) \quad | \quad f \in \mathbb{F}_q[x], \deg(f) \leq k-1\},$$

where $\{\theta_1, \cdots, \theta_n\}$ are a collection of n distinct elements belonging to the finite field $\mathbb{F}_q$.

We begin with an observation concerning the $((n-1) \times n)$ Vandermonde matrix:

$$P \quad = \quad \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \theta_1 & \theta_2 & \ldots & \theta_n \\ \vdots & \vdots & \vdots & \vdots \\ \theta_1^{n-2} & \theta_2^{n-2} & \ldots & \theta_n^{n-2} \end{bmatrix}.$$

Clearly P has rank $(n-1)$ and every $((n-1) \times (n-1))$ sub-matrix of P is nonsingular. It follows that the right nullspace of P contains a

vector $\underline{u} = [u_1 \cdots u_n]^T \in \mathbb{F}_q^n$ all of whose components are nonzero, i.e., $\underline{u}$ satisfies $P\underline{u} = \underline{0}$ and $u_i \neq 0$, $1 \leq i \leq n$.

Next, let f, g be polynomials over $\mathbb{F}_q$ with $\deg(f) \leq k-1$ and $\deg(g) \leq n-k-1$. Set $h(x) = f(x)g(x)$. Then $\deg(h) \leq n-2$ and we can write

$$h(x) \;=\; \sum_{j=0}^{n-2} h_j x^j.$$

It follows that

$$\sum_{i=1}^{n} u_i h(\theta_i) \;=\; \sum_{i=1}^{n} u_i \sum_{j=0}^{n-2} h_j \theta_i^j \;=\; \sum_{j=0}^{n-2} h_j \left(\sum_{i=1}^{n} u_i \theta_i^j \right) \;=\; 0.$$

$$\Longrightarrow \sum_{i=1}^{n} u_i f(\theta_i) g(\theta_i) \;=\; 0.$$

It follows from this that the dual of an RS code having generator matrix of the form

$$G \;=\; \begin{bmatrix} 1 & \dots & 1 \\ \theta_1 & \dots & \theta_n \\ \vdots & \vdots & \vdots \\ \theta_1^{k-1} & \dots & \theta_n^{k-1} \end{bmatrix}$$

is the block code having generator matrix of the form

$$H \;=\; \begin{bmatrix} 1 & \dots & 1 \\ \theta_1 & \dots & \theta_n \\ \vdots & \vdots & \vdots \\ \theta_1^{n-k-1} & \dots & \theta_n^{n-k-1} \end{bmatrix} \begin{bmatrix} u_1 & & & \\ & u_2 & & \\ & & \ddots & \\ & & & u_n \end{bmatrix}$$

with all $u_i \neq 0$. We will refer to any code having generator matrix G of the form

$$G \;=\; \begin{bmatrix} 1 & \dots & 1 \\ \theta_1 & \dots & \theta_n \\ \vdots & \vdots & \vdots \\ \theta_1^{k-1} & \dots & \theta_n^{k-1} \end{bmatrix} \begin{bmatrix} u_1 & & & \\ & u_2 & & \\ & & \ddots & \\ & & & u_n \end{bmatrix} \text{ with all } u_i \neq 0$$

as a GRS code. Clearly, this code has parameters $[n, k, n-k+1]$ and is hence also an MDS code. This establishes that the dual of an RS code is a GRS code and further that the dual of a GRS code is once again, a GRS code.

2.4 Systematic Encoding

Definition 2. An $[n, k]$ linear code $\mathcal{C}$ is said to be systematic if it possesses a generator matrix G of the form

$$G = [I_k \mid P],$$

where P is a $(k \times (n-k))$ matrix.

When we make reference to a systematic code, it is implicitly understood that the code is encoded using a generator matrix having this form. The advantage of encoding using such a matrix is that the k message symbols are explicitly present within the set of n code symbols and this is a very desirable property in practice. While not every linear code is systematic, there is an 'equivalent' code obtained by rearranging code symbols that is systematic and in this way, the requirement of making a code systematic is easily met.

Let $\mathcal{C}$ be an $[n, k]$ MDS code and let G_0 be a $(k \times n)$ generator matrix for $\mathcal{C}$. Since any k columns of G_0 are linearly independent, the matrix G_0 can be row reduced to yield a second generator matrix G for $\mathcal{C}$ that is of the form $G = [I_k \mid P]$. The $(k \times (n-k))$ matrix P has the interesting and useful property that any $(\ell \times \ell)$ square sub-matrix of P is nonsingular, $1 \leq \ell \leq \min\{k, n-k\}$. This property can be established using elementary row reduction. The converse is also straightforward to establish, namely that if the $(k \times n)$ generator matrix G of a linear code $\mathcal{C}$ is of the form $G = [I_k \mid P]$ where every $(\ell \times \ell)$ sub-matrix of P is non-singular, then $\mathcal{C}$ is an $[n, k]$ MDS code.

2.5 Cauchy MDS Codes

Our goal here is to present an explicit construction of a square $(m \times m)$ matrix A, called the Cauchy matrix, having the property that every

square sub-matrix of A obtained by selecting any ℓ rows and ℓ columns of A is non-singular.

Construction 1. (Cauchy Matrix) Let $\{a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_m\}$ be a set of $2m$ distinct elements belonging to the finite field $\mathbb{F}_q$. Let A be an $(m \times m)$ matrix, called the Cauchy matrix, whose $(i,j)^{th}$ entry A_{ij}, $1 \leq i, j \leq m$, is given by:

$$A_{ij} = \frac{1}{(a_i - b_j)},$$

i.e.,

$$A = \begin{bmatrix} \frac{1}{(a_1-b_1)} & \frac{1}{(a_1-b_2)} & \cdots & \frac{1}{(a_1-b_m)} \\ \frac{1}{(a_2-b_1)} & \frac{1}{(a_2-b_2)} & \cdots & \frac{1}{(a_2-b_m)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{(a_m-b_1)} & \frac{1}{(a_m-b_2)} & \cdots & \frac{1}{(a_m-b_m)} \end{bmatrix}. \tag{2.2}$$

We will show that the Cauchy matrix A is non-singular by identifying an inverse. It follows from the structure of A that this will also establish that every square sub-matrix of A is also non-singular.

The relevance of Cauchy matrices is that if we choose a generator matrix G for an $[n,k]$ code $\mathcal{C}$ to be of the form $G = [I_k \mid P]$ where P is a $(k \times (n-k))$ Cauchy matrix

$$P = \begin{bmatrix} \frac{1}{(a_1-b_1)} & \frac{1}{(a_1-b_2)} & \cdots & \frac{1}{(a_1-b_{n-k})} \\ \frac{1}{(a_2-b_1)} & \frac{1}{(a_2-b_2)} & \cdots & \frac{1}{(a_2-b_{n-k})} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{(a_k-b_1)} & \frac{1}{(a_k-b_2)} & \cdots & \frac{1}{(a_k-b_{n-k})} \end{bmatrix},$$

with $\{a_1, \ldots, a_k, b_1, \ldots, b_{n-k}\} \subseteq \mathbb{F}_q$ constituting a set of n distinct elements, then G generates a (systematic) MDS code [156].

2.5.1 Inverse of the Cauchy Matrix

We now present a proof of the invertibility of the Cauchy matrix appearing in [208]. Define the degree-m polynomials

$$A(x) = \prod_{i=1}^{m}(x - a_i), \quad B(x) = \prod_{i=1}^{m}(x - b_i).$$

The formal derivative $A'(x)$ of $A(x)$ is given from the product formula by:

$$A'(x) = \sum_{i=1}^{m} \prod_{j=1, j\neq i}^{m} (x - a_j),$$

so that

$$A'(a_\ell) = \prod_{j=1, j\neq \ell}^{m} (a_\ell - a_j).$$

Define:

$$A_i(x) = \frac{A(x)}{(x - a_i)A'(a_i)} = \prod_{j=1, j\neq i}^{m} \frac{(x - a_j)}{(a_i - a_j)}. \tag{2.3}$$

Then

$$A_i(x) = \begin{cases} 1, & x = a_i \\ 0, & x = a_\ell, \ell \neq i, \end{cases}$$

and thus serves as an indicator function for a_i. Analogously, let

$$B_u(x) = \frac{B(x)}{(x - b_u)B'(b_u)} = \prod_{\ell=1, \ell\neq u}^{m} \frac{(x - b_\ell)}{(b_u - b_\ell)},$$

be the indicator function for the elements $\{b_u\}_{u=1}^{m}$, so that:

$$B_u(a_i) = \prod_{\ell=1, \ell\neq u}^{m} \frac{(a_i - b_\ell)}{(b_u - b_\ell)}.$$

For any degree-$(m-1)$ polynomial $p(x)$, an application of Lagrange's interpolation formula gives us:

$$p(x) = \sum_{i=1}^{m} p(a_i) \prod_{j=1, j\neq i}^{m} \frac{(x - a_j)}{(a_i - a_j)} = \sum_{i=1}^{m} p(a_i) A_i(x).$$

Setting $p(x) = B_u(x)A(b_u)$, we obtain from (2.3) that

$$\frac{B_u(x)A(b_u)}{A(x)} = \sum_{i=1}^{m} \frac{B_u(a_i)A(b_u)}{A'(a_i)} \frac{1}{(x - a_i)}.$$

The LHS above satisfies

$$\frac{B_u(x)A(b_u)}{A(x)} \;=\; \begin{cases} 1, & x = b_u \\ 0, & x = b_\ell, \ell \neq u. \end{cases}$$

It follows that

$$\frac{B_u(b_\ell)A(b_u)}{A(b_\ell)} \;=\; \sum_{i=1}^{m} \left[\frac{-B_u(a_i)A(b_u)}{A'(a_i)}\right] \frac{1}{(a_i - b_\ell)} \;=\; \begin{cases} 1, & \ell = u \\ 0, & \text{else.} \end{cases}$$

It is evident from the above that the $(m \times m)$ matrix H whose $(u,i)^{th}$ entry is given by

$$H_{ui} \;=\; \frac{-B_u(a_i)A(b_u)}{A'(a_i)}, \quad 1 \leq u, i \leq m, \tag{2.4}$$

is the inverse of the Cauchy matrix. An alternate, more symmetric expression

$$H_{ui} \;=\; (a_i - b_u)B_u(a_i)A_i(b_u), \quad 1 \leq i, u \leq m,$$

can be obtained by noting in (2.4) that

$$A_i(x) \;=\; \frac{A(x)}{(x - a_i)A'(a_i)} \;\Longrightarrow\; \frac{A(b_u)}{A'(a_i)} \;=\; A_i(b_u)(b_u - a_i).$$

Notes

1. MDS codes of block length $q+1, q+2$: Let $\{\theta_1, \theta_2, \ldots, \theta_q\}$ denote the q elements in $\mathbb{F}_q$. It is straightforward to verify that the code having generator matrix

$$G \;=\; \begin{bmatrix} 1 & \ldots & 1 & 0 \\ \theta_1 & \ldots & \theta_q & 0 \\ \vdots & \vdots & \vdots & \\ \theta_1^{k-1} & \ldots & \theta_q^{k-1} & 1 \end{bmatrix}$$

is an $[n = q+1, k]$ MDS code [156]. If q is even, then the generator matrix

$$G \;=\; \begin{bmatrix} 1 & \ldots & 1 & 0 & 0 \\ \theta_1 & \ldots & \theta_q & 0 & 1 \\ \theta_1^2 & \ldots & \theta_q^2 & 1 & 0 \end{bmatrix}$$

yields an $[n = q + 2, k = 3]$ MDS code over $\mathbb{F}_q$ [156]. The dual of this code is an $[n = q + 2, k = q - 1]$ MDS code over $\mathbb{F}_q$. We also note that one can construct an $[n = k + 1, k]$ MDS code over a finite field $\mathbb{F}_q$ of any size.

2. The (linear) MDS conjecture: Let $N(k, q)$ denote the maximum possible block length of a k-dimensional MDS linear code over $\mathbb{F}_q$. If $k > q$, then it is known that $N(k, q) = k+1$. The MDS conjecture [209] states that if $2 \leq k \leq q$, then $N(k, q) = q + 1$, except when q is even and $k \in \{3, q - 1\}$, in which case $N(k, q) = q + 2$. The conjecture is shown to hold for prime q in [16] and for the case $k \leq 2p - 2$, when q is not prime and p is the characteristic of $\mathbb{F}_q$ in [17].

3. Vector MDS codes: Let $\mathcal{C}$ be an $(n, q^{k\alpha}, d_{\min})$ code of block length n, size $q^{k\alpha}$ and minimum distance $d_{\min}$ over a vector alphabet $\mathbb{F}_q^{\alpha}$. If $d_{\min} = (n - k + 1)$, then $\mathcal{C}$ is by the Singleton bound, an MDS code. Such MDS codes over a vector alphabet are frequently referred to as MDS array codes [23], [26]. Minimum storage regenerating codes [50], Even-Odd codes [25] and the Row-Diagonal Parity code [45] are examples of vector MDS codes. Both Even-Odd and Row-Diagonal Parity codes are double-erasure-recovering vector binary MDS codes, i.e., $q = 2$ and $d_{\min} = n - k + 1 = 3$.

3

Regenerating Codes

As noted in Section 1, the conventional repair of a Reed-Solomon code requires the download of an amount of data equal to the size of the data file being stored in order to repair a single failed node, despite the fact that each node only stores a small fraction of the contents of the data file. The amount of data downloaded to repair a single node is termed the repair bandwidth. To address this problem, Dimakis *et al.* [50] came up with a class of codes, termed as regenerating codes (RGCs), whose repair bandwidth is as small as possible. This seminal paper is all the more remarkable because a priori, it is not clear that any reduction in repair bandwidth is even possible. This section introduces RGCs and establishes their basic properties.

3.1 Definition and Terminology

We begin by describing the functioning of an RGC in a data-storage setting, before going on to provide a formal, mathematical definition. An RGC $\mathcal{C}$ is associated to a parameter set

$$\{(n, k, d), (\alpha, \beta), B, \mathbb{F}_q\},$$

where the role of the various parameters is explained below. The aim of the RGC is to store in efficient and reliable fashion data pertaining to a data file $\mathcal{B}$ that is comprised of B symbols, termed the message symbols, belonging to an underlying finite field $\mathbb{F}_q$. The B message symbols are first mapped onto a set of $n\alpha$ symbols over $\mathbb{F}_q$ and the $n\alpha$ symbols are then distributed evenly, across a set of n storage units called nodes, so that each node stores exactly α symbols. The creation of the $n\alpha$

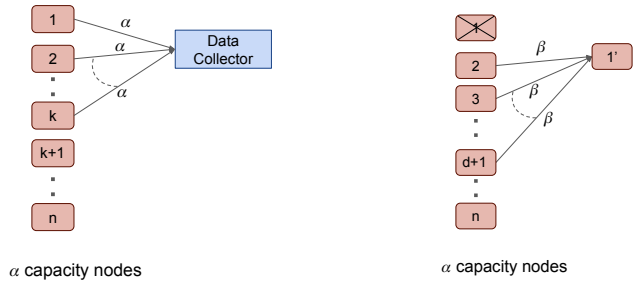

Figure 3.1: An illustration of the data-collection and node-repair properties of an $\{(n,k,d),\ (\alpha,\beta),\ B,\ \mathbb{F}_q\}$ RGC. Here, node $1'$ is a replacement node for the failed node 1.

codes symbols and their distribution across n nodes should be such that the two key properties described below and illustrated in Fig. 3.1, are satisfied:

1. *Data Collection Property:* It should be possible to recover the B message symbols, given access to the contents of any k of the n storage nodes.

2. *Node Repair or Regeneration Property:* If a particular node or storage unit fails, then it should be possible to recover from such a failure by having a replacement of the failed node connect to any d of the remaining $(n-1)$ nodes, and download β symbols from each of these d nodes to arrive at the α symbols stored by the replacement node. The repair is called *exact repair (ER)* if the contents of the replacement node following node repair are identical to the contents of the original failed node. The repair is called *functional repair (FR)* if, following node repair, the contents of the set of $(n-1)$ surviving nodes, together with the contents

of the new replacement node, once again meet the requirements of an RGC.

The parameter α is termed the sub-packetization level of the RGC, motivated by viewing the collection of α $\mathbb{F}_q$-symbols as a packet and each individual $\mathbb{F}_q$-symbol as a sub-packet. The d assisting nodes are called helper nodes and the parameter d, the repair degree[1]. The total number $d\beta$ of $\mathbb{F}_q$ symbols downloaded from the d helper nodes for repair of a failed node is termed the repair bandwidth of the RGC. The rate of the RGC is given by $R = \frac{B}{n\alpha}$. Its reciprocal $\frac{n\alpha}{B}$ is the storage overhead. We note that under FR, the contents of a node can change with time, and that ER is a particular and more stringent instance of functional repair. ER is more desirable in practice, as it simplifies logistics.

3.1.1 Formal Definition of Exact and Functional Repair

We begin with the case of exact repair.

Definition 3 (Exact Repair Regenerating Code)**.** An $(n, M, d_{\min})$ code $\mathcal{C}$ over an alphabet $\mathcal{A}$ is said to be an ER RGC having parameter set

$$\{(n, k, d), (\alpha, \beta), B, \mathbb{F}_q\},$$

where $\{k, d, \alpha, \beta, B\}$ are positive integers with $1 \leq k \leq d \leq (n-1)$, if the following conditions are satisfied:

1. $\mathcal{A} = \mathbb{F}_q^{\alpha}$,
2. $M = q^B$,
3. $d_{\min} \geq (n - k + 1)$,
4. and where, for every index $i \in [n]$, and every subset $S \subseteq [n] \setminus \{i\}$ of size $|S| = d$, there are functions

$$h_{i,j,S} : \mathbb{F}_q^{\alpha} \rightarrow \mathbb{F}_q^{\beta}, \ \forall j \in S,$$

as well as functions

$$f_{i,S} : \mathbb{F}_q^{d\beta} \rightarrow \mathbb{F}_q^{\alpha}$$

[1]This is in analogy with the analogous term first introduced in the context of a locally recoverable code. Locally recoverable codes are introduced in Section 10.

such that

$$c_i \quad = \quad f_{i,S}(h_{i,j,S}(c_j), j \in S), \tag{3.1}$$

for every codeword $(c_1, c_2, \cdots, c_n) \in \mathcal{C}$.

In any $(n, M, d_{\min})$ code, a codeword is uniquely determined given any subset of $(n - d_{\min} + 1)$ code symbols. Thus the condition $d_{\min} \geq (n - k + 1)$, has the implication that a codeword can be uniquely recovered from the contents of any k code symbols. This in the context of distributed storage is the data-collection property. Clearly, equation (3.1) corresponds to the node-repair property.

We present below the formal definition of an FR RGC by defining this notion first for a collection of codes, rather than for a single code, for reasons explained in Remark 2.

Definition 4 (Functional Repair Regenerating Code)**.** A (finite) collection $\{\mathcal{C}_\ell\}_{\ell=1}^{L}$ of L (n, M) codes over a common alphabet $\mathcal{A}$ is said to be a collection of FR RGCs having common parameter set

$$\{(n, k, d), (\alpha, \beta), B, \mathbb{F}_q\},$$

where $\{k, d, \alpha, \beta, B\}$ are positive integers with $1 \leq k \leq d \leq (n-1)$, if the following conditions are satisfied:

1. $\mathcal{A} = \mathbb{F}_q^\alpha$,
2. $M = q^B$,
3. All codes $\mathcal{C}_\ell$, $\ell = 1, 2, \cdots, L$ have minimum distance satisfying: $d_{\min}(\mathcal{C}_\ell) \geq (n - k + 1)$,
4. And where, for every index $\ell \in [L]$, $i \in [n]$, and every subset $S \subseteq [n] \setminus \{i\}$ of size $|S| = d$, there are functions

$$h_{i,j,S}^{(\ell)} : \ \mathbb{F}_q^\alpha \quad \rightarrow \quad \mathbb{F}_q^\beta, \quad \forall j \in S,$$

as well as functions

$$f_{i,S}^{(\ell)} : \ \mathbb{F}_q^{d\beta} \quad \rightarrow \quad \mathbb{F}_q^\alpha,$$

such that if we set

$$\hat{c}_i \quad = \quad f_{i,S}^{(\ell)}\Big(\; h_{i,j,S}^{(\ell)}(c_j), j \in S \; \Big), \tag{3.2}$$

we have

$$\{(c_1, \cdots, c_{i-1}, \hat{c}_i, c_{i+1}, \cdots, c_n) \mid (c_1, \cdots, c_n) \in \mathcal{C}_\ell\} \quad = \quad \mathcal{C}_{\ell'}$$

for some $\ell', 1 \leq \ell' \leq L$. An FR RGC is then simply a code that is an element of such a collection of FR RGCs and shares the same parameter set as does the collection of FR RGCs.

Remark 2. The naive approach would be to define an FR RGC by defining it as a code that, apart from the data collection property, has the property that following node repair, one arrives at a second code that is also an FR RGC. The above approach aims to avoid such a circular definition. The presence of a collection of FR RGCs, as appearing in the definition above, can be seen in the construction of an FR RGC appearing in [212]. The case of ER may be regarded as corresponding to the special case of FR when there is just a single code in the collection, i.e., when $L = 1$.

Linear RGC There are four classes of mappings associated with an RGC:

(a) The mapping from B message symbols to the $n\alpha$ contents of the n nodes,

(b) The mapping used to recover the B message symbols from the $k\alpha$ contents of a specific set of k nodes,

(c) The mapping $\mathbb{F}_q^\alpha \rightarrow \mathbb{F}_q^\beta$ used by node j to determine the β symbols to be forwarded to the replacement of failed node i, given knowledge of the remaining $(d-1)$ helper nodes,

(d) The mapping used by the replacement node to extract the α symbols to be stored from the $d\beta$ symbols supplied to the replacement node, by a specific set of d helper nodes.

We will say that an RGC is linear if all four mappings above are linear. All the RGCs that will be encountered in this monograph will be linear.

3.2 Bound on File Size

We adopt the information-theoretic approach[2] of Shah *et al.* [211] to establishing the fundamental bound on file size B for a given parameter set $\{(n,k,d),(\alpha,\beta),\mathbb{F}_q\}$, appearing in (3.7).

The RGC is of size q^B and we can therefore without loss of generality, associate a unique message vector $M \in \mathbb{F}_q^B$ to each codeword. We will assume that M as a random variable[3] is uniformly distributed over $\mathbb{F}_q^B$. The contents of each node as well as the repair data symbols that are passed between nodes are thus also random variables, that are functions of M.

Let W_i denote the random variable taking on values in $\mathbb{F}_q^\alpha$ that represents the contents of the ith node. Clearly,

$$H(W_i) \quad \leq \quad \alpha, \tag{3.3}$$

where we have taken the unit of entropy as $\log_2(|\mathbb{F}_q|) = \log_2(q)$ bits. In all of our discussion here, the entropy will always be measured in units of $\log_2(q)$ bits. For $A \subseteq [n]$, we use the notation W_A to denote the set

$$W_A \quad = \quad \{W_i \mid i \in A\}.$$

The data-collection property required of an ER RGC imposes the following additional constraints:

$$H(W_A) \;=\; B, \qquad \text{and} \qquad H(M \mid W_A) \;=\; 0 \tag{3.4}$$

for $A \subseteq [n]$, $|A| \geq k$. For $x, y \in [n]$, and $D \subseteq [n]$, such that $x \notin D$, $y \notin D$ and

$$|D \cup \{x\}| \quad = \quad d,$$

we use ${}_D S_x^y$ to denote the random variable corresponding to the helper data sent by the helper node x to the replacement of a failed node y,

[2]The original proof of file size upper bound for RGCs by Dimakis *et al.* [50] was using a network coding approach, which we discuss in Section 3.4.

[3]Strictly speaking M is a random vector, but we will use the term random variable to refer to either a random vector or random variable. Also, random variables typically take on real values, however, this is not an essential restriction.

when the set of d helper nodes is the set $D \cup \{x\}$. We will drop the prescript D and simply write S_x^y if D is understood from the context. As an example, this can happen if $n = (d+1)$, in which case $D = [n] \setminus \{x, y\}$.

Given subsets $X, Y, D \subseteq [n]$, with

$$|D \cup X| \;=\; d, \quad D \cap \{X \cup Y\} \;=\; \phi,$$

we define ${}_DS_X^Y \;=\; \{{}_AS_x^y \mid x \in X, y \in Y, x \neq y,\; A = D \cup X \setminus \{x\}\;\}$. For the case $X = Y$, we use the short-hand notation ${}_DS_X$ to indicate ${}_DS_X^X$. In all these cases, we will drop the prescript D if it is understood from the context and simply write S_X^Y, S_X in place of ${}_DS_X^Y$, ${}_DS_X$ respectively.

From the definition of an RGC, it follows that

$$H({}_DS_x^y) \;\leq\; \beta. \tag{3.5}$$

For every $i \in [n]$, the exact-repair condition imposes the constraint

$$H(W_i \mid S_X^i) \quad = \quad 0,\; |X| = d,\; i \notin X. \tag{3.6}$$

Theorem 2 (Fundamental Bound on File Size under Exact Repair)**.** In any $\{(n,k,d),(\alpha,\beta),B,\mathbb{F}_q\}$ exact-repair RGC, we must have:

$$B \;\leq\; \sum_{i=1}^{k} \min\{\alpha, (d-i+1)\beta\}. \tag{3.7}$$

Proof. Let $\mathcal{C}$ denote the RGC. Let D be an arbitrary subset of the n nodes of size $|D| = (d+1)$. Let $\mathcal{C}_D$ denote the restriction of $\mathcal{C}$ to the subset D of nodes. Clearly, the code $\mathcal{C}_D$ stores the same B message symbols as does the RGC $\mathcal{C}$. Furthermore, $\mathcal{C}_D$ is also an RGC having parameter set

$$\{((d+1),k,d),(\alpha,\beta),B,\mathbb{F}_q\}.$$

We will focus our attention from here on, on the code $\mathcal{C}_D$ instead of the code $\mathcal{C}$, and establish the upper bound on file size B. The same bound will then continue to apply to the code $\mathcal{C}$. We will also assume without loss of generality, that the n nodes are indexed so that $D = [d+1]$. From the data collection property of an RGC, we have

$$B \;=\; H(W_1, \ldots, W_k) \;=\; \sum_{i=1}^{k} H(W_i \mid W_{[i-1]}),$$

where $[i-1] := \{1, 2, \ldots, i-1\}$, $[0] := \phi$ and where, by $W_{[i-1]}$, we mean the collection $\{W_j \mid j \in [i-1]\}$. The theorem then follows from Lemma 1 below. □

Lemma 1.

$$H(W_i \mid W_{[i-1]}) \;\leq\; (d-i+1)\beta.$$

Proof. We will prove this lemma in three steps.

Step 1 - We set $A := [d+1] \setminus [i]$, and note that

$$H(W_i \mid S^i_{[i-1]}, S^i_A) \;=\; 0,$$

which follows from the repair property of an RGC.

Step 2 - We will show using Lemma 2 below, that this implies

$$H(W_i \mid S^i_{[i-1]}) \;\leq\; H(S^i_A). \tag{3.8}$$

Since, from the definition of an RGC, we have

$$H(S^i_A) \;\leq\; |A|\beta \;=\; (d-i+1)\beta,$$

this in turn implies that

$$H(W_i \mid S^i_{[i-1]}) \;\leq\; (d-i+1)\beta. \tag{3.9}$$

Step 3 - The information passed on by a helper node to a replacement of the failed node i, is clearly a function of the contents of the helper node. This implies that

$$H(S^i_{[i-1]} \mid W_{[i-1]}) \;=\; 0.$$

We will use this observation, coupled with Lemma 3 below, to show that

$$H(W_i \mid W_{[i-1]}) \;\leq\; H(W_i \mid S^i_{[i-1]}), \tag{3.10}$$

thus completing the proof of Lemma 1. □

It remains to state and prove Lemmas 2 and 3, appearing in the proof above. The random variables $\{U, X, Y, Z\}$ appearing in the two lemmas should be interpreted follows

$$X \;\Leftrightarrow\; W_i, \quad Y \;\Leftrightarrow\; S^i_{[i-1]}, \quad Z \;\Leftrightarrow\; S^i_A, \quad U \;\Leftrightarrow\; W_{[i-1]}.$$

Lemma 2. Let X, Y, Z be random vectors whose components take values in $\mathbb{F}_q$. Then,

$$H(X \mid Y,Z) \;=\; 0 \implies H(X \mid Y) \;\leq\; H(Z \mid Y) \;\leq\; H(Z).$$

Proof:

$$\begin{aligned} H(X,Z \mid Y) &= H(X \mid Y) + \underbrace{H(Z \mid X,Y)}_{\geq 0} \\ &= H(Z \mid Y) + \underbrace{H(X \mid Y,Z)}_{=0}. \end{aligned}$$

It follows that

$$H(X \mid Y) \;\leq\; H(Z \mid Y) \;\leq\; H(Z).$$

□

This completes Step 2 and we have established the inequality

$$H(W_i \mid S_{[i-1]}^i) \;\leq\; (d-i+1)\beta. \tag{3.11}$$

Lemma 3, appearing in Step 3, will complete the proof of Lemma 1, by showing that

$$H(W_i \mid W_{[i-1]}) \;\leq\; H(W_i \mid S_{[i-1]}^i). \tag{3.12}$$

Lemma 3.

$$H(X \mid U) \;\leq\; H(X \mid Y) \text{ when } H(Y \mid U) \;=\; 0.$$

Proof:

$$\begin{aligned} H(U,X,Y) &= H(U) + H(X \mid U) + H(Y \mid X,U) \\ &= H(Y) + H(X \mid Y) + H(U \mid X,Y). \end{aligned}$$

Since $H(Y \mid U) = 0 \implies H(Y \mid X,U) = 0$ and $H(U,Y) = H(U)$, we have that

$$\begin{aligned} H(X \mid U) &= H(X \mid Y) - [H(U) - H(Y) - H(U \mid X,Y)] \\ &= H(X \mid Y) - [H(U,Y) - H(Y) - H(U \mid X,Y)] \\ &= H(X \mid Y) - [H(U \mid Y) - H(U \mid X,Y)] \\ &\leq H(X \mid Y). \end{aligned}$$

□

Remark 3. While the proof given above is for the case of exact repair, it extends in straightforward fashion to the case of functional repair. In an RGC with functional repair, the contents of the nodes, as well as the data transferred for node repair can change with time.

Let us assume that we are at time instant t in the functional-repair setting. As in the case of ER, we restrict attention to a subset of $(d+1)$ nodes that are numbered 1 through $(d+1)$. Let t_i denote the last time instant at which node i was repaired prior to time t. We assume without loss of generality, that

$$t_1 \ < \ t_2 \ < \ \cdots \ < \ t_{k-1} \ < \ \ t_k \ < t.$$

With respect to the proof given above for the ER case, we now interpret W_i, for $1 \leq i \leq k$, as the contents of node i at time t. We interpret $S^i_{[i-1]}$, $i = 1, 2, \cdots, k$ as the data passed on by helper node $j \in [i-1]$ to the replacement of the ith node at time t_i, i.e., the time instant which node i failed. Similarly, S^i_A denotes the helper information passed on by nodes in set A for repair of node i at time t_i. With this, it can be verified by following the argument made for the ER case, that the information-theoretic arguments remain unchanged and we therefore arrive at the same bound.

3.3 Storage-Repair-Bandwidth Tradeoff

For a given file size B, the storage overhead and normalized repair bandwidth are given respectively by $\frac{n\alpha}{B}$ and $\frac{d\beta}{B}$. Thus for a fixed value of file size B, block length n, and repair degree d, the parameter α is indicative of the amount of storage overhead while β determines the normalized repair bandwidth. We will say that an RGC having parameters $\{(n,k,d),(\alpha,\beta),B,\mathbb{F}_q\}$ is optimal if (a) the file-size bound in (3.7) is met with equality and if further, (b) reducing either α or β causes the bound to be violated[4].

[4]The latter condition is inserted since at the extreme MSR case, one could have $B = \alpha k$ and β very large while satisfying (3.7), while at the same time the inequality could also be satisfied with $\beta = \frac{\alpha}{(d-k+1)}$. At the other extreme MBR end, equality could hold with $B = (dk - \binom{k}{2})\beta$ and α very large, while $\alpha = d\beta$ would suffice for equality to hold.

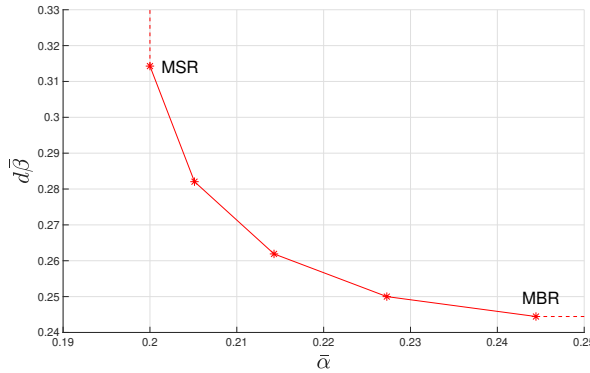

Figure 3.2: The normalized storage-repair-bandwidth tradeoff under functional repair, for the parameters $(k = 5, d = 11)$.

It will be convenient at this point, to introduce normalized versions of the parameters (α, β), given by

$$\bar{\alpha} := \frac{\alpha}{B}, \qquad \bar{\beta} := \frac{\beta}{B}.$$

Then by dividing both sides of (3.7) by B, we obtain

$$1 \leq \sum_{i=1}^{k} \min\{\bar{\alpha}, (d-i+1)\bar{\beta}\}. \tag{3.13}$$

For fixed (k, d), the locus of all pairs $(\bar{\alpha}, d\bar{\beta})$ that satisfy (3.13) with equality will be shown in Section 6 to be a piece-wise linear curve as can be seen in Fig. 3.2. For fixed value of block length n, this curve represents a tradeoff between storage overhead $n\bar{\alpha}$ on the one hand, and normalized repair bandwidth $d\bar{\beta}$ on the other. The network coding approach to deriving the fundamental bound on file size (see Section 3.4) tells us that for every set of parameters $\{(n, k, d), (\alpha, \beta)\}$ there exists an RGC having file size B satisfying (3.7). However, network coding only guarantees the existence of an RGC that is repaired using functional repair. For this reason, the plot of the pairs $(\bar{\alpha}, d\bar{\beta})$ that satisfy (3.13) with equality is referred to as the FR tradeoff.

The corresponding tradeoff under exact repair, called the ER tradeoff, is harder to characterize and is discussed further in Sections 6 and 7.

3.3.1 MSR and MBR Codes

Clearly, the smallest value of $\bar{\alpha}$ for which the equality can hold in (3.13) is given by $\bar{\alpha} = \frac{1}{k}$. Given $\bar{\alpha} = \frac{1}{k}$, the smallest permissible value of $d\bar{\beta}$ is given by $d\bar{\beta} = \frac{d}{k(d-k+1)}$. The corresponding pair $(\bar{\alpha}, d\bar{\beta})$ thus represents the point of the tradeoff corresponding to minimum possible storage overhead and additionally, minimum possible repair bandwidth, given that the storage overhead is as small as possible. Codes achieving (3.7) with $\bar{\alpha} = \frac{1}{k}$ and $d\bar{\beta} = \frac{d}{k(d-k+1)}$ are for this reason, known as minimum storage regenerating (MSR) codes.

Similarly, at the other end of the tradeoff, the smallest possible value of the normalized repair bandwidth is given by $d\bar{\beta} = \frac{d}{dk-\binom{k}{2}}$. Given $d\bar{\beta} = \frac{d}{dk-\binom{k}{2}}$, the smallest permissible value of $\bar{\alpha}$ is given by $\bar{\alpha} = d\bar{\beta}$. The corresponding pair $(\bar{\alpha}, d\bar{\beta})$ represents this time, the point of the tradeoff corresponding to minimum possible repair bandwidth and additionally, minimum possible storage overhead, given that the normalized repair bandwidth is as small as possible. Codes achieving (3.7) with $d\bar{\beta} = \frac{d}{dk-\binom{k}{2}}$ and $\bar{\alpha} = d\bar{\beta}$ are for this reason, known as minimum (repair) bandwidth regenerating (MBR) codes.

As noted in Definitions 3,4, the minimum Hamming distance $d_{\min}$ of an RGC must satisfy $d_{\min} \geq (n-k+1)$. By the Singleton bound, the largest size M of a code of block length n and minimum distance $d_{\min}$ is given by $M \leq Q^{n-d_{\min}+1}$, where Q is the size of alphabet of the code. Since the alphabet size $Q = q^{\alpha}$ in the case of an RGC, it follows that the size M of an RGC must satisfy $M \leq q^{k\alpha}$, or equivalently $q^{B} \leq q^{k\alpha}$, i.e., $B \leq k\alpha$. But $B = k\alpha$ in the case of an MSR code and it follows that every MSR code is an MDS code over the vector alphabet $\mathbb{F}_q^{\alpha}$. In the literature, MDS codes over a vector alphabet are also referred to as MDS array codes. For more on MDS array codes, see the notes section appearing at the end of Section 2.

From a practical perspective, ER RGCs are easier to implement as the contents of the n nodes in operation do not change with time. Partly for this reason and partly for reasons of tractability, with few exceptions, most constructions of RGCs belong to the class of ER RGCs. Examples of FR RGC include the $d = (k+1)$ construction in [212] as well as the construction in [98].

Early constructions of RGCs focused on the two extreme points of the storage-repair-bandwidth tradeoff, namely the MSR and MBR points. The storage industry places a premium on low storage overhead. This is not too surprising, given the vast amount of data, running into the exabytes, stored in today's data centers. In this connection, we note that the maximum rate of an MBR code is given by:

$$R_{\text{MBR}} = \frac{B}{n\alpha} = \frac{(dk - \binom{k}{2})\beta}{nd\beta} = \frac{dk - \binom{k}{2}}{nd},$$

which can be shown to satisfy the upper bound $R_{\text{MBR}} \leq \frac{1}{2}$, achieved with equality when $k = d = (n-1)$. This makes MSR codes of greater practical interest when minimization of storage overhead is a primary objective.

Apart from the requirements of low repair bandwidth and low value of storage overhead, from a practical perspective, there are some additional properties desired of an RGC. These include small field size, small sub-packetization level α and minimal disk read and computation needed for node repair.

3.4 Network Coding Approach to the File-Size Bound

The fundamental upper bound on file size appearing in Theorem 2 was originally established by Dimakis *et al.* [50] using a network coding approach [3], [122] and by invoking the cut-set bound. For this reason, the upper bound on file size bound is often referred to in the literature, as the cut-set bound. We present an overview of this proof below.

3.4.1 The Cut-Set Upper Bound

The derivation of the upper bound only requires the RGC to be an FR RGC. Since an ER RGC is also an FR RGC, the bound clearly applies to ER RGCs as well. Let $\mathcal{C}$ be an FR RGC having parameter set

$$\{(n,\ k,\ d),\ (\alpha,\ \beta),\ B,\ \mathbb{F}_q\ \}.$$

Since the RGC is of size q^B, as before, we associate without loss of generality, a unique message vector $M \in \mathbb{F}_q^B$ to each codeword. Over

time, nodes will undergo failures and every failed node will be replaced by a replacement node. Let us assume to begin with, that we are only interested in the behavior of the RGC over a finite-but-large number $N \gg n$ of node repairs. Moreover, we will assume that nodes are repaired using functional repair. For simplicity, we assume that repair is carried out instantaneously. Then, at any given time instant t, there are n functioning nodes whose collective contents constitute an RGC and a data collector should be able to connect to any subset of k nodes, download all of the contents of these k nodes and use these to recover the B message symbols, $\{u_i \in \mathbb{F}_q\}_{i=1}^{B}$. Clearly, in all, there are at most $N\binom{n}{k}$ distinct data collectors, each corresponding to a distinct choice of k nodes to which the data collector connects.

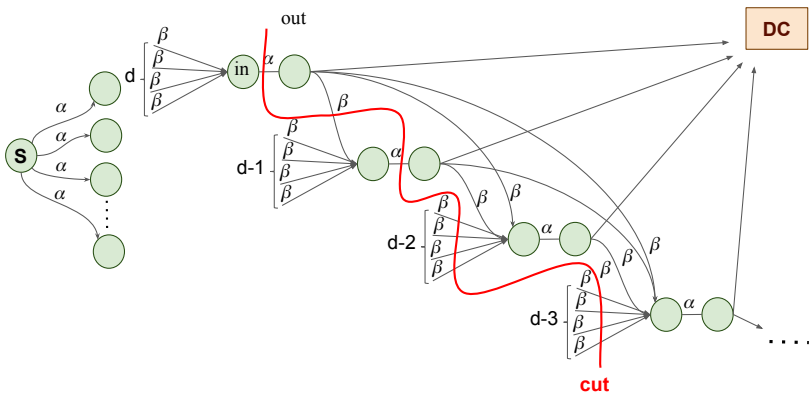

Figure 3.3: The directed, capacitated graph associated to an RGC that over time, undergoes a finite number N of node repairs. The label on each edge, indicates the capacity of that edge. Here DC denotes the data collector.

Next, we create a source node that possesses the B message symbols $\{u_i\}_{i=1}^{B}$, and draw edges connecting the source to the initial set of n nodes among which, data pertaining to the message symbols was distributed in coded form. We also draw edges connecting the d helper nodes that assist a replacement node and the replacement node, as well as edges connecting each data collector with the corresponding set of k nodes from which the data collector downloads data. All edges are directed in the direction of information flow. We associate a capacity β with edges emanating from a helper node to a replacement node

and an ∞ capacity with all other edges. Each node can only store α symbols over $\mathbb{F}_q$. We incorporate this constraint by using a standard graph-theory construct, in which a node is replaced by 2 nodes separated by a directed edge (leading towards a data collector) of capacity α. We have in this way, arrived at a graph (see Fig. 3.3) in which there is a single source S and at most $N\binom{n}{k}$ sinks $\{T_i\}$.

Each sink T_i would like to be able to reconstruct all the B source symbols $\{u_i\}$ from the symbols it receives. This is precisely the multicast setting of network coding. A principal result in network coding tells us that in a multicast setting, one can transmit messages along the edges of the graph in such a way that each sink T_i is able to reconstruct the source data, provided that the minimum capacity of a cut separating S from T_i is $\geq B$. A cut separating S from T_i is simply a partition of the nodes of the network into 2 sets: a subset A_i of the nodes containing S and whose set-theoretic complement A_i^c, contains T_i. The capacity of the cut is the sum of the capacities of all the edges leading from a node in A_i to a node in A_i^c. A careful examination of the graph will reveal that the minimum capacity Q of a cut separating a sink T_i from source S is given by $Q = \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}$ (Fig. 3.3 shows an example cut separating source from sink). This leads to the upper bound (3.7) on file size:

$$B \leq \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}.$$

3.4.2 Achievability

Network coding also employs the Combinatorial Nullstellensatz [5] to show that when only a finite number of node failures and corresponding regenerations take place, this bound is achievable, and moreover, achievable using linear network coding, i.e., achievable using only linear operations at each node in the network for a sufficiently large value q of the finite field $\mathbb{F}_q$. In a subsequent result [251], Wu used the specific structure of the graph to show that even in the case when the number of sinks is infinite, the upper bound in (3.7) continues to be achievable using linear network coding.

In this way, one can draw upon principles of network coding to characterize the maximum file size of an RGC given parameters $\{k, d, \alpha, \beta\}$ for the case of functional repair.

3.5 Overview of RGC-Related Topics in the Monograph

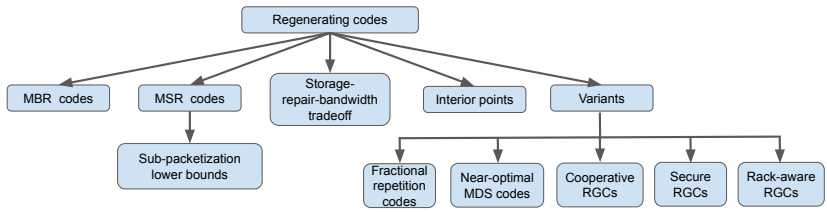

Figure 3.4: Topics related to RGCs that are covered elsewhere in the monograph.

Table 3.1: Constructions for MBR codes, MSR codes and interior-point ER RGCs that are presented in the monograph. All of the constructions appearing in the table are explicit. (We only provide brief descriptions however, of the Small-d MSR, Cascade and Moulin code constructions.)

Type of RGC	Code	Section
MBR	Polygonal [179]	4.1
MBR	Product-Matrix [185]	4.2
MSR	Product-Matrix [185]	5.1
MSR	Diagonal [255]	5.2
MSR	Coupled-Layer [137], [205], [256]	5.3
MSR	Small-d [239]	5.4
Interior point	Determinant [61], [63]	7.1
Interior point	Cascade [64]	7.2
Interior point	Moulin [54]	7.3

Fig. 3.4 presents an overview of the RGC-related topics discussed in this monograph. Sections 4 and 5 present several constructions of exact-repair MBR and MSR codes respectively. The storage-repair-bandwidth tradeoffs corresponding to FR and ER appear in Section 6. Constructions directed towards interior points of the tradeoff in the

case of ER appear in Section 7. Bounds on the sub-packetization level of an MSR code appear in Section 8. Several variants of RGCs such as fractional repetition codes and cooperative RGCs are discussed in Section 9. A tabular listing of the constructions of RGCs that appear in Sections 4, 5 and 7 is provided in Table 3.1.

Notes

1. Liquid storage: In [150]–[152], a different approach to coded distributed storage is adopted. The broad objectives remain the same however, namely, minimizing storage overhead as well as the amount of data download needed to carry out node repair while ensuring reliable data storage. The liquid storage systems described in [150]–[152] employ erasure codes having large block length, and as a result, the number of redundant nodes is proportionally larger. This permits an approach to node repair termed as lazy repair, in which the repair center is able to wait until several nodes have failed before proceeding with node repair. This is to take advantage of the fact that the simultaneous repair of t failed nodes can be carried out more efficiently in terms of the amount of helper data that needs to be downloaded for node repair, in comparison with separately carrying out the individual repair of the t nodes. In [150]–[152], the authors assume that nodes fail at a certain rate and the focus is on minimizing both the peak and average repair rate at which data needs to be read from the storage nodes by a centralized repair center.

4

MBR Codes

As noted in Section 3, MBR codes are the subclass of RGCs that have minimum possible normalized repair bandwidth and additionally, smallest value of storage overhead given that the normalized repair bandwidth is as small as possible. MBR codes thus correspond to one of the two extreme points of the storage-repair bandwidth tradeoff. The repair bandwidth, $d\beta$, of an MBR code equals the amount of data to be regenerated, α and the file size, B is given by $B = kd\beta - \binom{k}{2}\beta$.

An MBR code is said to possess the help-by-transfer (HBT) property if a failed node can always be regenerated without need for any form of finite-field computation carried out at the helper nodes. If in addition, no computation is required even at the replacement node, the code is said to possess the repair-by-transfer (RBT) property. In other words, the RBT property implies that node repair is accomplished by simply copying over a subset of symbols contained in the d helper nodes to the replacement node. It follows from this that if an MBR code possesses the RBT property, each scalar code symbol contained in a node must also be contained in at least one other node. As a form of converse result, it is shown in [132] that it is not possible to construct an MBR code in which even a single scalar symbol is repeated more than twice. This

statement is true regardless of whether or not the MBR code possesses the RBT property.

It follows that if an MBR code possesses the RBT property, its $n\alpha$ scalar code symbols must be comprised of a set of $\frac{n\alpha}{2}$ distinct code symbols, each replicated twice. In [213], it is shown that an MBR code with $d < n-1$, cannot possess the HBT (and hence the RBT) property. We provide in this section, two constructions (see Table 4.1) of MBR codes. In the first construction [179], $d = n-1$ and the code satisfies the RBT property. The second construction [185] yields general MBR codes, i.e., MBR codes for all $d \leq n-1$ (which do not in general, possess the RBT property).

Table 4.1: The explicit MBR code constructions described in this section.

MBR code	Parameters	Field size	Attributes
Polygonal [179]	$d = n-1$, $\beta = 1$	$O(n^2)$	RBT
Product-Matrix [185]	all d, $\beta = 1$	$O(n)$	-

4.1 Polygonal MBR Code

The polygonal MBR code construction by Rashmi *et al.* [179], [211] yields MBR codes satisfying the RBT property for all parameters $k \leq d = n-1$ and $\beta = 1$. We begin with an example construction for the case $\{(n = 5, k = 3, d = 4), (\alpha = 4, \beta = 1), B = 9\}$, which we will refer to as the pentagon MBR code.

Consider a complete graph with $n = 5$ vertices. It has $\binom{5}{2} = 10$ edges. The $B = 9$ symbols of the data file are encoded using a $[10, 9, 2]$ MDS code to produce ten code symbols. Each code symbol is assigned to a distinct edge. Each node of the pentagon MBR code stores the code symbols assigned to the edges incident on that node (see Fig. 4.1). We will now verify that the example construction indeed satisfies both data collection and RBT properties.

Data Collection: Any collection of $k = 3$ nodes contains nine distinct code symbols of the $[10, 9, 2]$ MDS code. This is sufficient to recover all 10 code symbols and in this way, the 9 message symbols that make up the data file.

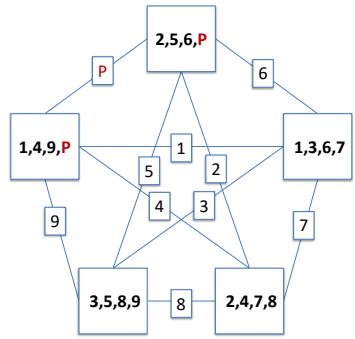

Figure 4.1: An example RBT MBR code construction for the parameter set $(n = 5, k = 3, d = 4)$. The file size $B = 9$ here.

Node Repair: To repair a failed node, each helper node simply provides to the replacement node, the code symbol associated to the edge it shares with the failed node. Thus, node repair is accomplished by merely transferring $\beta = 1$ code symbol from each of the $(n-1) = 4$ helper nodes to the replacement node.

4.1.1 The General Polygonal MBR Construction

The general polygonal MBR code construction yields MBR codes having parameter set of the form

$$\left\{(n, k, d = n-1),\ (\alpha = n-1, \beta = 1),\ B = k(n-1) - \binom{k}{2}\right\}$$

for any $n \geq 3$ and requiring $O(n^2)$ field size. The construction is most easily described in terms of a complete graph $\mathcal{G}$ on n vertices having $N = \binom{n}{2}$ edges. Let $\mathcal{C}_{\text{sc}}$ be a scalar $[N, B, N-B+1]$ MDS code. Each code symbol in $\mathcal{C}_{\text{sc}}$ is mapped on to an edge in $\mathcal{G}$. Each node is made to store the $\alpha = n-1$ code symbols corresponding to the edges incident on the particular node. Node repair then proceeds exactly as in the case of the example pentagon MBR code. The replacement node simply downloads from each helper node, the unique symbol of $\mathcal{C}_{\text{sc}}$ it shared in common with the failed node. With regard to the data collection property, downloading all the contents of any k nodes yields

$$\alpha k - \binom{k}{2} \quad = \quad dk - \binom{k}{2} \quad = \quad B$$

distinct code symbols from $\mathcal{C}_{\text{sc}}$. The MDS property of $\mathcal{C}_{\text{sc}}$ now allows all the B message symbols associated to the data file to be recovered. Note that the requirement of an MDS code of block length $N = \binom{n}{2}$ places a requirement of $O(n^2)$ on the finite field needed to construct the MBR code.

The construction of a family of RBT MBR codes with $d = n - 1$ and a reduced $O(n)$ field-size requirement is presented in [144].

4.2 Product-Matrix MBR Code

A product-matrix framework was introduced by Rashmi *et al.* [185] that provides constructions for both MSR and MBR codes. We describe in this subsection, the product-matrix MBR (PM-MBR) code construction. This construction yields MBR codes for all parameter sets of the form

$$\{(n,k,d),\ (\alpha = d, \beta = 1),\ B = kd - \binom{k}{2},\ \mathbb{F}_q\ \},$$

and requires a $q = O(n)$ field size.

Let G be an $(n \times d)$ matrix over $\mathbb{F}_q$ of the form:

$$\underbrace{G}_{n\times d} \triangleq \left[\ \underbrace{G_1}_{n\times k} \quad \underbrace{G_2}_{n\times(d-k)}\ \right],$$

where:

- Every $(k \times k)$ sub-matrix of G_1 is non-singular
- Every $(d \times d)$ sub-matrix of G is non-singular.

The two requirements can be met, for example, by choosing G to be a Cauchy or Vandermonde matrix, both of which require $O(n)$ field-size. The matrix G plays the role of generator matrix for the PM-MBR code.

Next, we introduce a data-bearing, symmetric, $(d \times d)$ matrix M of the form

$$\underbrace{M}_{d\times d} \triangleq \left[\begin{array}{cc} \underbrace{S}_{k\times k} & \underbrace{V}_{k\times(d-k)} \\ \underbrace{V^T}_{(d-k)\times k} & \underbrace{\mathbf{0}}_{(d-k)\times(d-k)} \end{array}\right],$$

where S is a symmetric $(k \times k)$ matrix. Since S is symmetric, the matrix M can store at most

$$\binom{k+1}{2} + k(d-k) \;=\; kd - \binom{k}{2} \;=\; B,$$

distinct elements. The matrix M is accordingly populated by the B message symbols associated to the data file and may be regarded as a the matrix analogue of the message vector associated to a scalar block code.

Each codeword in the PM-MBR code is then represented by an $(n \times d)$ code matrix C that is the product of the matrices G, M:

$$\underbrace{C}_{n\times d} \triangleq \underbrace{G}_{n\times d}\underbrace{M}_{d\times d}.$$

If $\underline{c}_i^T$ denotes the ith row of the code matrix C, $i \in \{1, \ldots, n\}$, the contents of the ith node are then precisely the components of $\underline{c}_i$.

Data Collection: Consider any collection of k nodes indexed by the subset $K \subseteq \{1, 2, \ldots, n\}$ of size $|K| = k$. Let $[G_{K,1}\ G_{K,2}]$ denote the $(k \times d)$ sub-matrix of $G = [G_1\ G_2]$ obtained by selecting the rows indexed by K, where $G_{K,1}$ and $G_{K,2}$ are the corresponding sub-matrices of G_1 and G_2 respectively. Let C_K denote the corresponding $(k \times d)$ sub-matrix of C. Then we can write:

$$\underbrace{C_K}_{k\times d} \;=\; \begin{bmatrix} \underbrace{G_{K,1}}_{k\times k} & \underbrace{G_{K,2}}_{k\times(d-k)} \end{bmatrix} \begin{bmatrix} S & V \\ V^T & \mathbf{0} \end{bmatrix} \;:=\; \begin{bmatrix} \underbrace{C_{K,1}}_{k\times k} & \underbrace{C_{K,2}}_{k\times(d-k)} \end{bmatrix},$$

so that

$$C_{K,1} \;=\; G_{K,1}S + G_{K,2}V^T \text{ and } C_{K,2} \;=\; G_{K,1}V.$$

During data recovery, both $C_{K,1}$ and $C_{K,2}$ are accessible. As any $(k \times k)$ sub-matrix of G_1 is non-singular by design, in particular, the sub-matrix $G_{K,1}$ is non-singular. This allows us to recover the matrix V from:

$$V \;=\; (G_{K,1})^{-1}C_{K,2}.$$

Having recovered V, we can then recover S using:

$$S \;=\; (G_{K,1})^{-1}[C_{K,1} - G_{K,2}V^T].$$

With this all B message symbols have been recovered.

Node Repair: Assume that node f has failed and let the helper nodes be indexed by the subset $D \subseteq \{1, 2, \ldots, n\}$ of size $|D| = d$. Let C_D, G_D denote the sub-matrices of C, G respectively, obtained by selecting the rows indexed by D. We then have:

$$\underbrace{C_D}_{d\times d} = \underbrace{G_D}_{d\times d}\begin{bmatrix} S & V \\ V^T & \mathbf{0} \end{bmatrix}.$$

Let $\underline{g}_f^T$ denote the fth row of G. The node repair process can be explained in three steps.

- Step 1: Each helper node-i, $i \in D$, computes $\underline{c}_i^T \underline{g}_f$ and transmits the resultant symbol to the replacement node. At the end of Step 1, the vector $C_D \underline{g}_f$ is available at the replacement node.
- Step 2: Since any $(d \times d)$ sub-matrix of G is non-singular, the replacement node can then compute:

$$(G_D)^{-1} C_D \underline{g}_f = \begin{bmatrix} S & V \\ V^T & \mathbf{0} \end{bmatrix} \underline{g}_f.$$

- Step 3: By taking the transpose, the replacement node obtains:

$$\left(\begin{bmatrix} S & V \\ V^T & \mathbf{0} \end{bmatrix} \underline{g}_f\right)^T = \underline{g}_f^T \begin{bmatrix} S & V \\ V^T & \mathbf{0} \end{bmatrix} = \underline{c}_f^T,$$

  and in this way, the contents of the failed node have been recovered.

Notes

1. Fractional repetition codes: Fractional repetition codes [59] may be regarded as generalizing the polygonal MBR construction. In a fractional repetition code, the underlying scalar code symbols are obtained by replicating an MDS code $\rho \geq 2$ times. However, unlike in the case of an MBR code, for the repair of each node, only a specific set of $d \leq n - 1$ helper nodes is guaranteed to be able to help in node repair. For this reason, fractional repetition codes are said to have table-based repair. Fractional repetition codes are discussed in greater detail in Section 9.2.

2. Binary MBR codes: There exist MBR codes over the binary field $\mathbb{F}_2$ with $\beta = 1$ if the parameters $\{n, k, d\}$ satisfy any of the following conditions (i) $k = d - 1 = n - 2$ (ii) $k = d = n - 2$ and (iii) $k = d - 1 = n - 3$. Details can be found in [132], [179].

Open Problem 1. Determine the smallest possible field size q of an MBR code having given parameters $\{n, k, d, \beta\}$.

5

MSR Codes

Among the class of RGCs, MSR codes have received the greatest attention for reasons that include the fact that MSR codes are MDS codes, have storage overhead that can be made as small as desired, and have been challenging to construct. An MSR code with parameters (n, k, d, α) has file size $B = k\alpha$ and repair bandwidth $\beta = \frac{\alpha}{d-k+1}$. MSR codes can also be viewed as vector MDS codes that incur the least-possible repair-bandwidth for the repair of a failed node.

While only β symbols are passed on to the replacement of a failed node by each of the d helper nodes, the number of symbols accessed by the helper node in order to generate these β symbols could be significantly larger than β. There is interest in practice, in the subclass of MSR codes having the property that the number of scalar symbols accessed at each helper node is also equal to the number β of symbols that are passed on for node repair. Such MSR codes are termed as optimal-access MSR codes.

An early construction of an MSR code with parameters (n, k, d) and (α, β) satisfying $d = (n-1) \geq 2k-1$, $\beta = 1$, can be found in [227] and is briefly discussed in the notes subsection. A detailed description of three constructions of an MSR code is presented in the present section, along

Table 5.1: Explicit MSR code constructions described in this section. Here $r = (n-k)$ and $s = (d-k+1)$. The * in the last row is to indicate that Small-d MSR codes have lowest possible sub-packetization level under the assumption of helper-set-independent repair, see Section 5.4.

MSR code	Parameters	Field size	Attributes
PM -MSR [185]	$d \geq 2(k-1)$, $\alpha = s$	$n\alpha$	low-rate
Diagonal MSR[255]	all d, $\alpha = s^n$	sn	optimal-update
CL-MSR [137], [205], [256]	$d = n-1$, $\alpha = r^{\lceil \frac{n}{r} \rceil}$	n	optimal-access with minimum α
Small-d MSR [239]	$d \in \{k+1, k+2, k+3\}$, $\alpha = s^{\lceil \frac{n}{s} \rceil}$	$O(n)$	optimal-access with minimum* α

with a brief description of a fourth MSR code. The first construction is the product-matrix MSR (PM-MSR) code [185], i.e., the MSR code constructed using a product-matrix framework as was the case with the PM-MBR code. PM-MSR codes, like PM-MBR codes, have smallest level of sub-packetization possible of an RGC, corresponding to setting parameter $\beta = 1$. This is followed by a description of the Diagonal MSR code [255] construction, a construction which yields MSR codes for all (n, k, d) parameter sets. The third construction presented is the coupled-layer MSR (CL-MSR) code [137], [205], [256]. The CL-MSR code is an optimal-access MSR code with parameter $d = (n-1)$ that turns out to have least-possible sub-packetization level of an optimal-access MSR code. Following the three detailed descriptions, we provide a brief summary of the attributes of a fourth MSR code construction, termed the Small-d MSR code construction [239]. The Small-d MSR code construction yields MSR codes for small values of d that have the optimal-access property. Table 5.1 presents an overview of the four code constructions. Brief discussions of other constructions of MSR codes can be found in the notes subsection.

5.1 Product-Matrix MSR Code

The product-matrix MSR (PM-MSR) construction by Rashmi *et al.* [185] yields MSR codes with parameters satisfying $d \geq 2(k-1)$ and $\beta = 1$. We begin with the case $d = 2(k-1)$ and then show how the general $d \geq 2(k-1)$ code can be constructed by appropriately shortening the $d = 2(k-1)$ code. The process of code shortening is explained in Section 5.1.3.

Given that we are operating at the MSR point with $d = 2(k-1)$ and $\beta = 1$, it follows that the resultant MSR code will have parameter set given by:

$$\{(n, k, d = 2(k-1)),\ (\alpha = (k-1), \beta = 1),\ B = k\alpha\}.$$

Note that $d = 2(k-1) = 2\alpha$. Let M be a $(2\alpha \times \alpha)$ matrix having the structure:

$$\underbrace{M}_{2\alpha\times\alpha} = \begin{bmatrix} \underbrace{S_1}_{\alpha\times\alpha} \\ \underbrace{S_2}_{\alpha\times\alpha} \end{bmatrix},$$

where the matrices S_1, S_2 are symmetric, of size $(\alpha \times \alpha)$. It follows that the total number of distinct symbols that can be contained in the matrix is given by $\alpha(\alpha+1) = \alpha k$ which is precisely the file size of the MSR code it is planned to construct. In the first step of the construction, the matrix M is populated with the $B = k\alpha$ message symbols.

Encoding is carried out using an $(n \times d)$ matrix J given by

$$\underbrace{J}_{n\times d} = \begin{bmatrix} \underbrace{G}_{n\times\alpha} & \underbrace{\Lambda G}_{n\times\alpha} \end{bmatrix},$$

where G is an $(n \times \alpha)$ matrix and Λ is an $(n \times n)$ diagonal matrix. The matrix J may be regarded as playing the role of generator matrix in the construction. The matrices G and Λ are required to be chosen such that the following properties hold:

- Every $(d \times d)$ sub-matrix of J is non-singular,
- Every $(\alpha \times \alpha)$ sub-matrix of G is non-singular,
- The n diagonal elements of Λ are distinct.

We now present a Vandermonde matrix J of the required form that meets all the above requirements. Let $\mathbb{F}_q$ be a finite field having size $q \geq n\alpha$. Let γ be a primitive element of $\mathbb{F}_q$, i.e., γ is a generator of the multiplicative group $\mathbb{F}_q^*$ of $\mathbb{F}_q$ and set $\theta_i = \gamma^{i-1}$, for all $1 \leq i \leq n$. Then

the Vandermonde matrix

$$J = \begin{bmatrix} 1 & \theta_1 & \theta_1^2 & \dots & \theta_1^{(d-1)} \\ 1 & \theta_2 & \theta_2^2 & \dots & \theta_2^{(d-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \theta_n & \theta_n^2 & \dots & \theta_n^{(d-1)} \end{bmatrix}$$

meets all the requirements. It is of the form $J = [G\ \Lambda G]$ where:

$$G = \begin{bmatrix} 1 & \theta_1 & \dots & \theta_1^{\alpha-1} \\ 1 & \theta_2 & \dots & \theta_2^{\alpha-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \theta_n & \dots & \theta_n^{\alpha-1} \end{bmatrix} \text{ and } \Lambda = \begin{bmatrix} \theta_1^\alpha & & & \\ & \theta_2^\alpha & & \\ & & \ddots & \\ & & & \theta_n^\alpha \end{bmatrix}.$$

The $(n \times \alpha)$ code matrix C is then given by

$$\underbrace{C}_{n\times\alpha} = \underbrace{J}_{n\times d}\underbrace{M}_{d\times\alpha} = \begin{bmatrix} G & \Lambda G \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}.$$

As in the case of the PM-MBR code, the ith node stores the α symbols contained in the ith row $\underline{c}_i$ of C. Let the ith row of G be denoted by $\underline{g}_i^T$ and let λ_i be the ith diagonal element of Λ.

Node Repair: Suppose node f has failed. The fth node stores the fth row of C given by

$$\underline{c}_f^T = \begin{bmatrix} \underline{g}_f^T & \lambda_f \underline{g}_f^T \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \underline{g}_f^T S_1 + \lambda_f \underline{g}_f^T S_2.$$

Our goal in node repair, is to recreate this vector using helper data. Let $D \subseteq \{1, 2, \dots, n\} \setminus \{f\}$, with $|D| = d$, be the indices of the d helper nodes. Let J_D be the sub-matrix of J obtained by selecting the $d = 2\alpha$ rows of J whose indices lie in D. Let C_D be the sub-matrix of C containing rows with indices lying in D. Then

$$\underbrace{C_D}_{d\times\alpha} = \underbrace{J_D}_{d\times d} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix},$$

and the symbols of C_D are precisely the contents of the d helper nodes.

Step 1: The helper node i sends $\underline{c}_i^T \underline{g}_f$ to the replacement node. Aggregating repair information from all helper nodes, the replacement node obtains $C_D \underline{g}_f$.

Step 2: The replacement node then computes

$$(J_D)^{-1} C_D \underline{g}_f \;=\; \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \underline{g}_f \;=\; \begin{bmatrix} S_1 \underline{g}_f \\ S_2 \underline{g}_f \end{bmatrix}$$

and thus recovers $S_1 \underline{g}_f$ and $S_2 \underline{g}_f$.

Step 3: Since S_1 and S_2 are symmetric, the replacement node can then carry out the computation

$$(S_1 \underline{g}_f)^T + \lambda_f (S_2 \underline{g}_f)^T \;=\; \underline{g}_f^T S_1 + \lambda_f \underline{g}_f^T S_2 \;=\; \underline{c}_f^T,$$

to recover the content of the failed node.

Data Collection: Let the subset $K \subseteq \{1, 2, \ldots, n\}$, with $|K| = k$ represent the indices of the nodes whose contents are to be used to recover the data file. Let $J_K = [G_K \; (\Lambda G)_K]$ be the $(k \times d)$ sub-matrix of J obtained by picking rows with indices in K. Similarly, let C_K denote the $(k \times \alpha)$ sub-matrix of C corresponding to K given by

$$\underbrace{C_K}_{k \times \alpha} \;=\; \begin{bmatrix} \underbrace{G_K}_{k \times \alpha} & \underbrace{(\Lambda G)_K}_{k \times \alpha} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}.$$

Our goal is to recover S_1 and S_2. Let Λ_K be the $(k \times k)$ sub-matrix of Λ consisting of rows and columns whose index lies in K. As Λ is a diagonal matrix, it can be easily verified that $(\Lambda G)_K = \Lambda_K G_K$. It follows that

$$C_K \;=\; \begin{bmatrix} G_K & \Lambda_K G_K \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \;=\; G_K S_1 + \Lambda_K G_K S_2.$$

Next, compute

$$C_K G_K^T \;=\; G_K S_1 G_K^T + \Lambda_K G_K S_2 G_K^T \;=\; P + \Lambda_K Q \;=\; A,$$

where $P = G_K S_1 G_K^T$ and $Q = G_K S_2 G_K^T$. Since S_1 and S_2 are symmetric matrices, it follows that P and Q are also symmetric, and of size $(k \times k)$. Let a_{ij}, p_{ij} and q_{ij} be the (i, j)-th entries of A, P and Q respectively. Then,

$$a_{ij} = p_{ij} + \lambda_i q_{ij} \text{ and } a_{ji} = p_{ji} + \lambda_j q_{ji}.$$

Since $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$, we have

$$\begin{bmatrix} a_{ij} \\ a_{ji} \end{bmatrix} = \begin{bmatrix} 1 & \lambda_i \\ 1 & \lambda_j \end{bmatrix} \begin{bmatrix} p_{ij} \\ q_{ij} \end{bmatrix}.$$

For $i \neq j$, $\lambda_i \neq \lambda_j$ and we can solve for p_{ij} and q_{ij}. Thus, we have access to all the off-diagonal elements of both P and Q.

Let $\underline{e}$ be such that $\underline{e}^T G_K = \underline{0}^T$. Such an $\underline{e}$ can be found since G_K is of size $(\alpha + 1) \times \alpha$. Moreover, all the entries of $\underline{e}$ must be non-zero since any α rows of G, and hence of G_K, are required to be linearly independent. Therefore,

$$\underline{e}^T P \;=\; \underline{e}^T G_K S_1 G_K^T \;=\; \underline{0}^T.$$

In each of the k equations here there is only one unknown namely, the diagonal element p_{ii}. In this way, the diagonal elements of P can be recovered. The diagonal entries of Q can be recovered in identical fashion. Given P and Q, the matrices S_1, S_2 can be recovered in straightforward fashion from $P = G_K S_1 G_K^T$ and $Q = G_K S_2 G_K^T$. This completes the data-collection process.

5.1.1 Extension to the Case $d > 2(k-1)$

The extension we present here is by shortening of an MSR code, which is the method adopted in [185] to provide constructions for $d \geq 2k - 1$. An alternative approach, that also makes use of the PM framework, can be found in [143], that directly yields MSR constructions for any $d \geq 2k - 1$ without need of shortening. We begin by explaining the concept of shortening as it applies to scalar linear codes.

5.1.2 Shortening of a Scalar Linear Code

Let $\mathcal{C}$ be an $[n, k]$ systematic, linear scalar code. Then $\mathcal{C}$ has a generator matrix of the form

$$G \;=\; \left[\underbrace{I_k}_{k \times k} \quad \underbrace{P}_{k \times (n-k)} \right],$$

and the first k code symbols in any codeword of $\mathcal{C}$ are message symbols. Let $\mathcal{C}_1 \subseteq \mathcal{C}$ be the subcode of $\mathcal{C}$ corresponding to code symbol $c_1 = 0$,

i.e., $(c_1, \ldots, c_n) \in \mathcal{C}_1 \implies c_1 = 0$. Then $\mathcal{C}_1$ is an $[n, k-1]$ code. The first code symbol in all the codewords in $\mathcal{C}_1$ is zero. Deleting this symbol leads us to the code $\mathcal{C}_1'$, which is an $[n-1, k-1]$ code. We will refer to the code $\mathcal{C}_1'$ as the code obtained by shortening the code $\mathcal{C}$ on or with respect to, the first coordinate. If $S \subseteq \{1, \ldots, k\}$ is a subset of size $1 \leq |S| = s \leq (k-1)$, then it is clear that through repeated shortening, we can construct an $[n-s, k-s]$ code $\mathcal{C}_S'$ by considering the subcode of $\mathcal{C}$ that is obtained by setting s message symbols to zero.

5.1.3 Shortening of a Linear MSR Code

Next, let $\mathcal{C}$ be an $\{(n, k, d), (\alpha, \beta), \mathbb{F}_q, B\}$ linear MSR code, i.e., an MSR code that is linear as an RGC (see Section 3). Thus, the $n\alpha$ symbols stored across the n storage nodes are linear functions of the B message symbols.

The size of the data file equals $k\alpha$ which is precisely the number of $\mathbb{F}_q$ symbols contained in any set of k nodes. Clearly, by making an appropriate linear transformation of code symbols, we may assume that the contents of the first k nodes $\{\underline{c}_i\}_{i=1}^{k}$ are precisely the B message symbols. Consider the subcode $\mathcal{C}_1$ of $\mathcal{C}$ that corresponds to the contents of the first s, $1 \leq s \leq (k-1)$ nodes being equal to zero. It can be verified that if one deletes or removes these nodes, one will be left with an MSR code having parameters:

$$\{(n-s, k-s, d-s),\ (\alpha, \beta),\ B = \alpha(k-s)\}.$$

5.1.4 Extending Parameter Set of the PM-MSR Code

Suppose now it is desired to construct an MSR code having parameters

$$\{(n, k, d = 2(k-1)+s),\ (\alpha = d-k+1, \beta = 1),\ B = \alpha k\},$$

one begins with the construction of a PM-MSR code having parameters

$$\{(n+s, k+s, 2(k+s-1)),\ (\alpha = k+s-1, \beta = 1),\ B = \alpha(k+s)\}.$$

Shortening with respect to s nodes then converts this into an MSR code having parameters

$$\{(n, k, d = 2(k-1)+s),\ (\alpha = k+s-1, \beta = 1),\ B = \alpha k\}.$$

In this way, the PM-MSR construction can be made to realize MSR codes having parameters $d \geq 2(k-1)$.

5.1.5 Rate of the PM-MSR Code

PM-MSR codes exist only for $d \geq 2(k-1)$ from which it follows that $n \geq d+1 \geq 2k-1$. The rate R of the code then satisfies:

$$R \;=\; \frac{k}{n} \;\leq\; \frac{k}{2k-1} \;=\; \frac{1}{2} + \frac{1}{2(2k-1)},$$

which is just a little over half. This relatively low rate is a drawback of the PM-MSR code.

5.2 Diagonal-Matrix-Based MSR Code

In this subsection we describe a construction for a linear MSR code family due to Ye and Barg [255]. The construction is explicit, employs field size that is linear in the block length n, and is able to generate MSR codes for any parameter set (n,k,d). The sub-packetization level $\alpha = s^n$ with $s \triangleq (d-k+1)$ is, however, exponential in the parameter n. We will refer to these codes here as the Diagonal MSR construction since the construction employs diagonal matrices.

The construction can be described in terms of an $(r\alpha \times n\alpha)$ p-c matrix H of the form:

$$H \;=\; \begin{bmatrix} I & I & \cdots & I \\ A_1 & A_2 & \cdots & A_n \\ \vdots & \vdots & \ddots & \vdots \\ A_1^{r-1} & A_2^{r-1} & \cdots & A_n^{r-1} \end{bmatrix}, \tag{5.1}$$

where $r = (n-k)$ and where each sub-matrix A_i is a diagonal $(\alpha \times \alpha)$ matrix over $\mathbb{F}_q$. Thus to fully specify the respective MSR code, it suffices to identify the matrices $\{A_i\}$.

The parameters of the Diagonal MSR code are of the form:

$$\{(n,k,d),\ (\alpha = s^n, \beta = s^{n-1}),\ B = \alpha k,\ \mathbb{F}_q\},$$

with field-size requirement given by $q \geq sn$. In the construction, each sub-matrix A_i, for $i \in [n]$, is a diagonal matrix taking on the form:

$$A_i \quad = \quad \sum_{a \in [\alpha]} \lambda_{i,a_i} e_a e_a^T,$$

where $(a_1, \cdots, a_n) \in \mathbb{Z}_s^n$ represents a base-s expansion of $(a-1)$, i.e.,

$$(a-1) = \sum_{i=1}^{n} a_i s^{i-1}$$

and the vectors $e_a \in \mathbb{F}_q^\alpha$ are unit vectors such that the a-th element of e_a is 1 and all other elements are zero. Thus, the matrix $e_a e_a^T$ is an $(\alpha \times \alpha)$ diagonal matrix having a 1 in the a-th row and a-th column and zeros everywhere else. The elements $\{\lambda_{i,u} \mid i \in [n], u \in [0, s-1]\}$ are chosen to be distinct and hence form a subset of $\mathbb{F}_q$ of size $\geq ns$.

Thus the ith matrix A_i is an $(\alpha \times \alpha)$ matrix, whose diagonal elements are indexed by the variable a, where a takes on values in the set $[s^n] = [\alpha]$. The 'a'th diagonal element equals λ_{i,a_i}, and thus is a function of i and the ith component a_i of a.

Let $\underline{c} = (\underline{c}_1^T, \cdots, \underline{c}_n^T)^T$ be a codeword in the Diagonal MSR code, where $\underline{c}_i = (c_i(1), \cdots, c_i(\alpha))^T \in \mathbb{F}_q^\alpha$ is stored in node $i \in [n]$. Then,

$$\begin{aligned}
H\underline{c} \quad &= \quad \underline{0}, \\
\Leftrightarrow \sum_{i=1}^{n} A_i^j \underline{c}_i \quad &= \quad \underline{0} \text{ for all } j \in [0, r-1], \\
\Leftrightarrow \sum_{i=1}^{n} \sum_{a \in [\alpha]} \lambda_{i,a_i}^j e_a e_a^T \underline{c}_i \quad &= \quad \underline{0} \text{ for all } j \in [0, r-1], \\
\Leftrightarrow \sum_{i=1}^{n} \lambda_{i,a_i}^j c_i(a) \quad &= \quad 0 \text{ for all } j \in [0, r-1],\ a \in [\alpha]. \quad (5.2)
\end{aligned}$$

It follows that the $r\alpha$ equations shown in (5.2) characterize the Diagonal MSR code. We will refer to the p-c equation appearing in (5.2), as the (j, a)-th parity-check.

Data Collection: The data collection property can be established by showing that the Diagonal MSR code can recover from any $r = (n-k)$

erasures. Let the set of node indices corresponding to the r erasures be denoted by $E \subseteq [n]$, $|E| = r$. Then the equation (5.2) reduces to:

$$\sum_{i \in E} \lambda_{i,a_i}^{j} c_i(a) \;=\; \kappa^*, \quad j \in [0, r-1], \; a \in [\alpha],$$

where κ^* denotes a known quantity that can be computed from the contents of the unerased nodes. As the set of $\{\lambda_{i,a_i} \mid i \in E\}$ are distinct for every $a \in [s^n]$, we can recover $\{c_i(a) \mid i \in E, a \in [s^n]\}$, thereby recovering all the erased symbols.

Node Repair: Let $i_0 \in [n]$ be the index of the failed node that needs to be repaired, and let the subset $D \subseteq [n] \setminus \{i_0\}$ of size $|D| = d$ denote the indices of the d helper nodes. The ith helper node for $i \in D$ sends the following $\beta = s^{n-1}$ symbols as helper information:

$$\left\{ h_{i,i_0}(a) = \sum_{u=0}^{s-1} c_i(a(i_0, u)) \;\; \middle| \;\; a \in [s^n], a_{i_0} = 0 \right\},$$

where $a(i_0, u)$ is the integer, whose s-ary representation

$$(a_1, \cdots, a_{i_0-1}, u, a_{i_0+1}, \cdots, a_n),$$

is the same as that of a except that the i_0th component, a_{i_0}, is replaced by u. Equivalently, $a(i_0, u) = a - a_{i_0} s^{i_0 - 1} + u s^{i_0 - 1}$. Since all α symbols contained in a helper node are accessed in order to generate the β helper symbols, it follows that the Diagonal MSR code does not have the optimal-access MSR property.

Focusing on the $(j, a(i_0, u))$-th p-c equation for $a \in [s^n]$ such that $a_{i_0} = 0$ we obtain:

$$\lambda_{i_0,u}^{j} c_{i_0}(a(i_0, u)) \;=\; - \sum_{i \in [n] \setminus \{i_0\}} \lambda_{i,a_i}^{j} c_i(a(i_0, u)), \text{ all } u \in [0, s-1].$$

Summing over u, we obtain:

$$\begin{aligned} \sum_{u=0}^{s-1} \lambda_{i_0,u}^{j} c_{i_0}(a(i_0, u)) \;&=\; - \sum_{u=0}^{s-1} \sum_{i \in [n] \setminus \{i_0\}} \lambda_{i,a_i}^{j} c_i(a(i_0, u)) \\ &=\; - \sum_{i \in [n] \setminus \{i_0\}} \lambda_{i,a_i}^{j} h_{i,i_0}(a). \end{aligned}$$

Spelling out these p-c equations for all $j \in [0, r-1]$ in matrix form, we obtain:

$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_{i_0,0} & \lambda_{i_0,1} & \cdots & \lambda_{i_0,s-1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{i_0,0}^{r-1} & \lambda_{i_0,1}^{r-1} & \cdots & \lambda_{i_0,s-1}^{r-1} \end{bmatrix}}_{V_{i_0}} \begin{bmatrix} c_{i_0}(a(i_0,0)) \\ c_{i_0}(a(i_0,1)) \\ \vdots \\ c_{i_0}(a(i_0,s-1)) \end{bmatrix}$$

$$= -\underbrace{\begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ \lambda_{1,a_1} & \cdots & \lambda_{i_0-1,a_{i_0-1}} & \lambda_{i_0+1,a_{i_0+1}} & \cdots & \lambda_{n,a_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{1,a_1}^{r-1} & \cdots & \lambda_{i_0-1,a_{i_0-1}}^{r-1} & \lambda_{i_0+1,a_{i_0+1}}^{r-1} & \cdots & \lambda_{n,a_n}^{r-1} \end{bmatrix}}_{L_{i_0}} \begin{bmatrix} h_{1,i_0}(a) \\ \vdots \\ h_{i_0-1,i_0}(a) \\ h_{i_0+1,i_0}(a) \\ \vdots \\ h_{n,i_0}(a) \end{bmatrix}. \quad (5.3)$$

Case $d = (n-1)$: For the case when $d = n-1$, $s = d-k+1 = r$ and V_{i_0} is an $(r \times r)$ Vandermonde matrix and L_{i_0} is an $(r \times n-1)$ Vandermonde matrix. Also, the symbols in the RHS of the above equation given by $\{h_{i,i_0}(a) \mid i \in [n] \setminus \{i_0\}\}$ are known. Therefore by the invertibility of V_{i_0} we can recover failed node symbols

$$\{c_{i_0}(a(i_0,u)) \mid u \in [0, s-1]\}.$$

By varying $a \in [s^n]$ such that $a_{i_0} = 0$, we can recover all the failed node symbols:

$$\{c_{i_0}(a(i_0,u)) \mid u \in [0,s-1], a \in [s^n], a_{i_0} = 0\} = \{c_{i_0}(a) \mid a \in [s^n]\}.$$

Case $d < (n-1)$: In this case, V_{i_0} is an $(r \times s)$ Vandermonde matrix and L_{i_0} is an $(r \times n-1)$ Vandermonde matrix. We will show that all failed symbols can be recovered by establishing that any d symbols of $\{h_{i,i_0}(a) \mid i \in [n] \setminus \{i_0\}\}$, are enough to recover the remaining $n-1-d = r-s$ symbols. This will be done by proving that $\{h_{i,i_0}(a) \mid i \in [n] \setminus \{i_0\}\}$ are code symbols of an $[n-1, d]$ MDS code.

We start by defining an $((r-s) \times r)$ matrix N_{i_0} with row vectors lying in the left null space of V_{i_0}. The ith row of N_{i_0}, $N_{i_0}(i,:)$ is defined as shown below for any $i \in [0, r-s-1]$:

$$N_{i_0}(i,:) = \begin{matrix} 0 & \cdots & i-1 & i & i+1 & \cdots & i+s-1 & i+s & i+s+1 & \cdots & r-1 \end{matrix}$$
$$\begin{bmatrix} 0 & \cdots & 0 & f_0 & f_1 & \cdots & f_{s-1} & f_s & 0 & \cdots & 0 \end{bmatrix}.$$

where $f(x) = \sum_{i=0}^{s} f_i x^i = \prod_{u=0}^{s-1}(x - \lambda_{i_0,u})$. Then we have:

$$N_{i_0}V_{i_0} = \begin{bmatrix} f(\lambda_{i_0,0}) & f(\lambda_{i_0,1}) & \cdots & f(\lambda_{i_0,s-1}) \\ \lambda_{i_0,0}f(\lambda_{i_0,0}) & \lambda_{i_0,1}f(\lambda_{i_0,1}) & \cdots & \lambda_{i_0,s-1}f(\lambda_{i_0,s-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{i_0,0}^{r-s-1}f(\lambda_{i_0,0}) & \lambda_{i_0,1}^{r-s-1}f(\lambda_{i_0,1}) & \cdots & \lambda_{i_0,s-1}^{r-s-1}f(\lambda_{i_0,s-1}) \end{bmatrix} = \underbrace{\mathbf{0}}_{((r-s)\times s)}, \tag{5.4}$$

and

$$N_{i_0}L_{i_0} = \begin{bmatrix} f(\lambda_{1,a_1}) & \cdots & f(\lambda_{i_0-1,a_{i_0-1}}) & f(\lambda_{i_0+1,a_{i_0+1}}) & \cdots & f(\lambda_{n,a_n}) \\ \lambda_{1,a_1}f(\lambda_{1,a_1}) & \cdots & \lambda_{i_0-1,a_{i_0-1}}f(\lambda_{i_0-1,a_{i_0-1}}) & \lambda_{i_0+1,a_{i_0+1}}f(\lambda_{i_0+1,a_{i_0+1}}) & \cdots & \lambda_{n,a_n}f(\lambda_{n,a_n}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{1,a_1}^{r-s-1}f(\lambda_{1,a_1}) & \cdots & \lambda_{i_0-1,a_{i_0-1}}^{r-s-1}f(\lambda_{i_0-1,a_{i_0-1}}) & \lambda_{i_0+1,a_{i_0+1}}^{r-s-1}f(\lambda_{i_0+1,a_{i_0+1}}) & \cdots & \lambda_{n,a_n}^{r-s-1}f(\lambda_{n,a_n}) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ \lambda_{1,a_1} & \cdots & \lambda_{i_0-1,a_{i_0-1}} & \lambda_{i_0+1,a_{i_0+1}} & \cdots & \lambda_{n,a_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{1,a_1}^{r-s-1} & \cdots & \lambda_{i_0-1,a_{i_0-1}}^{r-s-1} & \lambda_{i_0+1,a_{i_0+1}}^{r-s-1} & \cdots & \lambda_{n,a_n}^{r-s-1} \end{bmatrix}$$

$$\times \begin{bmatrix} f(\lambda_{1,a_1}) & & & & & \\ & \ddots & & & & \\ & & f(\lambda_{i_0-1,a_{i_0-1}}) & & & \\ & & & f(\lambda_{i_0+1,a_{i_0+1}}) & & \\ & & & & \ddots & \\ & & & & & f(n,a_n) \end{bmatrix}.$$

Notice that $N_{i_0}L_{i_0}$ is p-c matrix of an $[n-1, n-1-(r-s) = d]$ generalized Reed-Solomon (GRS) code as $f(\lambda_{i,a_i}) \neq 0$ for all $i \neq i_0$. From equations (5.3) and (5.4) we get,

$$N_{i_0}L_{i_0} \begin{bmatrix} h_{1,i_0}(a) \\ \vdots \\ h_{i_0-1,i_0}(a) \\ h_{i_0+1,i_0}(a) \\ \vdots \\ h_{n,i_0}(a) \end{bmatrix} = 0.$$

By the GRS property we can recover all the $n-1$ symbols in $\{h_{i,i_0}(a) \mid i \in [n]\backslash\{i_0\}\}$ from any d-symbol subset. This implies that the symbols in

the RHS of the equation (5.3) are known. Therefore by the invertibility of the sub-matrix of V_{i_0} comprising of the first s rows of V_{i_0}, we can recover the failed node symbols

$$\{c_{i_0}(a(i_0, u)) \mid u \in [0, s-1]\}.$$

By varying $a \in [s^n]$ such that $a_{i_0} = 0$, we can recover all the failed node symbols:

$$\{c_{i_0}(a(i_0, u)) \mid u \in [0, s-1], a \in [s^n], a_{i_0} = 0\} = \{c_{i_0}(a) \mid a \in [s^n]\}.$$

Remark 4. Diagonal MSR codes turn out to also satisfy the optimal-update property where, to update a single symbol out of the α symbols in a systematic node, one is required to update only $(n-k)$ parity symbols.

An extension of the Diagonal MSR code that has the (h, d) optimal-repair property for any $h \in [2, n-k]$, $d \in [k, n-h]$ appears in [255]. By (h, d) optimal-repair property is meant, the recovery of h erasures by downloading

$$\frac{\alpha h}{d-k+h}$$

symbols each from d helper nodes, which is the minimal repair bandwidth possible for MDS codes [31]. The sub-packetization level of these extended codes is of the form s^n where $s = \text{lcm}(2, 3, \cdots, n-k)$. The h node repair discussed here assumes a centralized repair setting whereas an alternate, cooperative repair approach is discussed in Section 9.3.

5.3 Coupled-Layer MSR Code

In [256], Ye and Barg presented an explicit construction of a high-rate, optimal-access MSR code with $\alpha = r^{\lceil \frac{n}{r} \rceil}$, field size no larger than $r\lceil \frac{n}{r} \rceil$, and $d = (n-1)$, where $r = n-k$. Essentially the same construction was independently rediscovered by Sasidharan *et al.* [205] two months later, from a different coupled-layer perspective, where layers of an arbitrary MDS codes are coupled by a simple pairwise-coupling transform to yield an MSR code.

Just prior to the appearance of these two papers, in an earlier version of [137], Li *et al.* show how a systematic MSR code can be converted into an MSR code by increasing the sub-packetization level by a factor of r using a pairwise-symbol transformation. This result is then extended in [137] to a technique that takes an MDS code, increases sub-packetization level by a factor of r and converts it into a code in which the optimal repair of r nodes can be carried out. By applying this transform repeatedly $\lceil \frac{n}{r} \rceil$ times, it is shown that any scalar MDS code can be transformed into an MSR code. It turns out that the three papers [137], [205], [256], either explicitly or implicitly, employed as a key part of the construction, essentially the same pairwise-coupling transform.

In this subsection, we present this optimal-access MSR code construction contained in [137], [205], [256] from the coupled-layer perspective appearing in [205]. We will refer to the resultant MSR code as the coupled-layer MSR (CL-MSR) code[1]. This code has the additional attribute of having the lowest-possible level of sub-packetization of any linear, optimal-access MSR code, provided $n \neq 1 \pmod{r}$, as it attains a lower bound on sub-packetization level of a linear, optimal-access MSR code, see Section 8.2.

A CL-MSR code has parameter set of the form

$$\{(n = rt,\ k = r(t-1),\ d = n-1), (\alpha = r^t,\ \beta = r^{t-1}), B = \alpha k, \mathbb{F}_q\},$$

and as can be seen, code parameters are a function of two integer-valued variables, namely $r \geq 1$, and $t \geq 2$. The field-size requirement is given by $q \geq n$. The rate R of this code is hence given by $R = \frac{r(t-1)}{rt} = \frac{t-1}{t}$ and can be made arbitrarily close to 1 by making t large enough. The principal steps in the construction of an CL-MSR code are the following:

(a) An $[n = rt,\ k = r(t-1)]$ scalar MDS code $\mathcal{C}_{\text{MDS}}$ is first selected,

(b) The $n = rt$ code symbols of each codeword in $\mathcal{C}_{\text{MDS}}$ are arranged so as to form a two-dimensional $(r \times t)$ array,

[1] Vajha *et al.* [240] present an implementation and evaluation of the coupled-layer MSR code in the Ceph distributed storage system. In the paper, the Coupled-LAYer code is given the acronym, the Clay code. The implementation in Ceph is described in Section 17.

(c) Each codeword belonging to the CL-MSR code $\mathcal{C}$, is uniquely associated to a set of $\alpha = r^t$ codewords drawn from $\mathcal{C}_{\text{MDS}}$ that are not necessarily distinct,

(d) The α codewords from $\mathcal{C}_{\text{MDS}}$ are vertically stacked so as to form a data cube, which we will refer to as the uncoupled data cube,

(e) The symbols within the uncoupled data cube are transformed using a simple, linear pairwise-symbol transformation that replaces selected pairs of symbols over $\mathbb{F}_q$ contained within the uncoupled data cube, by their transformed versions. The data cube obtained via this transformation is called the coupled data cube.

Let

$$\{B(x, y, \underline{z}) \;\mid\; (x, y) \in \mathbb{Z}_r \times \mathbb{Z}_t,\ \underline{z} \in \mathbb{Z}_r^t\}$$

denote the $n\alpha = (r \times t \times r^t)$ symbols of the uncoupled data cube (see Fig. 5.1). Then, for fixed value $\underline{z}_0$ of the planar (or horizontal-layer) index $\underline{z}$, the $n = rt$ symbols $\{B(x, y, \underline{z}_0) \mid (x, y) \in \mathbb{Z}_r \times \mathbb{Z}_t\}$ constitute the n code symbols of a codeword from $\mathcal{C}_{\text{MDS}}$.

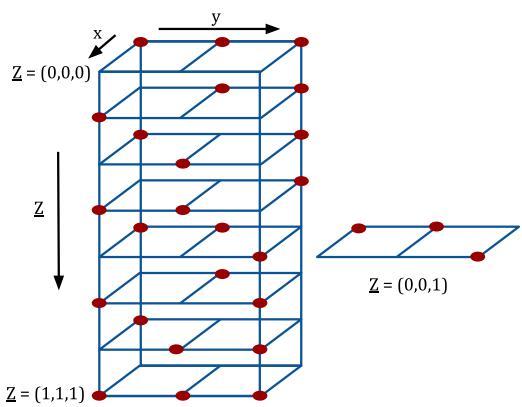

Figure 5.1: An example uncoupled data cube for the case ($r = 2$, $t = 3$). As can be seen, the location of the red dots within a plane, provides a pictorial representation of the index $\underline{z}$ associated to the plane.

Let H be an $((n - k) \times n)$ p-c matrix of the scalar code $\mathcal{C}_{\text{MDS}}$. Let $\theta_{\ell,(x,y)}$ denote the element of H lying in the ℓ-th row, and (x, y)-th

column. The symbols in the uncoupled data cube then satisfy the equations:

$$\sum_{(x,y)\in\mathbb{Z}_r\times\mathbb{Z}_t} \theta_{\ell,(x,y)} B(x,y;\underline{z}) \;=\; 0, \tag{5.5}$$

for all $\ell \in [0, n-k-1]$ and all $\underline{z} \in \mathbb{Z}_r^t$.

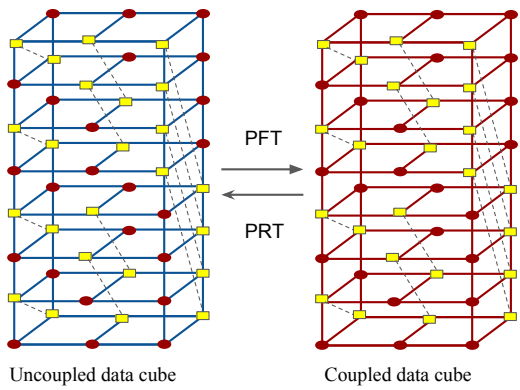

Figure 5.2: Paired symbols within either the uncoupled or coupled data cube are depicted using yellow rectangles connected by dotted lines. The pairwise forward transform (PFT) and pairwise reverse transform (PRT) are used to transform symbol-pairs between the two data cubes.

Next, the symbols in the uncoupled data cube $B(\cdot)$ are paired. The symbol $B(x, y, \underline{z})$ with $z_y \neq x$ is paired with the symbol $B(z_y, y, \underline{z}(y, x))$ where, we use the notation $\underline{z}(y, x)$ to denote the vector in which the y-th component of $\underline{z}$ is replaced by x:

$$\underline{z}(y,x) \;\triangleq\; (z_0, \cdots, z_{y-1}, x, z_{y+1}, \cdots, z_{t-1}).$$

The symbols $B(x, y, \underline{z})$ with $z_y = x$ remain unpaired. Equivalently, we may regard these symbols as fixed points in this pairing process, i.e., each symbol $B(x, y, \underline{z})$ with $z_y = x$ is paired with itself.
Next, let

$$\{A(x,y,\underline{z}) \mid (x,y) \in \mathbb{Z}_r \times \mathbb{Z}_t, \underline{z} \in \mathbb{Z}_r^t\},$$

denote the $n\alpha$ symbols of a second data cube, termed the coupled data cube. The contents of the coupled data cube will shortly be related to the contents of the uncoupled data cube, as depicted in Fig. 5.2. There is

an analogous pairing of symbols within the coupled data cube. Thus the symbol $A(x, y, \underline{z})$ with $z_y \neq x$ is paired with the symbol $A(z_y, y, \underline{z}(y, x))$ and the symbols $A(x, y, \underline{z})$ with $z_y = x$ are paired with themselves.

Let u be a nonzero element in the finite field $\mathbb{F}_q$, satisfying $u^2 \neq 1$. The symbols of the coupled data cube are derived from those of the uncoupled data cube via the following transformation:

$$\begin{aligned} \begin{bmatrix} A(x, y, \underline{z}) \\ A(z_y, y, \underline{z}(y, x)) \end{bmatrix} &= \begin{bmatrix} 1 & u \\ u & 1 \end{bmatrix}^{-1} \begin{bmatrix} B(x, y, \underline{z}) \\ B(z_y, y, \underline{z}(y, x)) \end{bmatrix}, \\ & \qquad \text{for } z_y \neq x, \\ A(x, y, \underline{z}) &= B(x, y, \underline{z}), \text{ for } z_y = x. \end{aligned} \tag{5.6}$$

We will refer to set of equations (5.6), as the pairwise forward transform (PFT). The mapping in the reverse direction, given by:

$$\begin{aligned} \begin{bmatrix} B(x, y, \underline{z}) \\ B(z_y, y, \underline{z}(y, x)) \end{bmatrix} &= \begin{bmatrix} 1 & u \\ u & 1 \end{bmatrix} \begin{bmatrix} A(x, y, \underline{z}) \\ A(z_y, y, \underline{z}(y, x)) \end{bmatrix}, \\ & \qquad \text{for } z_y \neq x, \\ B(x, y, \underline{z}) &= A(x, y, \underline{z}), \text{ for } z_y = x. \end{aligned} \tag{5.7}$$

will be referred to as the pairwise reverse transform (PRT).

Remark 5. (4-symbol MDS property) Note for the case $z_y \neq x$, we have that

$$\begin{aligned} &[A(x, y, \underline{z}) \;\; A(z_y, y, \underline{z}(y, x)) \; B(x, y, \underline{z}) \;\; B(z_y, y, \underline{z}(y, x))] \; = \\ &\qquad [A(x, y, \underline{z}) \;\; A(z_y, y, \underline{z}(y, x))] \begin{bmatrix} 1 & 0 & 1 & u \\ 0 & 1 & u & 1 \end{bmatrix}. \end{aligned}$$

Since any two columns of the (2×4) matrix appearing on the extreme right of the equation above are linearly independent, it follows that all four symbols

$$\{A(x, y, \underline{z}), \; A(z_y, y, \underline{z}(y, x)), \; B(x, y, \underline{z}), \; B(z_y, y, \underline{z}(y, x))\}$$

can be computed from knowledge of any two symbols from the 4-symbol set, i.e., the four symbols form a $[4, 2]$ MDS code.

In terms of physical storage, in the CL-MSR code, the $n = rt$ nodes are indexed by the pairs $(x, y) \in \mathbb{Z}_r \times \mathbb{Z}_t$ with node (x, y) storing the symbols

$$\{A(x, y, \underline{z}) \mid \underline{z} \in \mathbb{Z}_r^t\},$$

of the coupled data cube.

Substituting the PRT (5.7) into the p-c equations (5.5) of the CL-MSR code associated with the uncoupled data cube $B(\cdot)$, we obtain the equivalent p-c equations placed on the symbols of the coupled data cube $A(\cdot)$:

$$\sum_{(x,y)\in\mathbb{Z}_r\times\mathbb{Z}_t} \theta_{\ell,(x,y)} A(x, y, \underline{z}) + \sum_{y\in\mathbb{Z}_t} \sum_{x\neq z_y} u\theta_{\ell,(x,y)} A(z_y, y, \underline{z}(y, x)) \;=\; 0, \tag{5.8}$$

for all $\ell \in [0, n - k - 1]$ and $\underline{z} \in \mathbb{Z}_r^t$.

Node Repair: Let (x_0, y_0) be the failed node. Let us define a subset $\mathcal{P}(x_0, y_0)$ of $\beta = r^{t-1}$ "helper" planes given by

$$\mathcal{P}(x_0, y_0) \;\triangleq\; \left\{\underline{z} \in \mathbb{Z}_r^t \;\mid\; z_{y_0} = x_0\right\}.$$

To recover the r^t erased symbols $\{A(x_0, y_0, \underline{z}) \mid \underline{z} \in \mathbb{Z}_r^t\}$, each of the remaining nodes $(x, y) \neq (x_0, y_0)$ pass on the $\beta = r^{t-1}$ code symbols

$$\{A(x, y, \underline{z}) \mid \underline{z} \in \mathcal{P}(x_0, y_0)\}$$

that lie in these helper planes.

Consider a symbol $A(x, y, \underline{z})$ with $y \neq y_0$, lying in a helper plane $\underline{z} \in \mathcal{P}(x_0, y_0)$. The companion $A(z_y, y, \underline{z}(x, y))$ of $A(x, y, \underline{z})$ also lies in one of the helper planes since the y_0th component of $\underline{z}(x, y)$ is x_0. Thus both symbols are passed on to the replacement node.

For each $\underline{z} \in \mathcal{P}(x_0, y_0)$, we next rewrite (5.8) by placing the erased code symbols on the left and using the symbol κ^* to denote linear combinations of all the known helper information to the right. This leads to

$$\theta_{\ell,(x_0,y_0)} A(x_0, y_0, \underline{z}) + \sum_{x\neq x_0,\; x\in\mathbb{Z}_r} u\theta_{\ell,(x,y_0)} A(x_0, y_0, \underline{z}(y_0, x)) = \kappa^*, \tag{5.9}$$

$\forall \ell \in [0, r-1]$. As a result, for each fixed $\underline{z} \in \mathcal{P}(x_0, y_0)$, there are r unknowns and r equations from which the r unknowns

$$A(x_0, y_0, \underline{z}) \bigcup_{x \neq x_0,\ x \in \mathbb{Z}_r} A(x_0, y_0, \underline{z}(y_0, x))$$

can be recovered, since the choice of $\{\theta_{\ell,(x,y)}\}$ ensures the $(r \times r)$ coefficient matrix is non-singular. Repeating this process for all $\underline{z}$ in $\mathcal{P}(x_0, y_0)$ allows us to recover

$$\bigcup_{\underline{z} \in \mathcal{P}(x_0, y_0)} \left\{ A(x_0, y_0, \underline{z}) \bigcup_{x \neq x_0,\ x \in \mathbb{Z}_r} A(x_0, y_0, \underline{z}(y_0, x)) \right\} = \{A(x_0, y_0, \underline{z}) \mid \underline{z} \in \mathbb{Z}_r^t\},$$

which is precisely the set of all erased symbols.

Data Collection: To establish the data collection property, it is sufficient to show that the entire data file can be recovered in the presence of any $(n-k) = r$ node erasures. Let $\mathcal{E} \subseteq \mathbb{Z}_r \times \mathbb{Z}_t$ represent a fixed erasure pattern of size $|\mathcal{E}| = r$. We describe these erasures using a nomenclature that is plane dependent. For a given plane $\underline{z}$, erasures $(x, y) \in \mathcal{E}$ with $z_y = x$ are termed as serious erasures. The intersection score (IS) of a plane is then defined to be the number of serious erasures in the plane, and an illustrative example appears in Fig. 5.3.

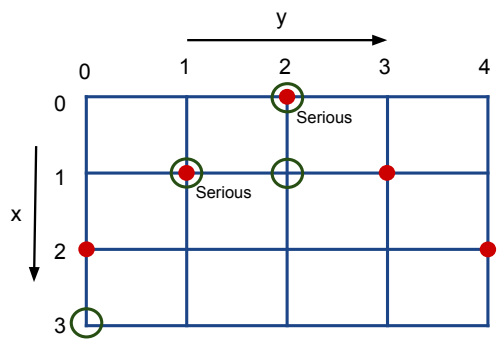

Figure 5.3: Shown above is plane with index $\underline{z} = (2, 1, 0, 1, 2)$ for the case $(r = 4, t = 5)$ where the four black circles indicate the four erasures. The intersection score of this plane is 2 as it has two serious erasures, corresponding to the circles enclosing red dots.

We present below, a sequential decoding algorithm corresponding to rounds $0, 1, 2, \cdots$ in that order, that is described in [205]. In the sequential algorithm, erased symbols in planes having intersection score IS $= i$ are decoded in the ith round, by making use of symbols recovered in prior rounds.

IS $= 0$ case: Let $\underline{z}$ be a plane with IS$= 0$. This implies that (z_y, y) is not an erased node, for all $y \in \mathbb{Z}_t$. As a result, for any $(x, y) \notin \mathcal{E}$, both $A(x, y, \underline{z})$ and $A(z_y, y, \underline{z}(y, x))$ are known. This allows the symbols $B(x, y, \underline{z})$, $(x, y) \notin \mathcal{E}$ to be computed using the PRT, see (5.7). Therefore equation (5.5) reduces to

$$\sum_{(x,y)\in\mathcal{E}} \theta_{\ell,(x,y)} B(x, y, \underline{z}) = \kappa^*,$$

where κ^* is a known value. Thus we get r equations in r unknowns. The choice of $\{\theta_{\ell,(x,y)}\}$ guarantees the resultant $(r \times r)$ coefficient matrix is invertible. In this way, one can recover all symbols $\{B(x, y, \underline{z}) \mid (x, y) \in \mathcal{E}\}$ of the uncoupled data cube, corresponding to all planes $\underline{z}$ having IS $= 0$.

IS > 0 case: Next, let $\underline{z}$ be a plane with IS$= i$. We will first show that the symbols $\{B(x, y, \underline{z}) \mid (x, y) \notin \mathcal{E}\}$ can be computed using unerased code symbols as well as code symbols that were recovered from prior rounds of the sequential decoding process. For the case $(x, y) \notin \mathcal{E}$ and $(z_y, y) \notin \mathcal{E}$, both the symbols $A(x, y, \underline{z}), A(z_y, y, \underline{z}(y, x))$ are known. Therefore $B(x, y, \underline{z})$ can be computed from the PRT via

$$B(x, y, \underline{z}) \quad = \quad A(x, y, \underline{z}) + uA(z_y, y, \underline{z}(y, x)).$$

Now for the case when $(x, y) \notin \mathcal{E}$ and $(z_y, y) \in \mathcal{E}$, the plane $\underline{z}(y, x)$ has IS$= i-1$. Therefore we would have recovered the symbol $B(z_y, y, \underline{z}(y, x))$ in round $i - 1$. We know symbol $A(x, y, \underline{z})$ as it is unerased. Using the symbols $A(x, y, \underline{z})$ and $B(z_y, y, \underline{z}(y, x))$ and the 4-symbol MDS property noted above, the symbol $B(x, y, \underline{z})$ can be computed. In this way, we know $B(x, y, \underline{z})$ for any $(x, y) \notin \mathcal{E}$. As a result, equation (5.5) can be reduced to the form

$$\sum_{(x,y)\in\mathcal{E}} \theta_{\ell,(x,y)} B(x, y, \underline{z}) \quad = \quad \kappa^*,$$

where κ^* is a known value. Thus we end up once again with r equations in r unknowns. The choice of the $\{\theta_{\ell,(x,y)}\}$ guarantees the resultant $(r \times r)$ coefficient matrix is invertible. In this way, one can recover $\{B(x, y, \underline{z}) \mid (x, y) \in \mathcal{E}\}$ for all planes $\underline{z}$ having IS $= i$.

At the end of this decoding process we will have recovered all the uncoupled code-symbols $\{B(x, y, \underline{z}) \mid (x, y) \in \mathbb{Z}_r \times \mathbb{Z}_t, \underline{z} \in \mathbb{Z}_r^t\}$. By applying the PFT we can compute the erased node symbols $\{A(x, y, \underline{z}) \mid (x, y) \in \mathcal{E}, \underline{z} \in \mathbb{Z}_r^t\}$.

5.3.1 Extension to Other Parameters

Although the construction is explained for the case when (n, k, d) are of the form $(n = rt, k = r(t-1), d = n-1)$, the construction can be used to generate MSR codes for general parameter sets $(n, k, d = n-1)$ by shortening the code as described earlier in the section on the product-matrix MSR code.

For the case of $d < n-1$, it turns out that the coupled-layer construction technique can be applied, to result in an $[n = st, k = d+1-s]$ vector MDS code having sub-packetization $\alpha = s^t$. This coupled-layer MDS code is such that the repair of node (x_0, y_0) with optimal repair bandwidth, is possible only if the d helper nodes from each of which $\beta = s^{t-1}$ symbols are downloaded, includes all the $(s-1)$ nodes corresponding to the set $\{(x, y_0) \mid x \neq x_0\}$. Thus the coupled-layer construction does not yield an MSR code for the case $d < n-1$, since this represents a form of table-based repair.

5.4 Small-d MSR Codes

This subsection discusses an optimal-access MSR code construction for small values of d. Small values of d are of interest since they correspond to low repair degree. Additionally, many high-rate codes where the gap between n and k is small, may also fall into the small-d category. In [239], Vajha *et al.* present a construction for optimal-access MSR codes with

$$(n = st, k),\ d \in \{k+1, k+2, k+3\},\ (\alpha = s^t, s = (d-k+1)),$$

with $s \in \{2, 3, 4\}$ and $t \geq 2$, and field size q linear in block length, i.e., $q = O(n)$.

These codes have two additional attributes. Consider a setting in a linear RGC where node f has failed and we are interested in the data transferred by helper node h to the failed node f, and where the indices of the remaining $(d-1)$ helper nodes are specified by a set $D \subset [n]$ of size $(d-1)$. Since the RGC is linear, the data transferred can be represented in the form

$$S_{hf}^{(D)} \underline{c}_h$$

where $S_{hf}^{(D)}$ is a $(\beta \times \alpha)$ matrix and $\underline{c}_h$ represents the $(\alpha \times 1)$ vector corresponding to the data stored in node h. It turns out in the case of the Small-d MSR code construction, that the matrix $S_{hf}^{(D)}$ appearing above, is a function of the failed node f alone, and so we can simply write S_f in place of $S_{hf}^{(D)}$. This property is termed the constant-repair-matrix property. We note as an aside, that since the code is an optimal-access MSR code, the entries of each matrix S_f are either 0 or 1 with each row of S_f containing a single 1.

It turns out that not only do Small-d MSR code possess the constant-repair-matrix property, they also have the smallest possible sub-packetization level α possible, of any linear, optimal-access MSR code having the property that the repair matrix $S_{hf}^{(D)}$ is independent of the remaining helper nodes in D, so that we can write $S_{hf}^{(D)} = S_{hf}$. We term this latter property with respect to repair matrices, the helper-set-independence property. Clearly, the constant-repair-matrix property implies the helper-set-independence property.

By shortening a Small-d MSR code, one can construct additional optimal-access (n, k) MSR codes that also have constant repair matrices. These also have minimum sub-packetization level possible of a linear, optimal-access MSR code having the helper-set-independence property, provided $n \neq 1 \pmod{s}$ where $s = d - k + 1$. Details can be found in [239].

Open Problem 2. Construct an optimal-access MSR code having least-possible sub-packetization level for the case when $d = (n-1)$ and $n = 1 \mod r$.

Open Problem 3. Provide explicit constructions of optimal-access MSR codes having least-possible sub-packetization level, for all possible (n, k, d), with $d < (n-1)$.

Open Problem 4. Construct MSR codes with least-possible sub-packetization level for all (n, k, d)[2]. (There is no optimal-access requirement here).

Notes

1. Early constructions of high-rate MSR codes: The rate of the PM-MSR code can be at most a little larger than 0.5, as shown in Section 5.1.5. The construction of ER-MSR codes in the high-rate regime remained an open problem for quite some time. A high-rate MSR code was first provided in [167] for the parameter set $(n = k+2, k, d = k+1)$. The existence of ER-MSR codes for all (n, k, d) as B goes to infinity was established in [31]. The Zigzag code [229], [248] was the first non-asymptotic, high-rate construction for any $(n, k, d = n-1)$. Zigzag codes have sub-packetization level $\alpha = (n-k)^{k+1}$ and are non-explicit in general, as they make use of the Combinatorial Nullstellensatz [5] to establish the data-collection property. These code possess the optimal-access and optimal-update properties.

2. Non-explicit, high-rate MSR codes: A high-rate optimal-access MSR construction for the case $d = n-1$ with sub-packetization level $\alpha = r^{\lceil \frac{n}{r} \rceil}$ where $r = n-k$ appeared in [200]. The sub-packetization level of this linear code matches with the lower bound on sub-packetization of linear, optimal-access MSR codes derived in [11], provided $n \neq 1 \pmod r$. This construction was extended in [190] to the case $d < (n-1)$, with $\alpha = s^{\lceil \frac{n}{s} \rceil}$ where $s = d-k+1$. In [55], the authors generalize the PM-MSR construction to obtain an (n, k, d) MSR code with $\alpha = \binom{(t-1)(d-k+1)}{(t-1)}$, where $t \geq \frac{d}{d-k+1}$ is an integer. This code is based on multilinear

[2]The construction claim in [204] of an MSR code with $d < (n-1)$ and low sub-packetization level is incorrect, as pointed out by the authors of [204] in their revised posting on arXiv [206].

algebra, as is the Moulin code construction described in Section 7.3. The constructions in [55], [190], [200] are non-explicit as the Combinatorial Nullstellensatz [5] is employed to establish the data-collection property.

3. Systematic MSR codes: Vector MDS codes for which the optimal repair property holds only for the systematic nodes are referred to as systematic MSR codes. An early construction of a systematic MSR code with $\beta = 1$ for the case $d = (n-1) \geq (2k-1)$ can be found in [212]. In a subsequent paper [227] that builds upon [212], the authors provide a construction for MDS codes that can repair both systematic as well as parity nodes under the restriction $d \geq 2k-1, n \geq 2k$, under the assumption that all the un-erased systematic nodes participate in node repair. Thus for the case $d = (n-1) \geq 2k-1$, the construction in [227] yields an MSR code. Other early constructions of systematic MSR codes with $d = n-1$ can be found in [32], [229]. A general construction, valid for all (n, k, d) parameters sets, first appeared in [76]. A lower bound $\alpha \geq r^{\frac{k-1}{r}}$, where $r = n-k$, on the sub-packetization level of linear, systematic MSR codes with $d = n-1$ having the optimal-access property is derived in [233]. It is shown in [11] that this can extended to the slightly tighter bound $\alpha \geq r^{\lceil \frac{k-1}{r} \rceil}$. In [2], [33], [249], non-explicit, optimal-access, linear systematic MSR code constructions with $d = n-1$ having α matching the lower bound $\alpha \geq r^{\lceil \frac{k-1}{r} \rceil}$ for $k \neq 1 \pmod r$ are presented. Explicit constructions of optimal-access, linear systematic MSR codes with $d = (n-1)$ and $\alpha = r^{\lceil \frac{k-1}{r} \rceil}$ for $k \neq 1 \pmod r$ are provided for $(n-k) = 2, 3$, in [186]. Optimal-access, linear systematic MSR codes with $d = n-1$ having optimal sub-packetization level $r^{\lceil \frac{k-1}{r} \rceil}$ for $k \neq 1 \pmod r$ can be constructed over a field of size $q \geq n$ using the transformation presented in [137].

4. Optimal-access MSR codes for all (n, k, d) by Ye and Barg: In [255], apart from the Diagonal MSR code construction, the authors present the construction of a second class of MSR codes which we will refer to here as the Permuted-Diagonal MSR construction.

This construction yields an optimal-access MSR code for any (n, k, d) with sub-packetization level $\alpha = s^{n-1}$ where $s = d - k + 1$ and where the field size q satisfies $q \geq n + 1$. Permuted-Diagonal MSR codes are the only known explicit optimal-access MSR codes for any (n, k, d) having field size $O(n)$. As is the case with Diagonal MSR codes, these codes can also be extended to have the (h, d) optimal repair property for any $h \in [2, n - k]$, $d \in [k, n - h]$. In [148], a modification of the Permuted-Diagonal MSR code for the $d = n-1$ case is presented, which reduces the field size requirement to $q = 3$ for even r, and $q \geq r + 1$ for odd r where $r = n - k$.

6

Storage-Repair-Bandwidth Tradeoff

This section deals with storage-repair-bandwidth tradeoffs in the case of functional and exact repair. As seen in Section 3, in the case of FR the tradeoff is governed by the following equation, obtained by replacing the inequality in (3.13) with equality:

$$1 \quad = \quad \sum_{i=0}^{k-1} \min\{\bar{\alpha}, (d-i)\bar{\beta}\}. \tag{6.1}$$

In the present section, we will show that the FR tradeoff takes on the form of a piecewise linear curve. We will provide a more formal definition of the ER tradeoff and establish that apart from the MBR and MSR points, and possibly, a small region adjacent to the MSR point, the tradeoff under ER is clearly separated from the FR tradeoff.

6.1 Piecewise Linear Nature of FR Tradeoff

The aim here is to show that the locus of the set of pairs $(\bar{\alpha}, d\bar{\beta})$ with $\bar{\alpha} \geq 0$, $d\bar{\beta} \geq 0$, satisfying (6.1), is a piecewise-linear curve, with k corner points. We begin by partitioning the first quadrant in the $(x = \bar{\alpha}, y = d\bar{\beta})$ plane into the $(k+1)$ pairwise disjoint regions $\{\mathcal{R}_\ell \mid \ell = 0, 1, \cdots, k\}$ identified in Table 6.1.

Table 6.1: Partitioning the first quadrant in the $(x = \bar{\alpha}, y = d\bar{\beta})$ plane into the $(k+1)$ pairwise disjoint regions $\{\mathcal{R}_\ell \mid \ell = 0, 1, \cdots, k\}$. The storage-repair-bandwidth tradeoff under functional repair, is a piecewise-linear curve, represented by a straight line in each of the $(k+1)$ regions $\{\mathcal{R}_\ell\}$.

ℓ	$(x = \bar{\alpha}, y = d\bar{\beta}) \in \mathcal{R}_\ell$ iff
0	$d\bar{\beta} \leq \bar{\alpha}$,
$1 \leq \ell \leq k-1$,	$(d-\ell)\bar{\beta} \leq \bar{\alpha} < (d-\ell+1)\bar{\beta}$,
k	$\bar{\alpha} < (d-k+1)\bar{\beta}$.

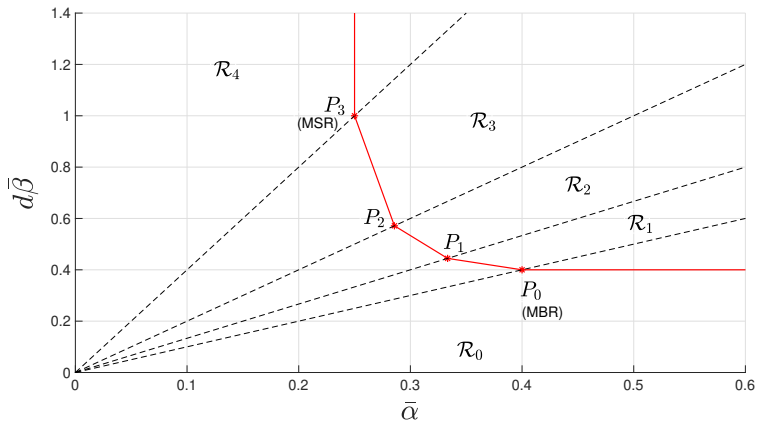

Figure 6.1: Illustrating the piecewise-linear nature (in red) of the normalized FR tradeoff for $(k = 4, d = 4)$. The $\{P_i\}$ denote the $k = 4$ corner points with P_0, P_3 representing the MBR and MSR points respectively.

We will show that in each region $\mathcal{R}_\ell, 0 \leq \ell \leq k$, the locus is a straight line (see Fig. 6.1).

1. When $(x = \bar{\alpha}, y = d\bar{\beta}) \in \mathcal{R}_0$, (6.1) takes on the form:

$$(dk - \binom{k}{2})\bar{\beta} \quad = \quad 1,$$

which represents a horizontal straight line.

2. When $(x = \bar{\alpha}, y = d\bar{\beta}) \in \mathcal{R}_\ell$, for $1 \leq \ell \leq (k-1)$, (6.1) takes on the form of the straight line

$$\ell\bar{\alpha} + \bar{\beta}\Big(\sum_{i=\ell}^{k-1}(d-i)\Big) \quad = \quad 1.$$

3. When $(x = \bar{\alpha}, y = d\bar{\beta}) \in \mathcal{R}_k$, (6.1) takes on the form

$$k\bar{\alpha} \quad = \quad 1,$$

which represents a vertical straight line.

This establishes the piecewise-linear nature of the locus of points satisfying (6.1). Clearly, there are k corner points $\{P_\ell, \ell = 0, 1, \cdots, (k-1)\}$, with corner point P_ℓ corresponding to the point of intersection of the straight lines associated with adjacent regions $(\mathcal{R}_\ell, \mathcal{R}_{\ell+1})$.

1. For the case $\ell = 0$, the coordinates of the corner point P_ℓ is hence obtained by solving

$$\sum_{i=0}^{k-1}(d-i)\bar{\beta} \quad = \quad \bar{\alpha} + \sum_{i=1}^{k-1}(d-i)\bar{\beta}, \; = \; 1,$$

i.e.,

$$\begin{aligned} d\bar{\beta} \quad &= \quad \frac{d}{dk - \binom{k}{2}}, \\ \bar{\alpha} \quad &= \quad d\bar{\beta}. \end{aligned}$$

This corner point $P_0 = \left(\frac{d}{dk-\binom{k}{2}}, \frac{d}{dk-\binom{k}{2}}\right)$ corresponds to the normalized values $(\bar{\alpha}, d\bar{\beta})$ of an MBR code.

2. For the case $1 \leq \ell \leq (k-2)$, the coordinates of the corner point $P_\ell = \left(\bar{\alpha}, d\bar{\beta}\right)$ is obtained by solving for $\bar{\alpha}$ and $d\bar{\beta}$ from the equations below:

$$\ell\bar{\alpha} + \sum_{i=\ell}^{k-1}(d-i)\bar{\beta} \quad = \quad (\ell+1)\bar{\alpha} + \sum_{i=\ell+1}^{k-1}(d-i)\bar{\beta} \; = \; 1,$$

i.e.,

$$\begin{aligned} \ell\bar{\alpha} + \sum_{i=\ell}^{k-1}(d-i)\bar{\beta} &= 1, \\ \bar{\alpha} &= (d-\ell)\bar{\beta}. \end{aligned}$$

3. For the case $\ell = (k-1)$, the coordinates of the corner point P_ℓ is obtained by solving

$$(k-1)\bar{\alpha} + (d-k+1)\bar{\beta} = k\bar{\alpha} = 1,$$

i.e.,

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

This corner point $P_{k-1} = \left(\frac{1}{k}, \frac{d}{k(d-k+1)}\right)$ corresponds to the normalized values $(\bar{\alpha}, d\bar{\beta})$ of an MSR code.

6.2 ER Tradeoff

Our aim here is to characterize the normalized pairs $(\bar{\alpha}, d\bar{\beta})$ for which it is possible to construct an ER RGC having parameters (n,k,d) over some finite field $\mathbb{F}_q$. We begin by noting that for any parameter set (n,k,d), there exist constructions of ER MSR and ER MBR codes. In the case of MBR codes this is apparent from the product-matrix construction of an MBR code. In the case of an MSR code, this is clear from the Diagonal MSR construction appearing in Section 5.2. Thus at the points on the FR tradeoff corresponding to the MSR and MBR points, there exist ER RGCs having the same normalized parameters as FR RGCs. Since the FR tradeoff represents an outer bound to the ER tradeoff[1], this tells us that the ER and FR tradeoffs share the MSR and MBR points in common.

With this in mind, we define the ER tradeoff for fixed (n,k,d) as the locus of all normalized pairs $(\bar{\alpha}, d\bar{\beta})$ that meet the following requirements:

[1] Meaning that for the same parameter set $\{(n,k,d),(\alpha,\beta)\}$, the file size B under ER is no larger than the file size under FR.

- $\frac{1}{k} \leq \bar{\alpha} \leq \frac{d}{dk-\binom{k}{2}}$,
- There exists an ER RGC having normalized parameters $(\bar{\alpha}, d\bar{\beta})$,
- There does not exist an ER RGC having normalized parameters (x, y) of the form $(x < \bar{\alpha}, y = d\bar{\beta})$ or $(x = \bar{\alpha}, y < d\bar{\beta})$.

When we speak of an interior point in the ER tradeoff, we mean a point lying on this locus having normalized value $\bar{\alpha}$ satisfying: $\frac{1}{k} < \bar{\alpha} < \frac{d}{dk-\binom{k}{2}}$.

While much effort has been expended and significant progress on the problem has been made, complete characterization of the ER tradeoff still remains open. Clearly, the FR tradeoff provides a trivial outer bound to the ER tradeoff.

6.3 Non-existence of ER Codes Achieving FR Tradeoff

As in the case of the ER tradeoff, when we speak of an interior point in the FR tradeoff, we mean a point lying on the piecewise-linear tradeoff in the FR case (see Section 6.1) having normalized value $\bar{\alpha}$ satisfying: $\frac{1}{k} < \bar{\alpha} < \frac{d}{dk-\binom{k}{2}}$. It turns out that there do not exist ER RGCs whose parameters correspond to an interior point in the normalized FR tradeoff, apart possibly from a small region adjoining the MSR point. The precise statement is given below.

Theorem 3. (Theorem 7 in [211]) ER RGCs having normalized parameter set $((n, k, d), (\bar{\alpha}, \bar{\beta}))$ such that they correspond to an interior point on the FR tradeoff do not exist, except possibly for a small region in the $(\bar{\alpha}, d\bar{\beta})$ plane corresponding to the range given below for the parameter $\bar{\alpha}$:

$$(d-k+1)\bar{\beta} < \bar{\alpha} \leq \left[(d-k+2) - \frac{d-k+1}{d-k+2}\right]\bar{\beta}.$$

In this subsection, we provide a sketch of the proof of non-existence only in the case of interior corner points having normalized coordinates

$(\bar{\alpha}, \bar{\beta})$ satisfying $\bar{\alpha} = (d-p)\bar{\beta}$, for $1 \leq p \leq (k-2)$. We will do this by showing that there do not exist ER RGCs having parameter set

$$\{(n,k,d),(\alpha,\beta),B,\mathbb{F}_q\},$$

where the file-size B satisfies the bound

$$B \leq \sum_{i=0}^{k-1} \min\{\alpha,(d-i)\beta\}, \tag{6.2}$$

with equality for α of the form $\alpha = (d-p)\beta$ for some integer p, with $1 \leq p \leq (k-2)$. We refer the reader to [211] for the complete proof.

Proof: We follow the derivation in [211]. Let $\mathcal{C}$ be an ER RGC having parameter set $\{(n,k,d),\ (\alpha,\beta),\ B,\ \mathbb{F}_q\}$ with $\alpha = (d-p)\beta, 1 \leq p \leq (k-2)$, that satisfies the cut-set bound in (6.2) with equality. We will show that this leads to a contradiction. We restrict attention in the proof, to a subset D of $(d+1)$ nodes that by themselves, form a regenerating code $\mathcal{C}_D$ having parameter set $\{(d+1,k,d),(\alpha,\beta),B\}$ that clearly, also achieves the cut-set bound with equality. We continue to adopt the notation W_A, S_A^B etc., that was introduced in Section 3.2. Let $A \subseteq D$ denote a subset of D of size $|A| = k$. We have that

$$\begin{aligned}
B &= H(W_A) \leq H(W_D) \\
&= I\left(W_D; \left\{S_{D\setminus\ell}^{\ell}\right\}_{\ell\in D}\right) \\
&\leq H\left(\left\{S_{D\setminus\ell}^{\ell}\right\}_{\ell\in D}\right) \\
&= H\left(\left\{S_{m}^{D\setminus\{m\}}\right\}_{m\in D}\right) \\
&\leq \sum_{m\in D} H(S_m^{D\setminus\{m\}}) \\
&= \sum_{m\in D} \beta \\
&= (d+1)\beta.
\end{aligned}$$

In arriving at this result, we have used two properties established in [211]. For any pair of distinct nodes $\{\ell, m\} \subseteq D$, we have

$$H(S_m^{\ell}) = \beta,$$

and this appears as Property 3 in [211]. For any three distinct nodes $\{\ell_1, \ell_2, m\} \subseteq D$, we have :

$$H(S_m^{\ell_1} \mid S_m^{\ell_2}) \quad = \quad 0,$$

and this appears as part of Property 5 in [211] (after setting the parameter θ appearing in [211] to 0 since our focus here is only on corner points that lie within the interior).

On the other hand, if an ER RGC attains the cut-set upper bound in (6.2) with $\alpha = (d-p)\beta$, for p an integer lying in the range $1 \leq p \leq (k-2)$, we must have

$$\begin{aligned} B &= \sum_{i=0}^{k-1} \min\{\alpha,\ (d-i)\beta\} \\ &= \sum_{i=0}^{k-1} \min\{(d-p)\beta,\ (d-i)\beta\} \\ &= 2(d-p)\beta + \sum_{i=2}^{k-1} \min\{(d-p)\beta,\ (d-i)\beta\} \\ &\geq 2(d-p)\beta + (k-2)\beta \\ &\geq (d+2)\beta, \end{aligned}$$

leading to a contradiction. □

6.4 Outer Bounds on the Tradeoff Under ER

ER Codes Cannot Have Tradeoff Approaching the FR Tradeoff A first result in this direction, was established by Tian [237] for the specific parameter set $(n = 4, k = 3, d = 3)$. Tian was able to show that for these parameter values the file size B under exact repair is upper bounded by $B \leq 4\alpha + 6\beta$. This result was arrived at via a computer-aided proof that makes use of the linear programming approach to inequalities involving entropic expressions introduced by Yeung [110], [258]. When this result was compared to the FR tradeoff for the same code parameters, a non-vanishing gap between ER and FR tradeoffs can be seen, as shown in Fig. 6.2.

The general version of this result, namely that the ER tradeoff is strictly away from the FR tradeoff for every set (n, k, d) of parameters,

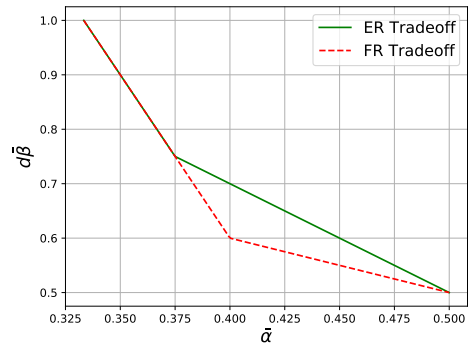

Figure 6.2: Normalized ER tradeoff for $(n = 4, k = 3, d = 3)$

was first established by Sasidharan *et al.* [198]. This is established for all interior points other than those lying in the small region adjacent to the MSR point identified in Theorem 3. This result was obtained by exploiting the proof of the non-existence result appearing in [211], and described in brief in Section 6.3 above, to show that there is a non-vanishing gap between a lower bound on file size under FR and an upper bound on file size under ER.

The proof of a non-vanishing gap in file size between the cases of ER and FR, corresponds to providing an upper bound on file size under ER that is tighter than in the case of FR. This upper bound on file size under ER, was subsequently improved in [57].

The (n, d, d) Case For the case $k = d$, the following outer bound on the maximum file size of a linear RGC,

$$B \leq \frac{d+1}{\ell+2}\left(\ell\alpha + \frac{d}{\ell+1}\beta\right), \tag{6.3}$$

where $\ell = \lfloor d\beta/\alpha \rfloor \in \{0, 1, \ldots, d\}$, was derived in [173] by carefully analyzing the p-c matrices of ER RGCs. This bound establishes a piecewise linear outer bound on the ER tradeoff as it applies to linear RGCs, that is tighter than the bound provided by the FR tradeoff. The same bound was independently derived in [60] by solving an optimization problem involving the file-size of a linear ER RGC.

In [238], the outer bound in (6.3) is shown to hold for the specific parameter set $(n = 5, k = 4, d = 4)$ even in the case of a general ER RGC, by adopting a computational approach to handling information-theoretic inequalities. By a general ER RGC, we mean an ER RGC that is not necessarily linear. A subsequent outer bound, appearing in [56] and that also applies to a general ER RGC, coincides with that in [60], [173] when specialized to the linear setting and to the case of parameter sets of the form (n, d, d).

The class of Determinant codes [61], [63] discussed in Section 7.1, turn out to achieve the outer bound in (6.3). This both establishes the tightness of the bound in (6.3) as it applies to linear ER RGCs, as well as the optimality of Determinant codes when one restricts attention to linear ER RGCs.

The Best-Known Upper Bound on File Size Under ER In [162], Mohajer and Tandon derived an upper bound

$$B \leq \min_{0\leq p\leq k} \left\{ \frac{(3k-2p)\alpha + \frac{p(2(d-k)+p+1)\beta}{2} + (d-k+1)\min\{\alpha, p\beta\}}{3} \right\} \tag{6.4}$$

on file size for the general case, that was significantly tighter than prior bounds in the literature. The bound was derived by bounding the conditional joint entropy of certain repair data random variables in three different ways and adding them together to cancel out a few terms, that were otherwise difficult to estimate. The above Mohajer-Tandon bound was improved in [203] leading to a strictly better bound for the case $d > k$. The improved Mohajer-Tandon bound derived in [203] is given by

$$B \leq \min_{0\leq p\leq k} \left\{ \frac{\alpha(2(k-p)(1+a)+k(1+2a)) + b\min\{\alpha, p\beta\} + \frac{(1+2a)p(2(d-k)+p+1)\beta}{2}}{3+4a} \right\} \tag{6.5}$$

where $d - k + 1 = a(p-1) + b$ and $0 \leq b < (p-1)$. The bound in [203] adopts the same approach as in [162]. The improvement arises from identifying the symmetry in certain entropic terms observed by representing repair data random variables in matrix form, and leveraging this symmetry to avoid the need for employing certain union bounds. The improved Mohajer-Tandon bound remains the best-known outer bound on ER tradeoff for general (n, k, d).

Open Problem 5. Characterize the storage-repair-bandwidth tradeoff of an (n, k, d) regenerating code under exact repair.

Remark 6. The parameters of the Cascade and Moulin codes (described in Section 7) provide the best known inner bound to the ER tradeoff.

7

Interior-Point ER Codes

Let $(\bar{\alpha}_{\text{MSR}}, d\bar{\beta}_{\text{MSR}})$ and $(\bar{\alpha}_{\text{MBR}}, d\bar{\beta}_{\text{MBR}})$ denote the $(\bar{\alpha}, d\bar{\beta})$ values at the MSR and MBR points respectively, given by:

$$\begin{aligned}(\bar{\alpha}_{\text{MSR}}, d\bar{\beta}_{\text{MSR}}) &= \left(\frac{1}{k}, \frac{d}{k(d-k+1)}\right),\\ (\bar{\alpha}_{\text{MBR}}, d\bar{\beta}_{\text{MBR}}) &= \left(\frac{d}{dk-\binom{k}{2}}, \frac{d}{dk-\binom{k}{2}}\right).\end{aligned}$$

An ER RGC with normalized parameters $\{(n,k,d),\ (\bar{\alpha},\bar{\beta})\}$ will be said to be an interior-point ER (IP-ER) RGC if $\bar{\alpha}_{\text{MSR}} < \bar{\alpha} < \bar{\alpha}_{\text{MBR}}$. Given an $\bar{\alpha}_0$ lying strictly between $\bar{\alpha}_{\text{MSR}}$ and $\bar{\alpha}_{\text{MBR}}$, and the minimum possible value $\bar{\beta} = \bar{\beta}_0$ attainable by an ER RGC, the code operating at the $(\bar{\alpha}_0, d\bar{\beta}_0)$ point will be said to be an optimal IP-ER RGC. The locus of all such points $(\bar{\alpha}_0, d\bar{\beta}_0)$ is the ER storage-repair-bandwidth tradeoff. Clearly, $(\bar{\alpha}_{\text{MBR}}, d\bar{\beta}_{\text{MBR}})$ and $(\bar{\alpha}_{\text{MSR}}, d\bar{\beta}_{\text{MSR}})$ are at the two extreme ends of the ER tradeoff.

A listing of some of the codes in the literature that attain, or are conjectured to attain, some portion in the interior of the storage-repair-bandwidth tradeoff under ER is given in Table 7.1.

In this section, we will first present an (n,d,d) construction, i.e., a construction for the case $k = d$. The associated RGC will be referred

Table 7.1: A listing of some of the codes in the literature that attain, or are conjectured to attain, some portion in the interior of the storage-repair-bandwidth tradeoff under exact repair.

Code	Parameter Set	Extent to which Construction Attains the ER Tradeoff
$(4,3,3)$ Tian [237]	$(4,3,3)$	Achieves entire ER tradeoff
Canonical Layered [236]	(n,d,d)	Achieves single point for $(n,n-1,n-1)$ case
Improved Layered [210]	(n,k,d)	Entire ER tradeoff for $(n,k=3,n-1)$; Achieves single point for $(n,k=4,n-1)$ case
Determinant [61], [63]	(n,d,d)	Achieves entire ER tradeoff applicable to linear RGCs
Cascade [64]	(n,k,d)	Conjectured to achieve entire ER tradeoff
Moulin [54]		

to as the Signed Determinant code. This construction has an auxiliary parameter $\sigma \in \mathbb{Z}^d$. Setting $\sigma = 0$ yields the Determinant code presented in [61], [63] that attains the storage-repair-bandwidth tradeoff as it applies to linear ER RGCs for the (n,d,d) case. A discussion on the tradeoff in the (n,d,d) case, applicable to linear ER RGCs, can be found in Section 6.4. Following this, we will briefly discuss two code constructions, namely the Cascade code construction and the Moulin code construction, due respectively, to Elyasi and Mohajer [64] and Duursma *et al.* [54] sharing identical $(\bar{\alpha}, \bar{\beta})$ parameters for given (n,k,d). These codes yield the best-known inner bound to the storage-bandwidth tradeoff under ER. It is conjectured in [64] that the tradeoff achieved by the Cascade code construction (and hence also by the Moulin code construction) represents the storage-repair-bandwidth tradeoff under exact repair.

7.1 Determinant Code

Signed Determinant Code Let $\sigma \in \mathbb{Z}^d$ be a fixed d-length vector of integers. Let $\sigma(j)$ denote the jth entry of σ. We now describe the

Signed Determinant code due to Elyasi and Mohajer [61], [63], [64] for parameters (n, d, d), i.e., for the case $k = d$. The code is called the Signed Determinant code because of the sign factor introduced by the components of σ. It turns out that if one is interested solely in the case $k = d$, i.e., the (n, d, d) case, one can set the vector $\sigma = 0$, i.e., $\sigma(j) = 0$, all $j \in [d]$. As noted above, setting $\sigma = 0$ yields the Determinant code construction appearing in [61], [63]. While both papers [61], [63] describe the same Determinant code, the repair process described in [63] has the advantage that the helper data supplied by a helper node does not depend upon the identity of the remaining $(d-1)$ helper nodes[1]. We have retained σ in the expressions below, as the vector σ is needed when the Signed Determinant code is used as a building block to construct Cascade code [64]. Our description of the Signed Determinant code below, follows the description of the code given in [64]. The repair process of the Signed Determinant code described below is helper-set independent.

The Signed Determinant construction is parameterized by an integer variable m, with $1 < m < d$. The associated (α, β, B) parameters are then given by:

$$\begin{aligned} \alpha_m &= \binom{d}{m}, \\ \beta_m &= \binom{d-1}{m-1}, \qquad (7.1) \\ B_m &= m\binom{d}{m} + m\binom{d}{m+1} = m\binom{d+1}{m+1}. \qquad (7.2) \end{aligned}$$

Let

$$\begin{aligned} \mathcal{V} &= \{v_{Aj} \mid A \subset [d], |A| = m, j \in A\}, \\ \mathcal{W} &= \{w_{Sj} \mid S \subseteq [d], |S| = m+1, j \in S\} \end{aligned}$$

be two sets of symbols that take on values in a finite-field $\mathbb{F}$. Let

$$\mathcal{W}' = \{w_{Sj} \mid w_{Sj} \in \mathcal{W}, j \neq \max S\}$$

[1] This is the helper-set-independent property described in Section 5.4. It turns out that in the case of the Determinant and Signed Determinant codes that the repair process is linear and involves constant repair matrices. These terms are defined in Section 5.4.

be a subset of $\mathcal{W}$. Then $\mathcal{V} \cup \mathcal{W}'$ is of size B_m and this is the set of message symbols associated to the data file being stored. The symbol $w_{S,\max S}$ for every S is determined by the p-c equation

$$\sum_{j \in S} (-1)^{\tau_S(j)} w_{Sj} \quad = \quad 0,$$

where $\tau_S(j)$ is the position of j, given that the elements of S are listed in ascending order. In other words, $\tau_S(j) = |\{i \in S \mid i \leq j\}|$ for any $j \in S$. The symbols in $\mathcal{V} \cup \mathcal{W}$ are used to populate two matrices V, W having respective size $(\binom{d}{m} \times d)$ and $(\binom{d}{m+1} \times d)$. The two matrices will respectively be referred to as the V-array and the W-array. The rows of the V-array are indexed by m-subsets of $[d]$ and the columns by $1, 2, \ldots d$. The symbol $v_{Aj} \in \mathcal{V}$ occupies a cell in the V-array, determined by row A and column $j \in A$. In similar fashion, the rows of the W-array are indexed by the $(m+1)$-subsets of $[d]$ and the columns by $1, 2, \ldots d$. The symbol $w_{Sj} \in \mathcal{W}$ occupies a cell in the W-array, determined by the row S and the column $j \in S$. Note that each row in the V-array contains m symbols, the remaining $(d-m)$ cells in each row are empty. Similarly each row in the W-array contains $(m+1)$ symbols, the remaining $(d-m-1)$ cells in each row are empty.

Next, we will construct a matrix which we will refer to as the data matrix D (at times, we will also refer to D as the D-array) having $\binom{d}{m}$ rows and d columns. Again, the rows are indexed by m-subsets of $[d]$ and the columns by $1, 2, \ldots, d$. The d_{Aj}th entry of D is given by:

$$d_{Aj} \quad = \quad \begin{cases} (-1)^{\sigma(j)} \cdot v_{Aj}, & \text{if } j \in A \\ (-1)^{\sigma(j)} \cdot w_{A \cup \{j\}, j} & \text{if } j \notin A \end{cases}.$$

The population of the data matrix D is illustrated for $d = 4$ in Fig. 7.1. The codeword array C of size $(\alpha_m \times n)$ formed by the contents of the n nodes each containing α_m symbols is generated :

$$C \quad = \quad D\Phi$$

where D is the $(\alpha_m \times d)$ data matrix and Φ is a $(d \times n)$ generator matrix. The generator matrix Φ is required to satisfy that set of every d columns must be of full-rank.

	1	2	3	4
1	v_{11}	v_{12}		
2	v_{21}		v_{22}	
3	v_{31}			v_{32}
4		v_{41}	v_{42}	
5		v_{51}		v_{52}
6			v_{61}	v_{62}

	1	2	3	4
1	w_{11}	w_{12}	w_{13}	
2	w_{21}	w_{22}		w_{23}
3	w_{31}		w_{32}	w_{33}
4		w_{41}	w_{42}	w_{43}

	1	2	3	4
1	v_{11}	v_{12}	w_{13}	w_{23}
2	v_{21}	w_{12}	v_{22}	w_{33}
3	v_{31}	w_{22}	w_{32}	v_{32}
4	w_{11}	v_{41}	v_{42}	w_{43}
5	w_{21}	v_{51}	w_{42}	v_{52}
6	w_{31}	w_{41}	v_{61}	v_{62}

Figure 7.1: The V, W and D matrices used in the construction of $(n, 4, 4)$ Signed Determinant code with $m = 2$ and $\sigma = (0, 0, 0, 0)$. For simplicity, in the figure, the m-subsets of $\{1, 2, 3, 4\}$ have been ordered in lexicographically ascending order and indexed from 1 to 6. Similarly, the $(m + 1)$-subsets of $\{1, 2, 3, 4\}$ have also been ordered lexicographically and indexed from 1 to 4.

Data Collection: Let J denote the set of $k = d$ nodes from which data is to be recovered. By construction, the matrix Φ restricted to the columns indexed by J is invertible. Thus the data matrix D can be recovered.

Node Repair: Without loss of generality, suppose that the first node has failed, and that the helper nodes are $\{2, 3, \ldots, d + 1\}$. Let R denote a matrix of size $(\binom{d}{m-1} \times \alpha_m)$, which we will term as the repair matrix, since we will use R to generate the repair data that is used to repair failed node 1. It turns out that R has rank no larger than $\beta_m = \binom{d-1}{m-1}$ so that in practice, R can be replaced by a $(\beta_m \times \alpha_m)$ matrix. If $\underline{\phi}_j, 1 \leq j \leq n$ denotes the jth column of Φ, then the repair data passed on by the helper nodes to the failed node is given by

$$\mathcal{Z}_0 \quad = \quad R \times D \times [\underline{\phi}_2 \ \underline{\phi}_3 \ \cdots \ \underline{\phi}_{d+1}].$$

By construction, the matrix $[\underline{\phi}_2\ \underline{\phi}_3\ \cdots\ \underline{\phi}_{d+1}]$ is invertible and thus the replacement of the failed node has access to the product matrix

$$\mathcal{Z} \quad = \quad RD.$$

Our goal is to identify a matrix R such that we can recover the contents $\underline{c}_1 = D\underline{\phi}_1$ of the failed node given the product RD. Each row of $\underline{c}_1$ is indexed by an m-subset A of $[d]$, and we write c_{A1} to denote the entry of $\underline{c}_1$ in row A.

The matrix R is of size $\binom{d}{m-1} \times \binom{d}{m}$. The entries of R are completely determined from the symbols $\{\phi_{i1} \mid 1 \leq i \leq d\}$ making up $\underline{\phi}_1$. Thus R is solely a function of the index of the failed node, index 1 in the present case. The rows and columns of R are respectively indexed by $(m-1)$-subsets and m-subsets of $[d]$. If r_{PA} denotes the entry in the P-th row and A-th column of R, then

$$r_{PA} \quad = \quad \begin{cases} (-1)^{\sigma(y)+\tau_A(y)} \cdot \phi_{y1}, & \text{if } y \text{ exists such that } P \cup \{y\} = A, \\ 0, & \text{otherwise.} \end{cases}$$

For an arbitrary $A \subset [d], |A| = m$, we now show that c_{A1} can be recovered using the equation:

$$c_{A1} \quad = \quad \sum_{i \in A} (-1)^{\sigma(i)+\tau_A(i)} R_{A\setminus\{i\}} D_i \tag{7.3}$$

where $R_{A\setminus\{i\}}$ is the row-vector of R associated to the row $A \setminus \{i\}$ and D_i is the ith column-vector of D. We begin by introducing some notation:

$$A_{\sim i} := A \setminus \{i\}, \quad A_{\sim i,y} := \{y\} \cup A \setminus \{i\} \quad \text{and } A_y := A \cup \{y\}.$$

We then have

$$\begin{aligned} &\sum_{i \in A} (-1)^{\sigma(i)+\tau_A(i)} R_{A\setminus\{i\}} D_i \\ &\quad = \quad \sum_{i \in A} (-1)^{\sigma(i)+\tau_A(i)} \sum_{L \subset [d], |L|=m} r_{A_{\sim i},L} d_{Li} \\ &\quad = \quad \sum_{i \in A} (-1)^{\sigma(i)+\tau_A(i)} r_{A_{\sim i},A} d_{Ai} \\ &\qquad\qquad + \sum_{i \in A} (-1)^{\sigma(i)+\tau_A(i)} \sum_{y \in [d]\setminus A} r_{A_{\sim i},A_{\sim i,y}} d_{A_{\sim i,y},i} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i\in A}\phi_{i1}d_{Ai} \\
&\quad + \sum_{i\in A}(-1)^{\sigma(i)+\tau_A(i)} \sum_{y\in[d]\setminus A}(-1)^{\sigma(y)+\tau_{A_{\sim i,y}}(y)}\phi_{y1}\ d_{A_{\sim i,y},i} \\
&= \sum_{i\in A}\phi_{i1}d_{Ai} \\
&\quad + \sum_{y\in[d]\setminus A}\phi_{y1}\sum_{i\in A}(-1)^{\sigma(i)+\sigma(y)+\tau_A(i)+\tau_{A_{\sim i,y}}(y)}(-1)^{\sigma(i)}w_{A_y,i}. \qquad (7.4)
\end{aligned}
$$

To proceed further, we observe that for $i \neq y$:

$$
\begin{aligned}
&\tau_A(i) + \tau_{A_{\sim i,y}}(y) \\
&\quad = |\{u \in A \mid u \le i\}| + |\{u \in A_{\sim i,y} \mid u \le y\}| \\
&\quad = |\{u \in A_y \mid u \le i\}| - \mathbf{1}(y < i) + |\{u \in A_y \mid u \le y\}| - \mathbf{1}(i < y) \\
&\quad = |\{u \in A_y \mid u \le i\}| + |\{u \in A_y \mid u \le y\}| - 1 \\
&\quad = \tau_{A_y}(i) + \tau_{A_y}(y) - 1.
\end{aligned}
$$

Substituting back in (7.4), we have

$$
\begin{aligned}
&\sum_{i\in A}(-1)^{\sigma(i)+\tau_A(i)}R_{A\setminus\{i\}}D_i \\
&\quad = \sum_{i\in A}\phi_{i1}d_{Ai} \\
&\qquad + \sum_{y\in[d]\setminus A}(-1)^{\sigma(y)+\tau_{A_y}(y)}\phi_{y1}\sum_{i\in A}(-1)^{\sigma(i)+\tau_{A_y}(i)-1}(-1)^{\sigma(i)}w_{A_y,i} \\
&\quad = \sum_{i\in A}\phi_{i1}d_{Ai} + \sum_{y\in[d]\setminus A}(-1)^{\sigma(y)+\tau_{A_y}(y)}\phi_{y1}\left[-\sum_{i\in A}(-1)^{\tau_{A_y}(i)}w_{A_y,i}\right] \\
&\quad = \sum_{i\in A}\phi_{i1}d_{Ai} + \sum_{y\in[d]\setminus A}(-1)^{\sigma(y)+\tau_{A_y}(y)}\phi_{y1}\cdot(-1)^{\tau_{A_y}(y)}w_{A_y,y} \\
&\quad = \sum_{i\in A}\phi_{i1}d_{Ai} + \sum_{y\in[d]\setminus A}\phi_{y1}d_{Ay} \ = \ \sum_{i\in[d]}d_{Ai}\phi_{i1} \ = \ c_{A1}.
\end{aligned}
$$

In this way, we are able to recover all the contents $\{c_{A1}\ A \subset [n], |A| = m\}$ of node 1. As noted earlier, the matrix R can be shown to have rank at most $\binom{d-1}{m-1}$ [64] and thus the repair bandwidth is no larger than $\beta_m = \binom{d-1}{m-1}$.

Determinant Code A careful reading of the derivation of the data-collection and node-repair properties of the Signed Determinant code shows that the arguments go through regardless of the specific value assigned to vector $\sigma \in \mathbb{Z}^d$. In particular, we can set the vector $\sigma = 0$, i.e., $\sigma(j) = 0$, all $j \in [d]$. Setting $\sigma = 0$ yields the Determinant code construction appearing in [61], [63] under the helper-set-independent repair process appearing in [63]. The ER tradeoff achieved by Determinant codes coincides with the piece-wise linear outer bound given in (6.3) if we set the parameter ℓ appearing in the bound, to equal the parameter m of the Signed Determinant code. Thus as noted previously in Section 6.4, the performance of Determinant codes characterizes the storage-repair-bandwidth tradeoff as it applies to a linear ER RGC having parameters of the form (n, d, d).

7.2 Cascade Code

In [64], the authors introduce a code termed as the Cascade code, that is constructed using multiple Signed Determinant codes as building blocks. The resulting Cascade code, as well as the Moulin code described below in Section 7.3, both have the best-known storage-repair-bandwidth tradeoff offered by an ER RGC, operating at an interior point. The parameters of a Cascade code for given (n, k, d) are given by:

$$\begin{aligned} \alpha(\mu) &= \sum_{m=0}^{\mu} (d-k)^{\mu-m} \binom{k}{m}, \\ \beta(\mu) &= \sum_{m=0}^{\mu} (d-k)^{\mu-m} \binom{k-1}{m-1}, \\ B(\mu) &= \sum_{m=0}^{\mu} k(d-k)^{\mu-m} \binom{k}{m} - \binom{k}{\mu+1}, \end{aligned} \tag{7.5}$$

where $\mu, 1 \leq \mu \leq k$, is an auxiliary integer parameter. In evaluating these expressions, we set $\binom{\ell_1}{\ell_2} = 0$ if $\ell_2 < 0$ or $\ell_2 > \ell_1$. It can be verified that setting $\mu = 1$ yields an MBR code, setting $\mu = k$, an MSR code. In addition, it turns out that setting $\mu = k - 1$ also yields a point on the FR tradeoff, close to the MSR point. The construction of the Cascade code is linear and the field size is $\Theta(n)$. As is the case with the

Determinant code, the repair process is helper-set-independent. Setting $d = k$ leads to the parameters of a Determinant code.

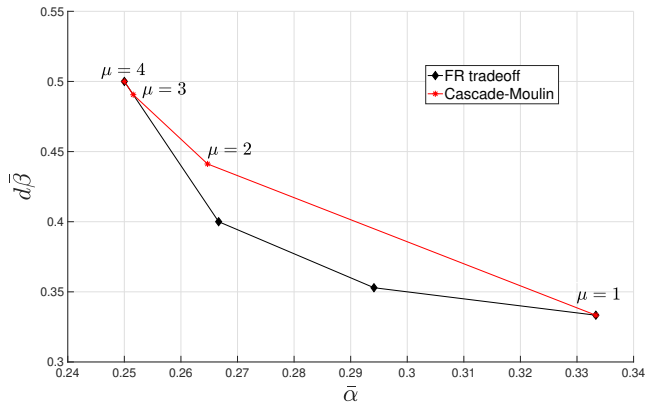

Figure 7.2: Comparing the common storage-repair-bandwidth performance of the Cascade and Moulin codes against the corresponding FR tradeoff for the parameter set $(n = 8, k = 4, d = 6)$. (To identify the auxiliary parameter appearing in the Moulin code construction, we set $s = \mu + 1$.)

The performance achieved by the Cascade code is compared in Fig. 7.2, with the FR tradeoff for an example parameter set $(n = 8, k = 4, d = 6)$. It has been conjectured in [64] that the performance of the Cascade code (and hence of the Moulin code described below as well), characterizes the piecewise-linear, tradeoff of an ER RGC.

7.3 Moulin Code

The Moulin[2] code is a linear ER RGC due to Duursma *et al.* [54] that is described in terms of a multilinear algebra framework. In the Moulin-code framework, each codeword is associated to a linear functional acting on a parent vector space, i.e., a linear transformation from the parent vector space to its underlying field of scalars. The symbols stored in a node are obtained by evaluating the linear functional on elements of

[2]Name given by the authors of [54] who indicate that their choice of name was inspired by the words cascade (waterfall or moulin) and multilinear algebra.

a subspace of the parent vector space. The repair data transferred from a helper node are derived by evaluating the linear functional on elements of a subspace properly contained within the subspace associated with the helper node. We limit ourselves in this subsection to providing a brief description of the Moulin-code construction, to convey a sense of the multilinear algebra framework on which the construction is based, and refer the reader to [54] for the more complete mathematical description.

Since the Moulin code and Cascade code have the same values of (α, β, B) for a given (n, k, d), they offer the same performance. (We identify auxiliary parameter s in the case of the Moulin code with the parameter $\mu + 1$ in the case of the Cascade code). Thus, both have the same sub-packetization level. The Moulin code also has a linear field-size requirement and even here, repair data passed on by a specific helper node to a failed node, does not depend upon the identity of the remaining $(d-1)$ helper nodes.

Multilinear Algebra Background Let X, Y be two finite-dimensional vector spaces over a field $\mathbb{F}$ having ordered bases $\{x_1, x_2, \ldots, x_{\ell_1}\}$ and $\{y_1, y_2, \ldots, y_{\ell_2}\}$ respectively. The tensor product $X \otimes Y$ of X and Y is an $\ell_1\ell_2$-dimensional vector space consisting of all tensors

$$x \otimes y \;=\; \sum_i a_i x_i \otimes \sum_j b_j y_j \;=\; \sum_{i,j} a_i b_j (x_i \otimes y_j),$$

where $(x_i \otimes y_j)$ may be regarded as an unbreakable expression. The tensor product extends naturally to more than two vector spaces.

Let V and W be vector spaces over $\mathbb{F}$ of dimension $d-k$ and k respectively, so that $U = V \oplus W$ is isomorphic to $\mathbb{F}^d$. Let $\mathcal{B}_W = \{w_1, w_2, \ldots, w_k\}$ be a basis of W. We define T^pW as the p-fold tensor product of W with itself:

$$T^pW \;=\; \left\{ \sum_{\substack{\underline{j}=(j_1,j_2,\ldots,j_p) \\ j_i \in [k], \forall i}} a_{\underline{j}} \cdot (w_{j_1} \otimes w_{j_2} \otimes \cdots \otimes w_{j_p}) \mid a_{\underline{j}} \in \mathbb{F} \right\},$$

where the sum is a formal sum and where $w_{j_1} \otimes w_{j_2} \otimes \cdots \otimes w_{j_p}$ may be regarded as an unbreakable expression. We set $T^0W = \mathbb{F}, T^1W = W$.

Next we define $\Lambda^q W$ as the exterior product of W with itself q times:

$$\Lambda^q W \;=\; \left\{ \sum_{\substack{\underline{j}=(j_1,j_2,\ldots,j_q) \\ 1\le j_1<j_2<\cdots<j_q\le k}} a_{\underline{j}} \cdot (w_{j_1} \wedge w_{j_2} \wedge \cdots \wedge w_{j_q}) \mid a_{\underline{j}} \in \mathbb{F} \right\},$$

where once again, the sum is a formal sum and $w_{j_1} \wedge w_{j_2} \wedge \cdots \wedge w_{j_q}$ is to be regarded as an unbreakable expression. Here as well, we set $\Lambda^0 W = \mathbb{F}, \Lambda^1 W = W$. Clearly $\dim(T^p W) = k^p$ and $\dim(\Lambda^q W) = \binom{k}{q}$. For any vector space S over $\mathbb{F}$, we define the dual space S^* to be the space of all functionals from S to $\mathbb{F}$. As is well-known, in the finite-dimensional case, S is isomorphic to S^*.

Structure of the Moulin code Given any set of integers (n, k, d, s) satisfying:

$$1 \;\le\; (s-1) \;\le\; k \;\le\; d \;\le (n-1),$$

there exists a Moulin ER $\{(n,k,d),(\alpha,\beta),B,\mathbb{F}\}$ RGC over a finite field $\mathbb{F}$ where

$$\begin{aligned}
\alpha &= \sum_{\substack{p\ge 0, q\ge 0 \\ p+q+1=s}} (d-k)^p \binom{k}{q}, \\
\beta &= \sum_{\substack{p\ge 0, q\ge 0, \\ p+q+2=s}} (d-k)^p \binom{k-1}{q}, \\
B &= \sum_{\substack{p\ge 0, q\ge 0, \\ p+q+1=s}} d(d-k)^p \binom{k}{q} - \sum_{\substack{p,q: p\ge 0, q\ge 0, \\ p+q=s}} (d-k)^p \binom{k}{q}, \qquad (7.6)
\end{aligned}$$

and where the field size $|\mathbb{F}|$ satisfies $|\mathbb{F}| \ge n$. In the description above, s is an auxiliary parameter of the Moulin-code construction. Thus one sees from (7.5) and (7.6), after setting $s-1=\mu$, that the Moulin code and Cascade codes share identical parameters.

The data file or equivalently, codeword being stored, is identified with a linear functional ϕ acting on the vector space $\mathbb{M}$ given by

$$\mathbb{M} \;=\; \bigoplus_{\substack{p\ge 0, q\ge 0 \\ p+q+1=s}} T^p V \otimes U \otimes \Lambda^q W.$$

Thus ϕ is an element of the dual space

$$\begin{aligned}\mathbb{M}^* &= \left(\bigoplus_{\substack{p\geq 0,q\geq 0\\ p+q+1=s}} T^pV\otimes U\otimes \Lambda^qW\right)^*\\ &= \bigoplus_{\substack{p\geq 0,q\geq 0\\ p+q+1=s}} (T^pV\otimes U\otimes \Lambda^qW)^*.\end{aligned}$$

When $d=k$, the summand $(T^pV\otimes U\otimes\Lambda^qW)$ reduces to $(W\otimes\Lambda^qW)$ because V vanishes and U degenerates to W. The codewords in the resultant (n,k,k) code corresponding to functionals acting on the direct sum

$$\mathbb{M}_{(n,k,k)} = \bigoplus_{0\leq q\leq (s-1)} W\otimes\Lambda^qW.$$

It is explained in [54] how the Moulin code construction for the general (n,k,d) case, can be viewed as being made up layers of (n,k,k) Moulin codes.

File Size Computation To estimate the file size B, we follow [54] and introduce the following terminology:

$$\begin{aligned}V\text{-spaces:}&\quad \{T^pV\otimes\Lambda^qW \mid p\geq 0,q\geq 0,p+q=s\},\\ U\text{-spaces:}&\quad \{T^pV\otimes U\otimes\Lambda^qW \mid p\geq 0,q\geq 0,p+q+1=s\},\\ W\text{-spaces:}&\quad \{T^pV\otimes W\otimes\Lambda^qW \mid p\geq 0,q\geq 0,p+q+1=s\}.\end{aligned}$$

As mentioned above, each codeword is associated to a functional ϕ belonging to the dual space

$$\mathbb{M}^* = \bigoplus_{\substack{p\geq 0,q\geq 0\\ p+q+1=s}} (T^pV\otimes U\otimes\Lambda^qW)^*.$$

However, we do not pick every functional within $\mathbb{M}^*$, but only those that satisfy certain parity-checks.

For any $p\geq 0, q\geq 1$, we define a transformation, termed as cowedge multiplication, by $\nabla : T^pV\otimes\Lambda^qW\to T^pV\otimes W\otimes\Lambda^{q-1}W$ in a recursive manner. Notice that the domain and range of ∇ are elements belonging

to V-spaces and W-spaces respectively. Let $\nu \in T^pV$, $\omega \in \Lambda^qW$. For $q = 1$,

$$\nabla(\nu \otimes \omega) = \nu \otimes \omega,$$

and for $q = 2$, $\omega = w_1 \wedge w_2$ for some $w_1, w_2 \in W$,

$$\begin{aligned} \nabla(\nu \otimes \omega) &= \nabla(\nu \otimes (w_1 \wedge w_2)) \\ &= \nu \otimes w_1 \otimes w_2 - \nu \otimes w_2 \otimes w_1. \end{aligned}$$

For $q > 2$, $\omega = \omega_1 \wedge w_1$ for some $\omega_1 \in \Lambda^{q-1}W, w_1 \in W$,

$$\begin{aligned} \nabla(\nu \otimes \omega) &= \nabla(\nu \otimes \omega_1 \wedge w_1) \\ &= \nabla(\nu \otimes \omega_1) \wedge w_1 + (-1)^{q-1}\nu \otimes w_1 \otimes \omega_1. \end{aligned}$$

The parity-check constraints are then given by the following equations. For every $\nu \in T^pV$ and $\omega \in \Lambda^qW$ for all possible $p \geq 1, q \geq 1$ and $p + q = s$, the following constraints need to be satisfied:

$$\phi(\nu \otimes w) = \phi(\nabla(\nu \otimes w)). \tag{7.7}$$

We have the additional constraints

$$\phi(\nabla(\omega)) = 0, \text{ for } \omega \in \Lambda^sW, \tag{7.8}$$

$$\phi(\nu) = 0, \text{ for } \nu \in T^sV. \tag{7.9}$$

Thus a data file is represented by a functional in $\mathbb{M}^*$ that satisfies (7.7), (7.8) and (7.9). The file-size can be obtained by subtracting from $\dim(\mathbb{M}^*)$ the number of linearly-independent parity-check constraints. The quantity $\dim(\mathbb{M}^*) = \dim(\mathbb{M})$ is the sum of dimensions of vector spaces that belong to the class of U-spaces. The p-c constraints are in one-one correspondence with the elements of the V-spaces. Thus by subtracting the dimensionality of the sum of the V-spaces from the dimensionality of $\mathbb{M}$, we obtain the lower bound on file size B given below

$$B \geq \sum_{\substack{p\geq 0, q\geq 0, \\ p+q+1=s}} d(d-k)^p \binom{k}{q} - \sum_{\substack{p\geq 0, q\geq 0, \\ p+q=s}} (d-k)^p \binom{k}{q}. \tag{7.10}$$

Node Contents Let A be a $(d \times n)$ matrix over $\mathbb{F}$ having the property that

- Any d columns of A are linearly independent and
- If B is the $(k \times n)$ sub-matrix of A obtained by selecting the first k rows of A, then any k columns of B are linearly independent.

We identify the column space of A with the vector space U, both having dimension d over $\mathbb{F}$. Let $\{u_1, i = 1, 2, \cdots, n\}$, be the elements in U associated with the columns of A. In the Moulin-code construction, the ith node stores

$$\{\phi(v) \mid v \in \bigoplus_{\substack{p\geq 0, q\geq 0 \\ p+q+1=s}} (T^p V \otimes u_i \otimes \Lambda^q W)\},$$

where ϕ is the functional associated to the data file stored. The number of symbols stored is thus given by

$$\alpha = \sum_{\substack{p\geq 0, q\geq 0 \\ p+q+1=s}} (d-k)^p \binom{k}{q}. \tag{7.11}$$

Repair Download In the Moulin-code construction, the β repair symbols sent by helper node h to replace node f are given by:

$$\{\phi(v) \mid v \in \partial_{u_f}^U (T^p V \otimes u_h \otimes \Lambda^q W), \text{ such that } p+q+2 = s\},$$

where $\partial_{u_f}^U$ is a certain co-boundary operator ([54] for details). It is shown (in [54]) that the number of symbols transferred to the helper node is given by

$$\beta = \sum_{\substack{p\geq 0, q\geq 0, \\ p+q+2=s}} (d-k)^p \binom{k-1}{q}. \tag{7.12}$$

Notes

1. Early interior-point constructions: The first IP-ER RGC having parameters (n, d, d) was constructed in [202], [236] by the process

of carefully layering codewords belonging to an MDS code having parameters $[w + n - d, w]$ where $w \geq 2$ is an auxiliary parameter. This code was shown to achieve a point on the FR tradeoff in the near-MSR region. The construction makes use of block designs and can be extended to build (n, k, d) IP-ER RGCs by precoding the symbols using an MDS code whose code symbols correspond to the evaluation of linearized polynomials. This linearized-polynomial approach, results however, in significantly large field-size. In [210], a thumb-rule to transform an (n, d, d) code to an (n, k, d) without field-size increase is presented, that increases instead, the value of sub-packetization level α. The resultant code construction, referred to as the Improved Layered Code construction, turns out to yield codes achieving the entire ER tradeoff for the case $(n, k = 3, d = n - 1)$. The construction also achieves a single interior point on the ER tradeoff associated with the parameter set $(n, k = 4, d = n - 1)$.

8

Lower Bounds on Sub-Packetization Level of MSR Codes

As has been the case in much of this monograph, we will restrict our attention in this section to linear RGCs, where linearity is as defined in Section 3. We will present two lower bounds[1] on the sub-packetization level α of an MSR code having parameters

$$\Big\{ (n, k, d = (n-1)),\ (\alpha = (n-k)\beta,\ \beta),\ B = \alpha k,\ \mathbb{F}_q \Big\}.$$

The first bound [11] is applicable to MSR codes possessing the optimal-access property and builds on the derivation of a prior bound, applicable in the case of systematic-node repair and appearing in [233]. The second bound [6] applies to MSR codes in general. More recently, in independent work, the results in [6] have been improved upon in [7] and [9]. This improved bound is briefly discussed in the notes subsection. We set $r = (n-k)$ throughout the section.

[1] A brief overview of known bounds on sub-packetization for the case $d < (n-1)$ is given in the notes subsection.

8.1 Properties of Repair Subspaces

We begin with some helpful notation. We assume throughout this section that the linear MSR code is encoded in systematic form, that the n nodes are ordered, and that the first k nodes, denoted by $\{u_1, u_2, \cdots, u_k\}$ store the message symbols, while the remaining r nodes denoted by $\{p_1, p_2, \cdots, p_r\}$ store parity. Let $\ell,\ 2 \leq \ell \leq (k-1)$, be an integer and set

$$\begin{aligned} U &= \{u_1, u_2, \cdots, u_\ell\}, \\ V &= \{u_{\ell+1}, u_{\ell+2}, \cdots, u_k\}, \\ P &= \{p_1, p_2, \cdots, p_r\}. \end{aligned}$$

While the sets U, V are clearly functions of ℓ, for much this section, ℓ can be regarded as a fixed integer and for this reason, we use the simplified notation above. Note that our choice of ℓ ensures that neither U nor V is empty.

Generator Matrix of MSR Code

Since the MSR code $\mathcal{C}$ is systematic, its generator matrix G can be expressed in the form:

$$G = \begin{bmatrix} I_\alpha & 0 & \dots & 0 \\ 0 & I_\alpha & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & I_\alpha \\ A_{p_1u_1} & A_{p_1u_2} & \dots & A_{p_1u_k} \\ A_{p_2u_1} & A_{p_2u_2} & \dots & A_{p_2u_k} \\ \vdots & \vdots & \vdots & \vdots \\ A_{p_ru_1} & A_{p_ru_2} & \dots & A_{p_ru_k} \end{bmatrix}, \tag{8.1}$$

where each A_{ij} is an $(\alpha \times \alpha)$ sub-matrix. The data-collection property of an MSR code forces every $(k \times k)$ block sub-matrix of G to be invertible. From this and elementary (block) row reduction, it follows that each sub-matrix A_{ij} is necessarily invertible.

Rows $[(i-1)\alpha + 1, i\alpha]$ of G are associated to the α contents of the ith node in the list of nodes $\{u_1, \dots, u_k, p_1, \dots, p_r\}$. Each codeword in

$\mathcal{C}$ can then be expressed in the form:

$$G\underline{m} \;=\; [\underline{c}_{u_1}^T \;\ldots\; \underline{c}_{u_k}^T \;\underline{c}_{p_1}^T \;\ldots\; \underline{c}_{p_r}^T]^T,$$

where $\underline{m} \in \mathbb{F}_q^k$ is the underlying vector of $k\alpha$ message symbols and $\underline{c}_{u_i}$ is the code symbol associated to node u_i. Let $\underline{c}_{u_i}^T = [c_{i1}, c_{i2}, \ldots, c_{i\alpha}]$.

Repair Matrices and Subspaces

Since the RGC is assumed to be linear, β symbols transferred from node p to node u for the repair of node u are of the form $S_{pu}\underline{c}_p$ where S_{pu} is a $(\beta \times \alpha)$ matrix. For any matrix A, we use the notation $\widehat{A}$ to refer to the subspace spanned by the rows of matrix A. We will refer to any matrix of the form S_{pu} as a repair matrix and the associated subspace $\widehat{S}_{pu}$ as a repair subspace.

Interference Alignment

In the repair of an MSR code, a phenomenon called interference alignment takes place. While the explanation of the phenomenon presented below is for the case $d = (n-1)$ that is the focus of the present section, the principle extends to the case $d < (n-1)$.

Consider the repair of a systematic node u_i. The helper information passed on by the xth parity node p_x is a collection of β symbols, each of which is a linear combination of the $k\alpha$ message symbols $\{c_{jm} \mid 1 \leq j \leq k,\ 1 \leq m \leq \alpha\}$ making up the data file. Of this, only the portion of this information that is a linear combination of the α symbols $\{c_{im} \mid 1 \leq m \leq \alpha\}$ contributes directly to the reconstruction of the contents of the failed node.

The helper information passed on by the jth, systematic node $u_j, j \neq i$, plays only an indirect role in the reconstruction of node u_i. This is because, the set of $k\alpha$ message symbols constitutes a collection of $k\alpha$ independent scalar random variables. It follows that the role of the jth systematic node $u_j, j \neq i$, in the repair process, is to supply a set of β symbols, that will allow the undesired contribution from each parity node that is a linear combination of the contents of the jth systematic node, to be cancelled out. For this to happen, the repair information passed on by the r parity nodes must be linearly aligned

so as to permit such cancellation by a collection of just β symbols from the systematic node u_j. This requirement is one of two conditions referred to as interference alignment. The other requirement is that the component of helper information passed on by the r parity nodes that is a linear combination of the contents $\{c_{im} \mid 1 \leq m \leq \alpha\}$ of the ith node, suffices for reconstruction of the failed node.

The lemma below formally phrases in matrix form these interference alignment conditions. Given a subspace S and a matrix A, we define the vector space $SA := \{v^T A \mid v \in S\}$.

Lemma 4 (Interference Alignment)**.** With notation as introduced above, for every $1 \leq i, j \leq k$, $j \neq i$, we must have

(a) (interference cancellation condition)

$$\widehat{S}_{u_j u_i} = \widehat{S}_{p_1 u_i} A_{p_1 u_j} = \cdots = \widehat{S}_{p_r u_i} A_{p_r u_j}.$$

(b) (full-rank condition)

$$\text{rank} \left(\begin{bmatrix} S_{p_1 u_i} A_{p_1 u_i} \\ \vdots \\ S_{p_r u_i} A_{p_r u_i} \end{bmatrix} \right) = \alpha.$$

The lemma appears in [233]. A formal proof can be found for example, in [11].

We will now use Lemma 4 to prove a second Lemma, Lemma 5 that appears in [11] and which deals with the intersection of repair subspaces. Lemma 5 will be used in turn, to establish the lower bounds on the sub-packetization level of an optimal-access MSR code. The statement of Lemma 5 is illustrated in Fig 8.1. Recall that $2 \leq \ell \leq (k-1)$,

$$\begin{aligned} U &= \{u_1, u_2, \cdots, u_\ell\}, \\ V &= \{u_{\ell+1}, u_{\ell+2}, \cdots, u_k\}, \\ P &= \{p_1, p_2, \cdots, p_r\}. \end{aligned}$$

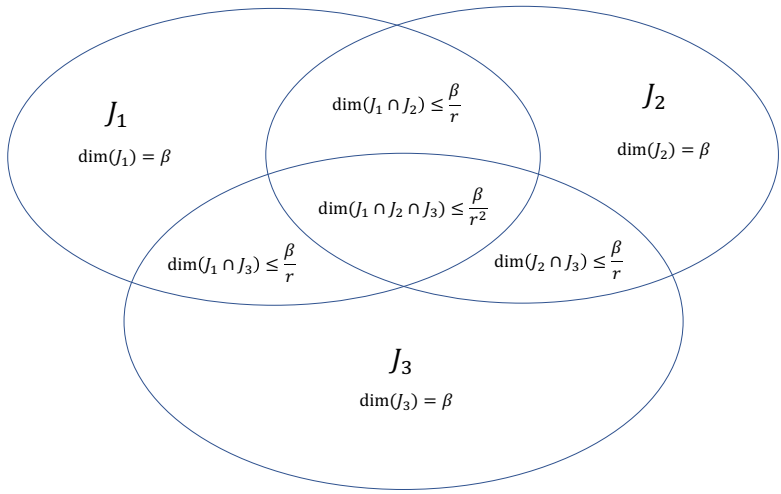

Figure 8.1: The figure illustrates the inequality appearing in Lemma 5, equation (8.3). Each of the three subspaces $\left\{ J_i := \widehat{S}_{pu_i} \mid i = 1, 2, 3 \right\}$ has dimension β. Their pairwise intersection has dimension $\leq \frac{\beta}{r}$ that is smaller by at least a factor of r. The intersection of all three of the subspaces has dimension $\leq \frac{\beta}{r^2}$ that is thus smaller by at least a factor of r^2.

Repair Subspace Intersection

Lemma 5 (Repair Subspace Intersection)**.**

$$\sum_{i=1}^{r} \dim \left(\bigcap_{u \in U} \widehat{S}_{p_i u} \right) \leq \dim \left(\bigcap_{u \in U \setminus \{u_\ell\}} \widehat{S}_{pu} \right), \tag{8.2}$$

where p is an arbitrary node in P, i..e, an arbitrary parity node. Furthermore, $\dim \left(\bigcap_{u \in U} \widehat{S}_{pu} \right)$ is the same for all $p \in P$. Hence for any $p \in P$,

$$\dim \left(\bigcap_{u \in U} \widehat{S}_{pu} \right) \leq \frac{\dim \left(\bigcap_{u \in U \setminus \{u_\ell\}} \widehat{S}_{pu} \right)}{r}. \tag{8.3}$$

Proof.

(a) *Invariance of Dimension of ℓ-fold Intersection of Repair Subspaces Contributed by a Parity Node*

Clearly, the nodes in $U \cup V$ are the systematic nodes and the nodes in P are the parity nodes. We will first prove that $\dim\left(\bigcap_{u\in U} \widehat{S}_{pu}\right)$ is the same for all $p \in P$, i.e., that the dimension of intersection of the ℓ subspaces $\{\widehat{S}_{pu}\}_{u\in U}$ obtained by varying the failed node $u \in U$ is the same, regardless of the parity node $p \in P$ from which the helper data originates.

By Lemma 4, $\forall p, p' \in P$ and $u_j \in U$,

$$\widehat{S}_{pu_j} A_{pu_{\ell+1}} = \widehat{S}_{p'u_j} A_{p'u_{\ell+1}}. \tag{8.4}$$

Since A_{ij} are invertible for all i, j, equation (8.4) implies $\forall p, p' \in P$:

$$\left(\bigcap_{j=1}^{\ell} \widehat{S}_{pu_j}\right) A_{pu_{\ell+1}} = \left(\bigcap_{j=1}^{\ell} \widehat{S}_{p'u_j}\right) A_{p'u_{\ell+1}}. \tag{8.5}$$

It follows then from non-singularity of the matrices A_{ij} and equation (8.5) that $\dim(\bigcap_{u\in U} \widehat{S}_{pu})$ is same for all $p \in P$. Now it remains to prove the main inequality (8.2).

(b) *$(\ell-1)$-fold Intersection of Repair Subspaces*

We proceed similarly in the case of an $(\ell-1)$-fold intersection, replacing ℓ by $\ell-1$ in (8.5). We will then obtain, $\forall p, p' \in P$:

$$\left(\bigcap_{j=1}^{\ell-1} \widehat{S}_{pu_j}\right) A_{pu_{\ell}} = \left(\bigcap_{j=1}^{\ell-1} \widehat{S}_{p'u_j}\right) A_{p'u_{\ell}}. \tag{8.6}$$

(c) *Relating ℓ-fold and $(\ell-1)$-fold Intersections*

Next consider the repair of the node u_ℓ. It follows from (8.6) that for any $p', p \in P$:

$$\begin{aligned}\left(\bigcap_{j=1}^{\ell} \widehat{S}_{pu_j}\right) A_{pu_{\ell}} &= \widehat{S}_{pu_\ell} A_{pu_\ell} \bigcap \left(\left(\bigcap_{j=1}^{\ell-1} \widehat{S}_{pu_j}\right) A_{pu_\ell}\right) \\ &\subseteq \left(\bigcap_{j=1}^{\ell-1} \widehat{S}_{p'u_j}\right) A_{p'u_\ell}. \end{aligned} \tag{8.7}$$

As a consequence of (8.7) it follows that for any $p \in P$:

$$\bigoplus_{i=1}^{r} \left(\bigcap_{u \in U} \widehat{S}_{p_i u} \right) A_{p_i u_\ell} \quad \subseteq \quad \left(\bigcap_{j=1}^{\ell-1} \widehat{S}_{p u_j} \right) A_{p u_\ell}. \tag{8.8}$$

By Lemma 4, we must have that

$$\operatorname{rank} \left(\left[\begin{array}{c} \widehat{S}_{p_1 u_\ell} A_{p_1 u_\ell} \\ \widehat{S}_{p_2 u_\ell} A_{p_2 u_\ell} \\ \widehat{S}_{p_3 u_\ell} A_{p_3 u_\ell} \\ \vdots \\ \widehat{S}_{p_r u_\ell} A_{p_r u_\ell} \end{array} \right] \right) \quad = \quad \alpha. \tag{8.9}$$

It follows as a consequence that

$$\bigoplus_{i=1}^{r} \widehat{S}_{p_i u_\ell} A_{p_i u_\ell} \quad = \quad \mathbb{F}_q^{\alpha}, \tag{8.10}$$

and hence for every $p \in P$, we must have that

$$\widehat{S}_{p u_\ell} A_{p u_\ell} \bigcap \bigoplus_{p' \in P, p' \neq p} \widehat{S}_{p' u_\ell} A_{p' u_\ell} \quad = \quad \{\underline{0}\}. \tag{8.11}$$

It follows from (8.11) that if we set

$$W_i \quad := \quad \left(\bigcap_{u \in U} \widehat{S}_{p_i u} \right) A_{p_i u_\ell}, \; 1 \leq i \leq r,$$

that

$$W_j \bigcap \left(\sum_{1 \leq i,j \leq r, i \neq j} W_i \right) \quad = \quad \{\underline{0}\}. \tag{8.12}$$

Hence, since the A_{ij}'s are non-singular, from equations (8.8) and (8.12) we can conclude that:

$$\sum_{i=1}^{r} \dim \left(\bigcap_{u \in U} \widehat{S}_{p_i u} \right) \; \leq \; \dim \left(\bigcap_{u \in U \setminus \{u_\ell\}} \widehat{S}_{p u} \right), \tag{8.13}$$

for any $p \in P$, which is precisely the desired equation (8.2).

□

8.2 Lower Bound for Optimal-Access MSR Codes

In this subsection, we present the lower bound for the sub-packetization level of optimal-access MSR codes due to Balaji and Kumar [11], that makes use of Lemma 5. In [233], Tamo *et al.* showed that $\alpha \geq r^{\frac{k-1}{r}}$ must hold in any systematic vector MDS code that is able to repair in help-by-transfer (optimal-access) fashion, the k systematic nodes. The lower bound from [11] that we derive below can be viewed as an extension of this lower bound to the case of all node, optimal-access repair.

In an optimal-access MSR code, the rows of the repair matrices are picked from among the standard basis vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_\alpha\}$. Lemma 5 above, presented an upper bound on the dimension of the intersection of repair subspaces. This places an upper bound on the number of repair subspaces that contain a fixed standard basis vector $\mathbf{e}_i$. There are only α standard basis elements in all, and each of the $(n-1)$ repair matrices $\{S_{n1}, \ldots, S_{n(n-1)}\}$ contains β of them.

This leads to a lower bound on α by means of counting in two ways, the inclusion relation between standard basis vectors and repair matrices in the light of the fact that number of repair matrices which contains a fixed standard basis element $\mathbf{e_i}$ is upper bounded. The argument is illustrated in Fig. 8.2.

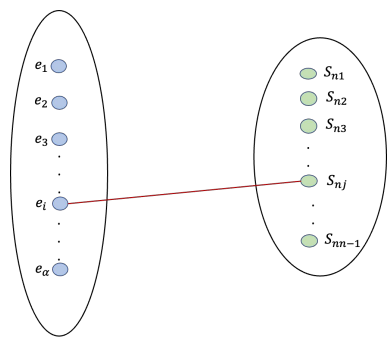

Figure 8.2: The bipartite graph appearing in the counting argument used to prove Theorem 4 is shown. Each node on the left corresponds to an element of the standard basis $\{\mathbf{e}_1, ..., \mathbf{e}_\alpha\}$. The nodes to the right are associated to the repair matrices $S_{n1}, ..., S_{nn-1}$. An edge connecting the vector $\mathbf{e}_i$ to node S_{nj} is drawn if $\mathbf{e}_i$ is a row vector of the repair matrix S_{nj}.

Theorem 4 (Bound on sub-packetization level of an optimal-access MSR code [11])**.** Let $\mathcal{C}$ be a linear optimal-access MSR code having parameter set

$$\{(n, k, d = (n-1)),\ (\alpha, \beta),\ B,\ \mathbb{F}_q)\}\,.$$

Then we must have:

$$\alpha \geq \min\{r^{\lceil \frac{n-1}{r} \rceil}, r^{k-1}\}.$$

Proof. (a) *Invariance of Repair Matrices to Choice of Generator Matrix*

We first observe that the repair matrices can be kept constant, even if the generator matrix of the code changes. This is because the repair matrices only depend upon relationships that hold among code symbols of any codeword in the code and are independent of the particular generator matrix used in encoding. In particular, the repair matrices are insensitive to the characterization of a particular node as being either a systematic or a parity node.

(b) *Implications for the Dimension of the Repair Subspace*

From Lemma 5, we have that

$$\begin{aligned} \dim\left(\bigcap_{u\in U} \widehat{S}_{pu}\right) &\leq \frac{\dim\left(\bigcap_{u\in U\setminus\{u_\ell\}} \widehat{S}_{pu}\right)}{r} \\ &\leq \frac{\dim\left(\bigcap_{u\in U\setminus\{u_\ell, u_{\ell-1}\}} \widehat{S}_{pu}\right)}{r^2} \\ &\leq \frac{\dim\left(\widehat{S}_{pu_1}\right)}{r^{\ell-1}} \\ &= \frac{\alpha}{r^\ell} = \frac{\alpha}{r^{|U|}}. \end{aligned} \tag{8.14}$$

Lemma 5 and its proof holds true for any set $U \subseteq [n]$ of size $2 \leq |U| \leq (k-1)$. As a result, equation (8.14), also holds for any set $U \subseteq [n]$ of size $2 \leq |U| \leq (k-1)$. We would like to extend the above inequality to hold even for the case when U is replaced by a set F of size $k \leq |F| \leq (n-1)$. Since the repair

matrices and their associated subspaces are invariant to the choice of generator matrix, from here onwards, we drop the distinction between systematic and parity nodes. In place of using the labels $\{u_1, u_2, \cdots, u_k\}$, $\{p_1, p_2, \cdots, p_r\}$, we will simply use the integers from 1 to n to denote the n nodes. It will be convenient in the argument, to assume that F is a collection of nodes having size $k \leq |F| \leq (n-1)$ that does not contain the nth node.

Let us suppose that $\alpha < r^{k-1}$. We will show that this assumption leads to $\alpha \geq r^{\lceil \frac{n-1}{r} \rceil}$, thereby proving the Theorem. If $\alpha < r^{k-1}$ and F is of size $(k-1)$, we get:

$$\dim\left(\bigcap_{u\in F} \widehat{S}_{nu}\right) \leq \frac{\alpha}{r^{k-1}} < 1, \tag{8.15}$$

which is possible iff

$$\dim\left(\bigcap_{u\in F} \widehat{S}_{nu}\right) = 0.$$

But this would imply that

$$\dim\left(\bigcap_{u\in F} \widehat{S}_{nu}\right) = 0.$$

for any subset F of nodes having size $(k-1) \leq |F| \leq (n-1)$. We are therefore justified in extending the inequality in (8.14) to the case when U is replaced by a subset F whose size now ranges from 2 to $(n-1)$, i.e.,

$$\dim\left(\bigcap_{u\in F} \widehat{S}_{nu}\right) \leq \frac{\alpha}{r^{|F|}} \tag{8.16}$$

for any $F \subseteq [n-1]$ with $2 \leq |F| \leq (n-1)$. A consequence of the inequality (8.16) is that

$$\dim\left(\bigcap_{u\in F} \widehat{S}_{nu}\right) \geq 1$$

implies that

$$|F| \leq \lfloor \log_r(\alpha) \rfloor. \tag{8.17}$$

Thus any given non-zero vector can belong to at most $\lfloor \log_r(\alpha) \rfloor$ repair subspaces among the repair subspaces $\{\widehat{S}_{n1}, \ldots, \widehat{S}_{nn-1}\}$.

(c) *Counting in a Bipartite Graph*

The remainder of the proof then follows the steps outlined in [233]. We form a bipartite graph with the standard basis vectors $\{\mathbf{e}_1, ..., \mathbf{e}_\alpha\}$ as the set of left nodes and $\{S_{n1}, ..., S_{nn-1}\}$ as the set of right nodes as shown in Fig. 8.2. We place an edge $(\mathbf{e}_i, S_{nj})$ in the edge set of this bipartite graph iff $\mathbf{e}_i \in \widehat{S}_{nj}$. Now since the MSR code is an optimal-access code, the rows of each repair matrix S_{nj} must all be drawn from the set $\{\mathbf{e}_1, ..., \mathbf{e}_\alpha\}$.

Counting the number of edges of this bipartite graph in terms of node degrees on the left and the right, we obtain from (8.17):

$$\begin{aligned} \alpha \lfloor \log_r(\alpha) \rfloor &\geq (n-1)\frac{\alpha}{r}, \\ \log_r(\alpha) &\geq \lfloor \log_r(\alpha) \rfloor \geq \left\lceil \frac{(n-1)}{r} \right\rceil, \\ \log_r(\alpha) &\geq \left\lceil \frac{(n-1)}{r} \right\rceil, \\ \alpha &\geq r^{\lceil \frac{n-1}{r} \rceil}. \end{aligned}$$

Thus we have shown that if $\alpha < r^{k-1}$, we must have $\alpha \geq r^{\lceil \frac{n-1}{r} \rceil}$. It follows that

$$\alpha \geq \min\{r^{\lceil \frac{n-1}{r} \rceil}, r^{k-1}\}.$$

□

8.3 Lower Bound for General MSR Codes

We present in this subsection, a lower bound on sub-packetization level that is exponential in the code dimension given by Alrabiah and Guruswami [6]. Throughout this subsection we set $U \cup V = [1, k]$ and $P = [k+1, n]$ and further we assume that the MSR codes discussed are defined over the finite field $\mathbb{F}_q$. We begin with a lemma from [233].

Lemma 6. [233] If there exists an $\{(n, k, d = (n-1)), (\alpha, \beta)\}$ MSR code with repair matrices $\{S_{pu} : u \in [k], p \in [n] \setminus \{u\}\}$, then it is

possible to construct an $\{(n-1, k-1, d=(n-2)), (\alpha, \beta)\}$ MSR code with new repair matrices $\{S'_{pu} : u \in [k-1], p \in [n-1] \setminus \{u\}\}$ with $S'_u = S'_{pu}, \forall u \in [k-1], p \in [n-1] \setminus \{u\}$ i.e., repair matrices of the new MSR code do not depend on p.

A proof of the lemma can be found in [233]. We now present some definitions.

Definition 5. Let $\{V_1, \ldots, V_t\}$ be a set of t, γ_1-dimensional subspaces of $\mathbb{F}_q^\alpha$ and let $\{W_1, \ldots, W_t\}$ be a set of t, γ_2-dimensional subspaces of $\mathbb{F}_q^\alpha$. Let $\gamma_2 \leq \gamma_1 \leq \alpha$. We define:

$$\begin{aligned}
&\mathcal{F}(V_1 \to W_1, V_2 \to W_2, \ldots, V_t \to W_t) \\
&= \{\psi : \psi \text{ is a } (\alpha \times \alpha) \text{ matrix over } \mathbb{F}_q \text{ and } V_i\psi \subseteq W_i, \forall 1 \leq i \leq t\}, \\
&\mathcal{I}(V_1 \to W_1, V_2 \to W_2, \ldots, V_t \to W_t) \\
&= \dim(\mathcal{F}(V_1 \to W_1, V_2 \to W_2, \ldots, V_t \to W_t)), \\
&\mathcal{F}(V_1, V_2, \ldots, V_t) \\
&= \mathcal{F}(V_1 \to V_1, V_2 \to V_2, \ldots, V_t \to V_t), \\
&\mathcal{I}(V_1, \ldots, V_t) \\
&= \mathcal{I}(V_1 \to V_1, V_2 \to V_2, \ldots, V_t \to V_t).
\end{aligned}$$

Lemma 7. [6] Let $U_1, U_2, \ldots, U_s$ be s γ-dimensional subspaces of $\mathbb{F}_q^\alpha$ such that $\cap_{i=1}^s U_i = \{\underline{0}\}$. Then

$$\sum_{i=1}^{s} \dim(U_i) \quad \leq \quad (s-1)\dim\left(\sum_{i=1}^{s} U_i\right).$$

Proof. We prove by induction. For the $s = 2$ case,

$$\begin{aligned}
\dim(U_1) + \dim(U_2) &= \dim(U_1 + U_2) + \dim(U_1 \cap U_2) \\
\dim(U_1) + \dim(U_2) &= (2-1) \times \dim(U_1 + U_2).
\end{aligned}$$

We now assume the inequality holds for $s = \ell$ and prove it also holds for $s = \ell + 1$.

$$\begin{aligned}
&\dim(U_1) + \dim(U_2) + \dim(U_3) + \ldots + \dim(U_{\ell+1}) \\
&\quad = \dim(U_1 + U_2) + \dim(U_1 \cap U_2) + \dim(U_3) + \ldots + \dim(U_{\ell+1}) \\
&\quad \leq \dim(U_1 + U_2) + (\ell - 1) \times \dim(U_1 \cap U_2 + U_3 + \ldots + U_{\ell+1}) \\
&\quad \leq \ell \times \dim(U_1 + U_2 + U_3 + \ldots + U_{\ell+1}).
\end{aligned}$$

□

Lemma 8. [6] For any $\{(n, k, d = n-1), (\alpha, \beta)\}$ MSR code with repair matrices $\{S_{pu} : u \in [k], p \in [n] \setminus \{u\}\}$ with $S_u = S_{pu}$ for all $u \in [k]$ and $p \in [n] \setminus \{u\}$ i.e., repair matrix S_{pu} does not depend on p,

$$\mathcal{I}(\widehat{S}_1, \ldots, \widehat{S}_t) \quad \leq \quad \left(\frac{2r-1}{2r}\right) \mathcal{I}(\widehat{S}_1, \ldots, \widehat{S}_{t-1}),$$

where $1 \leq t \leq k$.

Proof. Let $p \in [k+1, n]$. From Lemma 4 and invertibility of A_{pt}, the following left multiplication is a vector-space isomorphism:

$$L_{A_{pt}^{-1}} \quad : \quad \mathcal{F}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t) \to \mathcal{F}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t A_{pt} \to \widehat{S}_t).$$

where $L_{A_{pt}^{-1}}(\psi) = A_{pt}^{-1}\psi$. This can be argued as follows . For $i < t$, by Lemma 4,

$$L_{A_{pt}^{-1}}(\psi)(\widehat{S_i}) \quad = \quad \widehat{S_i} A_{pt}^{-1}\psi \quad = \quad \widehat{S_i}\psi \quad \subseteq \quad \widehat{S_i},$$

and for $i = t$,

$$L_{A_{pt}^{-1}}(\psi)(\widehat{S_t A_{pt}}) \quad = \quad \widehat{S_t} A_{pt} A_{pt}^{-1}\psi \quad = \quad \widehat{S_t}\psi \quad \subseteq \quad \widehat{S_t}.$$

Hence,

$$\mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t) \quad = \quad \mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t A_{pt} \to \widehat{S}_t).$$

Similarly left multiplication by inverse of A_{pt} coupled with right multiplication by $A_{(k+i)t}$ is also an isomorphism. Hence we have,

$$\begin{aligned}
\mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t) &= \mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t A_{pt} \to \widehat{S}_t A_{(k+1)t}), \\
\mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t) &= \mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t A_{pt} \to \widehat{S}_t A_{(k+2)t}).
\end{aligned}$$

Let,

$$\begin{aligned}
V_{pt} &= \mathcal{F}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t A_{pt} \to \widehat{S}_t A_{(k+1)t}), \\
W_{pt} &= \mathcal{F}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t A_{pt} \to \widehat{S}_t A_{(k+2)t}).
\end{aligned}$$

This implies,

$$2r \times \mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t) = \sum_{p=k+1}^{n} \dim(V_{pt}) + \sum_{p=k+1}^{n} \dim(W_{pt}). \quad (8.18)$$

From Lemma 4 (full-rank condition), we also have that

$$\begin{aligned}\bigcap_{p=k+1}^{n} V_{pt} &= \mathcal{F}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \mathbb{F}_q^{\alpha} \to \widehat{S}_t A_{(k+1)t}),\\ \bigcap_{p=k+1}^{n} W_{pt} &= \mathcal{F}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \mathbb{F}_q^{\alpha} \to \widehat{S}_t A_{(k+2)t}).\end{aligned}$$

Again from Lemma 4 (full-rank condition), we have that $\widehat{S}_t A_{(k+1)t} \cap \widehat{S}_t A_{(k+2)t} = \{\underline{0}\}$. Hence,

$$\left(\bigcap_{p=k+1}^{n} V_{pt}\right) \bigcap \left(\bigcap_{p=k+1}^{n} W_{pt}\right) = \{\underline{0}\}. \tag{8.19}$$

Now applying Lemma 7 to equation (8.18) using the condition in (8.19):

$$\begin{aligned}2r \times \mathcal{I}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1}, \widehat{S}_t) &= \sum_{p=k+1}^{n} \dim(V_{pt}) + \sum_{p=k+1}^{n} \dim(W_{pt})\\ &\leq (2r-1) \times \dim\left(\sum_{p=k+1}^{n} V_{pt} + \sum_{p=k+1}^{n} W_{pt}\right)\\ &\leq (2r-1) \times \dim\left(\mathcal{F}(\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_{t-1})\right)\\ &= (2r-1) I(\widehat{S}_1, \ldots, \widehat{S}_{t-1}).\end{aligned}$$

□

Theorem 5. [6] For any $\{(n, k, d = (n-1)), (\alpha, \beta)\}$ MSR code

$$\alpha \geq e^{\frac{k-1}{4r}}.$$

Proof. From Lemma 6, we can construct an $\{(n-1, k-1, d = (n-2)), (\alpha, \beta)\}$ MSR code with repair matrices not depending on p. By repeated application of Lemma 8 for this new derived MSR code,

$$I(\widehat{S}_1, \ldots, \widehat{S}_{k-1}) \leq \left(\frac{2r-1}{2r}\right)^{(k-1)} \alpha^2.$$

Since we have the identity matrix which keeps all the subspaces $\widehat{S}_i$ invariant, we have:

$$I(\widehat{S}_1, \ldots, \widehat{S}_{k-1}) \geq 1.$$

Hence we have

$$\left(\frac{2r-1}{2r}\right)^{k-1} \alpha^2 \quad \geq \quad 1.$$

By manipulation of the above inequality using $\log(1+x) \geq \frac{x}{1+x}$ for $x \geq 0$, we get the bound mentioned in the theorem. □

Notes

1. Lower bounds on the sub-packetization level of a general MSR code with $d = n-1$: The lower bound on the sub-packetization level of a general MSR code with $d = (n-1)$ given by

$$\alpha \quad \geq \quad e^{\Omega\left(\sqrt{(k-1)\log_e\left(\frac{r}{r-1}\right)}\right)},$$

appeared in [77]. This was improved in [107] to

$$\alpha \quad \geq \quad e^{\Omega\left(\sqrt{(k-1)\log_e\left(\frac{r}{r-1}\right)\log_e(r)}\right)},$$

and then improved again to the result appearing in Theorem 5. Recently, the lower bound given in Theorem 5, was further improved in two independent works [7], [9] to the following lower bound:

$$\alpha \quad \geq \quad e^{\frac{(k-1)(r-1)}{2r^2}}.$$

2. Optimal sub-packetization level of optimal-access MSR codes with $d = n-1$: Note that $\lceil \frac{n-1}{r} \rceil = \lceil \frac{n}{r} \rceil$ except when $n = 1 \mod r$. Hence for $n \neq 1 \mod r$, the sub-packetization level of the code in [200] and the CL-MSR code described in Section 5.3 matches with the lower bound given in Theorem 4, thereby determining the optimal sub-packetization level of optimal-access MSR codes for the case $d = n-1$. The case $n = 1 \mod r$ remains open however.

Open Problem 6. Determine the least possible sub-packetization level of an optimal-access (n, k, d) MSR code for the case when $d = (n-1)$ and $n = 1 \mod r$.

3. The case $d < n-1$: Lower bounds for the $d < n-1$ case can be derived by replacing $r = n-k$ with $s = d-k+1$ and replacing n with $d+1$ in the bounds presented in this section. This is because we can puncture the MSR code, retain only $d+1$ nodes, and then apply the bounds in this section. For optimal-access repair, when the choice of repair matrices does not depend on the identity of the set of remaining $d-1$ helper nodes used for repair, we have the following lower bound presented in [11],

$$\alpha \geq \min\left\{s^{\lceil \frac{n-1}{s} \rceil}, s^{k-1}\right\}.$$

Despite this assumption concerning the repair matrices, there are constructions in the literature [190], [239] of MSR codes which satisfy this assumption and have sub-packetization level achieving this lower bound.

Since the above tight bound on sub-packetization level of an optimal-access MSR code for $d < (n-1)$ is under the assumption of repair matrices that satisfy the helper-set-independent property, the following problem is still open.

Open Problem 7. Determine the minimum possible sub-packetization level of an optimal-access (n, k, d) MSR code for the case $d < n-1$.

Open Problem 8. Determine the minimum possible sub-packetization level of a general (n, k, d) MSR code.

Remark 7. Open Problems 6-8 are closely related to Open Problems 2-4. The difference is that in the earlier section, the focus was on code construction. Here it is on determining the smallest possible value of sub-packetization.

9

Variants of Regenerating Codes

9.1 MDS Codes that Trade Repair Bandwidth for Reduced Sub-Packetization Level

In this subsection, we discuss vector MDS codes that do not have minimum possible repair bandwidth and hence do not qualify to be an MSR code. These codes nevertheless offer some savings in repair bandwidth in comparison to the conventional repair of RS codes, while keeping the sub-packetization level α to a small level. The piggybacking framework introduced in [184] was one of the first such efforts. In [84], the authors introduced a family of codes that offer a choice of sub-packetization levels $\alpha = r^p$ for $1 \leq p \leq \lceil \frac{n}{r} \rceil$ over a field of size at least $n^{(r-1)\alpha+1}$, where $r = n - k$. The corresponding repair download from each helper node is given by $\beta = (1 + \frac{1}{p})r^{p-1}$. When $p = \lceil \frac{n}{r} \rceil$ these codes coincide with the MSR code construction presented in [200].

In [194], a framework termed as the ϵ-MSR framework was introduced, that enabled the construction of MDS codes that in exchange for a small increase in repair bandwidth by a multiplicative factor $(1 + \epsilon)$, offer in return, sub-packetization α that is impressively, logarithmic in n. In [138], a generic transformation for deriving MDS codes having low sub-packetization and near-optimal repair bandwidth, starting

from an MSR code is presented. More recently in [44], the Diagonal MSR code [255] was modified to obtain a vector MDS code having sub-packetization level $\alpha = u^{m+n-1}$, where $(n-k) = r = u^m$ for an integer m. The repair bandwidth of this code is shown to be asymptotically optimal for fixed r as $n \to \infty$. We discuss some of these developments below.

9.1.1 The Piggybacking Framework

The piggybacking framework introduced by Rashmi *et al.* [184] begins with a collection of α codewords drawn from an MDS code and proceeds to modify the code symbols as described below. Let $\mathcal{C}$ be an MDS code. Each individual code symbol can be regarded as a function of the message and let $(f_1(\underline{u}), f_2(\underline{u}), \ldots, f_n(\underline{u}))$ represent the codeword corresponding to message $\underline{u}$. Next, consider codewords of $\mathcal{C}$ corresponding to α distinct messages, $\{\underline{u}_1, \ldots, \underline{u}_\alpha\}$. The α code symbols $\{f_j(\underline{u}_i), i = 1, 2, \cdots, \alpha\}$ thus correspond to node j. The code is first modified by adding a function $g_{ij}(\underline{u}_1, \ldots, \underline{u}_{i-1})$ to the jth symbol of the ith codeword $f_j(\underline{u}_i)$, for all $i \in \{2, \ldots, \alpha\}, j \in \{1, \ldots, n\}$. The values so added are termed as piggybacks. Clearly, this modification does not affect our ability to decode the code, if the codewords are decoded in sequence starting with $i = 1$. Applying an invertible linear transform T_j to the α code symbols contained in the jth node, similarly does not affect our ability to decode the α codewords, nor a node's ability to serve as a helper node. By carefully choosing the piggybacking functions and the invertible linear transformations, it turns out that it is possible to reduce the repair bandwidth for the collective repair of the α MDS codewords, in comparison with the repair bandwidth needed for the conventional repair of α MDS codewords.

Three families of piggybacking-based MDS codes with reduced repair bandwidth and disk read are constructed in [184]. The piggybacking framework typically provides bandwidth savings between 25% to 50% over the conventional decoding of MDS codes.

We present in Fig. 9.1 an example of a code that illustrates the piggybacking principle. This code is the modification of a systematic $[4, 2]$ MDS code with sub-packetization level α set equal to 2 in such a

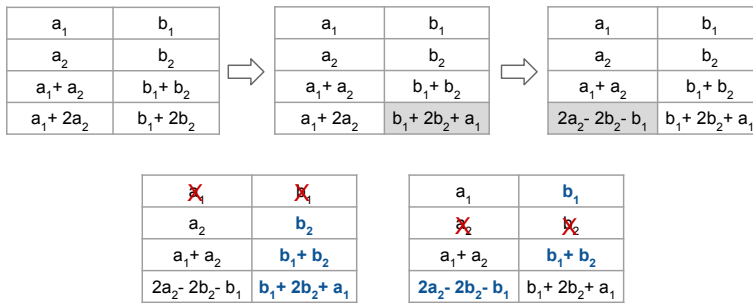

Figure 9.1: In this example, two codewords of a systematic [4,2] MDS code over the finite field $\mathbb{F}_5$, are piggybacked and appear as columns in the upper-left table. Each row represents the contents of one of the 4 nodes. The piggyback modification results in the code shown on the upper right. The tables in the bottom row correspond to failure of the first and second systematic nodes. The first systematic node can be repaired by reading $\{b_2,\ (b_1+b_2),\ (b_1+2b_2+a_1)\}$, (shown in blue), the second by reading $\{b_1,\ (b_1+b_2),\ (2a_2-2b_2-b_1)\}$.

way that the systematic nodes can be repaired by reading 3 symbols (instead of the 4 symbols customarily required for MDS decoding), resulting in a 25% savings in repair bandwidth and disk reads.

9.1.2 The ϵ-MSR Framework

An $(n,k,\alpha)_{\mathbb{F}}$ ϵ-MSR code is an $[n,k]$ vector MDS code over $\mathbb{F}^{\alpha}$ having the additional property that any failed node $i \in [n]$, can be repaired by downloading $\leq (1+\epsilon)\frac{\alpha}{n-k}$ symbols from each of the remaining $(n-1)$ nodes. Thus the number of helper nodes equals $(n-1)$ in this construction. It is shown by Rawat *et al.* in [194], [195] that it is possible to construct an ϵ-MSR code with $\alpha = O(\log n)$ for all $\epsilon > 0$. The ϵ-MSR code construction technique involves combining a short block-length MSR code with a code having large minimum distance.

Let $\mathcal{C}_I$ be an $(n = k+r,\ k,\ d = n-1)$ MSR code with sub-packetization level α over a finite field $\mathbb{F}$ having p-c matrix

$$H = \begin{bmatrix} A_{0,0} & A_{0,1} & \dots & A_{0,n-1} \\ A_{1,0} & A_{1,1} & \dots & A_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{r-1,0} & A_{r-1,1} & \dots & A_{r-1,n-1} \end{bmatrix},$$

where the sub-matrices $A_{i,j}$ are of size $(\alpha \times \alpha)$.

Let $\mathcal{C}_{II}$ be a second (not necessarily linear) code having block length N, size M and minimum distance $D = \delta N$ over an alphabet $\mathcal{A}$ of size $|\mathcal{A}| \leq n$ that we identify with a subset $\mathcal{A} \subseteq [0, n-1]$. We associate with every codeword $\underline{c} = (c_1\ c_2\ \cdots\ c_N)$ of $\mathcal{C}_{II}$, an $(rN\alpha \times N\alpha)$ matrix:

$$\mathcal{H}_{\underline{c}} = \begin{bmatrix} u_{1,\underline{c}}\ \mathrm{Diag}(A_{0,c_1}, \ldots, A_{0,c_N}) \\ \vdots \\ u_{r,\underline{c}}\ \mathrm{Diag}(A_{r-1,c_1}, \ldots, A_{r-1,c_N}) \end{bmatrix}.$$

Here the $\{u_{i,\underline{c}}\}$, are codeword-dependent, non-zero coefficients, drawn from $\mathbb{F}$. Next, we form an $(rN\alpha \times MN\alpha)$ matrix $\mathcal{H}$ by horizontally stacking the M matrices $\mathcal{H}_{\underline{c}}$ corresponding to M codewords in $\mathcal{C}_{II}$. It can be shown that the code having $\mathcal{H}$ as its p-c matrix, is an $(M, M-r, N\alpha)_{\mathbb{F}}$ ϵ-MSR code, where $\epsilon = (r-1)(1-\delta)$. Ensuring this requires judicious selection of the base MSR code $\mathcal{C}_I$ as well as the non-zero scalars $\{u_{i,\underline{c}}\}$. An additional requirement is that for a given $\epsilon > 0$, the code $\mathcal{C}_{II}$ should be chosen such that the parameter δ satisfies $\delta \geq 1 - \frac{\epsilon}{r-1}$. The ϵ-MSR codes constructed using this approach can be made to have sub-packetization level scaling logarithmically in the block length.

In [194], an ϵ-MSR code construction is provided, in which the Diagonal MSR code constructed in [255], is used as the building block. An example construction is described below.

Let $\mathcal{C}_I$ be chosen to be an $(n = 3, k = 1, d = 2, \alpha = 2^3 = 8)$ Diagonal MSR code. Let $\mathcal{C}_{II}$ be chosen to be a code with $(N = 20, M = 27, D = 13)$ over $\mathbb{F}_3$. Using these two codes, one can construct an $(M = 27, M - r = 25, N\alpha = 160)$ ϵ-MSR code with $\epsilon = 0.35$. Note that the $(n = 27, k = 25, d = 26)$ Diagonal MSR code requires a sub-packetization level of 2^{27}, whereas this ϵ-MSR code has a sub-packetization level of 160 ($\ll 2^{27}$) and repair bandwidth that is no more than 1.35 times that of the Diagonal MSR code.

9.1.3 Li-Liu-Tang Transformation

In [138], a generic transformation that makes use of MSR codes to build vector MDS codes having near-optimal repair bandwidth and small sub-packetization level is presented. Four different vector MDS codes are obtained by applying this transformation to various MSR codes

known in the literature. A fifth MDS code construction is also presented, that does not make use of the generic transformation.

The idea behind the generic transformation can be traced back to [229]. The p-c matrix H' of an $(n', k' = n' - r, d = n' - 1)$ MSR code having sub-packetization level α can be expressed in the following block-matrix form

$$H' = \begin{bmatrix} A'_{0,0} & A'_{0,1} & \cdots & A'_{0,n'-1} \\ A'_{1,0} & A'_{1,1} & \cdots & A'_{1,n'-1} \\ \vdots & \vdots & \ddots & \vdots \\ A'_{r-1,0} & A'_{r-1,1} & \cdots & A'_{r-1,n'-1} \end{bmatrix},$$

where each $A'_{i,j}$ is an $(\alpha \times \alpha)$ matrix. For the generic transformation to work each $A'_{i,j}$ is required to be non-singular. There are MSR code constructions in the literature that satisfy this requirement, for example, the Diagonal MSR code presented in [255].

Under the generic transformation, one passes on to a code having larger block length $n = sn'$, while maintaining the same sub-packetization level α. The p-c matrix H of the code having the block length $n = sn'$ takes on the form:

$$H = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,n-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{r-1,0} & A_{r-1,1} & \cdots & A_{r-1,n-1} \end{bmatrix},$$

with

$$A_{i,j} = x_{i,j} A'_{i,(j \pmod{n'})}, \quad \forall i \in [0, r-1], j \in [0, n-1],$$

where $\{x_{i,j}\}$ are indeterminates. It can be shown using the Combinatorial Nulstellensatz [5] that over a sufficiently large finite field, there exists an assignment of values to the $\{x_{i,j}\}$ under which this code is an $[n, k = n - r]$ vector MDS code. Under this argument, the requirement placed on the field size q is

$$q > \alpha \binom{n-1}{r-1} + 1.$$

For repair of node i, $(s-1)$ nodes with indices $j \neq i$ such that $j = i \pmod{n'}$ send α symbols whereas the remaining $(n-s)$ nodes send $\beta = \frac{\alpha}{r}$ symbols resulting in the repair bandwidth of

$$(n-1)\frac{\alpha}{r} + (s-1)\left(\alpha - \frac{\alpha}{r}\right) \quad = \quad (n-1)\frac{\alpha}{r}\left(1 + \frac{(s-1)(r-1)}{(n-1)}\right).$$

The term $\left(1 + \frac{(s-1)(r-1)}{(n-1)}\right)$ represents the factor by which the new code has larger bandwidth in comparison with an MSR code. Thus this method can substantially reduce the sub-packetization level while keeping the increase in repair bandwidth to a manageable level since the factor $\frac{(s-1)(r-1)}{(n-1)}$ is < 1.

In [138], the authors apply this simple generic transformation to four MSR codes: the Diagonal MSR code [255], the Permuted-Diagonal MSR code [255], an optimal-access MSR code in [148] and the CL-MSR code [137], [205], [256]. This yields four vector MDS codes $\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}$ respectively, that offer significantly reduced sub-packetization level for a modest increase in repair bandwidth. The drawback here is the large field-size requirement given by $q > \alpha\binom{n-1}{r-1} + 1$. For codes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$, the field size requirement is reduced to $q > rn'\lceil\frac{s}{r}\rceil$ with $r|(q-1)$, $q > r\lceil\frac{n'}{r}\rceil(s-1) + n'$ and $q > sr$ respectively, by identifying a specific assignment of the $\{x_{i,j}\}$ as opposed to appealing to the Combinatorial Nullstellensatz.

The fifth vector MDS code $\mathcal{C}_5$ in [138] also has same repair bandwidth as the four codes described above. This code is constructed directly without using the transformation and draws upon the form of the Diagonal MSR code [255]. The structure of $\mathcal{C}_5$ is similar to $\mathcal{C}_1$.

Open Problem 9. Characterize the tradeoff between repair bandwidth, sub-packetization level and field size for the general class of vector MDS codes.

9.2 Fractional Repetition Codes

Fractional repetition codes introduced by El Rouayheb and Ramchandran in [59] may be regarded as a generalization in a certain direction, of the polygonal MBR code presented in Section 4.1 having the RBT property.

A fractional repetition code is associated with a parameter set $\{n, k, d, \rho, B\}$. Let B be the size of file to be stored using the fractional repetition code. The B message symbols are first encoded using a scalar $[N, B]_q$ MDS code $\mathcal{C}$, called the outer code, to obtain a codeword $\underline{c} \in \mathcal{C}$. Each of the N code symbols in $\underline{c}$ is then replicated $\rho \geq 2$ times. The ρN symbols thus obtained, are stored across n nodes in such a way that

- Each node stores d code symbols of $\underline{c}$
- Each code symbol of $\underline{c}$ is stored in exactly ρ nodes.

Clearly, for this to be possible, we need $N\rho = nd$. Also, the sub-packetization level $\alpha = d$. The assignment of code symbols to nodes in fractional repetition codes in accordance with the above requirements, can be accomplished with the aid of combinatorial designs such as Steiner system. In [67], the author identifies necessary and sufficient conditions for the existence of fractional repetition codes. The parameter k is the smallest integer such that the B message symbols can be retrieved from any set of k nodes.

A key difference between an MBR code and a fractional repetition code is that in an MBR code, any collection of d nodes can be selected as helper nodes for the repair of a failed node. In contrast, a fractional repetition code only guarantees the existence of at least one set of d helper nodes that enable RBT of a failed node. This is also referred to in the literature as table-based repair. Given the ρ-wise replication of code symbols from the scalar code, it follows that in a fractional repetition code, RBT of up to $\rho - 1$ simultaneous node failures is possible.

We now present an example fractional repetition code construction [59] with parameters $\{n = 6, k = 3, d = 3, \rho = 2, B = 7\}$. The outer code here is an $[N = 9, B = 7]$ MDS code $\mathcal{C}$ and hence there are 9 code symbols $\{c_1, c_2, \ldots, c_9\}$ in each codeword of $\mathcal{C}$. Each of the $n = 6$ lines in Fig. 9.2 indicates a node. The $\alpha = 3$ points lying on a line denote the code symbols stored in the corresponding node. For example, symbols $\{c_1, c_4, c_7\}$ are stored in the node corresponding the straight line connecting the three points. Each code symbol or point, lies at the intersection of two lines, resulting in $\rho = 2$. It can be easily verified that any collection of $k = 3$ nodes contain at least $B = 7$ distinct

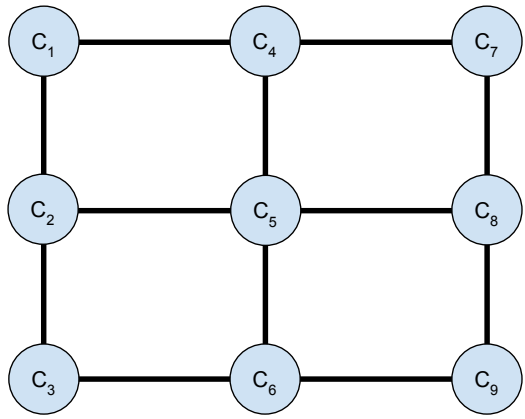

Figure 9.2: An example of a fractional repetition code having parameters $\{n = 6, k = 3, d = 3, \rho = 2, B = 7\}$. The code symbols $(c_i, i = 1, 2, \cdots, 9)$ form a codeword in a scalar $[9, 7]$ MDS code. Each straight line represents a node and the points (encircled code symbols) represent the contents of the node. Thus $\alpha = 3$ as three points lie on each line. We have $n = 6$ as there are 6 lines in all and $\rho = 2$ since each point lies at the intersection of two lines.

code symbols, from which the file can be retrieved. Next, suppose one of the nodes has failed. There are $d = 3$ lines intersecting the line corresponding to the failed node in a point and these $d = 3$ nodes will serves as helper nodes. The failed node can be repaired by just downloading one code symbol each from $d = 3$ helper nodes.

Note that an MBR code with identical parameters, i.e., $(n = 6, k = d = 3, \beta = 1)$, can only store a file of size $(dk - \binom{k}{2})\beta = 6$, whereas this fractional repetition code has file size $B = 7$. Thus the relaxation in code-design requirement arising from permitting restricted choice of helper-node sets has allowed, in this case, a fractional repetition code to store a larger number of message symbols in comparison to the corresponding MBR code. An upper bound on file size of fractional repetition codes is derived in [59] and constructions achieving this bound for some parameters are presented in [221].

In [125], the authors study fractional repetition codes that have sub-packetization α much larger than the replication degree ρ. A randomized version of fractional repetition codes can be found in [170]. Different generalizations of fractional repetition codes have been studied in the literature, including those in [81], [165], [264].

In related work [4], the (n, k, d) parameter range over which table-based repair results in a strictly-improved, storage-repair-bandwidth tradeoff when compared with the corresponding tradeoff that applies to an FR RGC having the same (n, k, d) parameters, is characterized.

9.3 Cooperative Regenerating Codes

Two approaches have been adopted in the RGC literature to handle the case when $t > 1$ nodes fail simultaneously. Under centralized-repair, a single repair center downloads helper data from a set of d helper nodes and uses this data to determine the contents of the t replacement nodes. In the case of an $[n, k]$ vector MDS code with sub-packetization α, the least amount of data download required from d helper nodes for the simultaneous repair of t failed nodes under centralized-repair [31] is given by

$$\frac{\alpha t d}{d - k + t}$$

and codes achieving this can be found described in [255]. The FR storage repair bandwidth tradeoff under centralized-repair of multiple node failures is explored in [97], [191], [265].

An alternate method of repairing multiple failed nodes simultaneously is cooperative-repair, under which there is a separate repair center for each replacement node. The repair centers are permitted to exchange data. The potential benefit of allowing such data exchange was first investigated by Hu *et al.* in [100]. As with an RGC, in a cooperative RGC, each of the n nodes store α symbols and the contents of any k nodes are sufficient to reconstruct the stored data file of size B.

The cooperative-repair of t node failures takes place in two phases. In the first phase, each of the t replacement nodes selects a set of d helper nodes and downloads β_1 symbols from each of them. In the second phase, every replacement node downloads β_2 symbols from each of the other $(t - 1)$ replacement nodes. Hence, the repair bandwidth per node is given by

$$\gamma = d\beta_1 + (t - 1)\beta_2.$$

The minimum storage cooperative regenerating (MSCR) point and minimum bandwidth cooperative regenerating (MBCR) point are determined in [118] and [218], and are given by:

$$\begin{aligned}(\alpha_{\text{MSCR}}, \gamma_{\text{MSCR}}) &= \Big(\frac{B}{k}, \frac{B(d+t-1)}{k(d+t-k)}\Big),\\ \alpha_{\text{MBCR}} &= \gamma_{\text{MBCR}} = \frac{B(2d+t-1)}{k(2d+t-k)}.\end{aligned}$$

Note that when $t = 1$, these reduce to the corresponding points for single node repair. A cooperative RGC operating at the MSCR point is once again an MDS code.

The entire storage versus repair bandwidth per node tradeoff curve under FR is derived in [218]. In the case of exact repair, explicit constructions of cooperative RGCs for all parameters at the MBCR point are presented in [244] and at the MSCR point in [257]. In [146], the cooperative-repair model was extended to a partial collaboration model under which, during the second phase of node repair, a replacement node exchanges β_2 symbols with $(t-s)$ other replacement nodes, where $1 \leq s \leq t$. The security of cooperative RGCs is investigated in [106], [126].

9.4 Secure Regenerating Codes

Three secrecy models in the context of an RGC were introduced by Pawar *et al.* in [169]:

- A passive eavesdropper model, where the eavesdropper can read the content and repair data of any $\ell < k$ nodes, but cannot modify the content of these nodes,
- An active omniscient adversary model, where the adversary knows the data stored in all the nodes, and can modify the content of b nodes where $2b < k$,
- An active limited-knowledge adversary model, where the adversary can read content and repair data of $\ell < k$ nodes and can modify the content of $b \leq \ell$ nodes among them.

Both passive eavesdropper and active adversary settings are associated to notions of capacity as described below. In the passive eavesdropper setting, the secrecy capacity B_s is defined to be the maximum amount of information that can be stored without any information being revealed to the eavesdropper. In the active adversary setting, the resiliency capacity B_r is defined to be the maximum amount of information that can be stored in such a manner that it can be reliably made available to a legitimate data collector, despite tampering by the adversary, of the data contained in b nodes.

The following upper bound on secrecy capacity of the passive eavesdropper model was derived in [169]:

$$B_s \leq \sum_{i=\ell+1}^{k} \min\{(d-i+1)\beta, \alpha\}. \tag{9.1}$$

For the case when α is unconstrained, i.e., $\alpha > (d-\ell)\beta$, the resultant bandwidth-limited secrecy capacity $B_{s,\text{BL}}$ is determined in [169] for the case $d = (n-1)$, where a bound and matching construction are presented. It was also shown that the resiliency capacity satisfies

$$B_r \leq \sum_{i=i_0}^{k} \min\{(d-i+1)\beta, \alpha\},$$

where the lower limit i_0 is equal to $2b+1$ in the omniscient case and to $b+1$ in the limited knowledge case.

In an alternate setting, Rashmi et al. in [181] assume a noisy channel for the transmission of data during repair and reconstruction, and introduce the notion of an (s,t)-resilient RGC that can correct up to t errors and s erasures during both repair and data collection. The model is aligned with the active adversary model where the adversary can tamper with the contents of b nodes. The capacity or file size B of an (s,t)-resilient RGC is shown to satisfy

$$B \leq \sum_{i=1}^{\kappa} \min\{(\Delta-i+1)\beta, \alpha\}$$

where $\Delta := (d-2t-s)$ and $\kappa := (k-2t-s)$. Constructions of MSR and MBR codes that are (s,t)-resilient are also provided in [181]. In

[255], the authors extend this model to the repair of multiple nodes and provide MSR constructions that are resilient to t errors during the repair process.

In [182], the authors extend the passive eavesdropper model to the setting where out of the ℓ nodes accessed, the eavesdropper can read the contents of ℓ_1 nodes and can observe the information passed on for the repair of $\ell_2 = \ell - \ell_1$ nodes. The upper bound in (9.1) also holds for this case. In the case of an MBR code, since the amount of data stored equals the amount of data received for node repair, the breakup of ℓ between ℓ_1, ℓ_2 is immaterial. This is not true in the case of an MSR code where $d\beta > \alpha$. In [182], the authors provide secure MBR code constructions matching the upper bound in (9.1) for all possible parameters. A secure, low-rate MSR code construction that achieves the upper bound (9.1) for $\ell_2 = 0$ is also presented in [182]. This secure MSR construction establishes a lower bound to the secure file size of an MSR code: $B_s \geq (k-\ell)(\alpha - \ell_2\beta)$ for $\ell_2 > 0$.

The upper bound on secure MSR file size $B_s \leq (k-\ell)\alpha$ given by (9.1) is improved in [75], [105], [187], [235]. In [188], the authors establish that the secrecy capacity of an MSR code with $d = n-1$ is given by

$$B_s = (k-\ell)\Big(1 - \frac{1}{n-k}\Big)^{\ell_2}\alpha$$

by providing a secure MSR construction matching the upper bound on secure file size given in [75]. The authors of [177] extended this construction to determine the secrecy capacity of MSR codes with $d < n-1$, for the $\ell_1 = 0$ case. In [113], secure MSR codes having smaller field size are constructed for all parameters. In [120], [121], [216], [253] the ER tradeoff for secure RGCs is studied.

9.5 Rack-Aware Regenerating Codes

The storage nodes in a data center are typically organized into racks that contain an equal number of nodes. The communication between nodes within a rack is less expensive than cross-rack communication. With this in mind, rack-aware regenerating codes (RRGCs) [99] focus on minimizing the number of symbols that are exchanged across racks during node repair.

In an RRGC, the n nodes are divided into r racks, such that each rack contains $\frac{n}{r}$ nodes, where n is a multiple of r. Each node continues to store α symbols. The data file of size B stored using an RRGC must be retrievable from any k nodes, as in the case of an RGC. For the repair of a failed node in an RRGC, the replacement node is given access to the entire content of all the nodes belonging to the same rack, as well as to an additional set of $d\beta$ symbols, obtained by downloading β symbols from each of d other, helper racks. The β symbols downloaded from any such helper rack can be a function of the entire content of that helper rack. Communication between nodes lying within the same rack does not count towards the repair bandwidth, so that the aim in node repair in the RRGC setting, is to minimize the quantity $d\beta$, referred to as the cross-rack repair bandwidth. The FR storage-bandwidth tradeoff for RRGCs was characterized in [95]. The minimum storage rack-aware (MSRR) and minimum bandwidth rack-aware regenerating (MBRR) points are given by,

$$\begin{aligned}(\alpha_{\text{MSRR}}, \beta_{\text{MSRR}}) &= \Big(\frac{B}{k}, \frac{B}{k(d-m+1)}\Big),\\ \alpha_{\text{MBRR}} = d\beta_{\text{MBRR}} &= \frac{dB}{kd - \frac{m(m-1)}{2}},\end{aligned}$$

where $m = \lfloor \frac{kr}{n} \rfloor$. Explicit ER constructions of MSRR codes for all parameters can be found in [41] and MBRR codes in [263]. There are other rack-aware models that have been studied in the literature, including those in [171], [224].

10

Locally Recoverable Codes

10.1 Background

While an RGC aims at minimizing the repair bandwidth, the principal aim in the case of a locally recoverable code (LRC) is on keeping to a small number the number of helper nodes contacted for repairing a failed node, termed the repair degree.

Several papers have appeared in the literature introducing the concept of locality in an error-correcting code from slightly different perspectives. These include the paper on subline coding by Han and Lastras-Montaño [87], the paper on pyramid codes by Huang *et al.* [102], the paper by Oggier and Datta [164] on self-repairing homomorphic codes and the paper by Gopalan *et al.* [72] presenting a comprehensive treatment of codes with locality. With the exception of [87], the focus in all the above papers is on linear codes. Apart from [87], the list of early papers containing a treatment of nonlinear LRCs include the papers by Papailopoulos and Dimakis [168], Forbes and Yekhanin [68] and Tamo and Barg [228].

We begin this section with a brief discussion on nonlinear LRCs before going on to treat the case of linear codes in greater detail.

10.2 Nonlinear LRCs

Definition 6 (Nonlinear LRC with All-Symbol Locality)**.** A code $\mathcal{C}$ of block length n and size M over an alphabet $\mathcal{A}_q$ of size q is said to be an (n, M) code with (all-symbol) locality (r, δ) if associated to every code symbol c_i, $i = 1, 2, \cdots, n$, of a codeword $\underline{c} = (c_1, \cdots, c_n) \in \mathcal{C}$, there is a set $S_i \subseteq [n]$ of size $n_i := |S_i| \leq (r + \delta - 1)$ such that the restriction $\mathcal{C}_i := \mathcal{C}|_{S_i}$ of $\mathcal{C}$ to S_i is a code of block length n_i and minimum distance $\geq \delta$. The code $\mathcal{C}_i := \mathcal{C}|_{S_i}$, is called the local code associated with code symbol c_i.

It is typically assumed that the size M is of the form $M = q^k$, so that the LRC can be viewed as encoding a set of k message symbols over the alphabet $\mathcal{A}_q$.

Theorem 6. Let $\mathcal{C}$ be an (n, M) LRC having (r, δ) locality and of size $M = q^k$ over an alphabet $\mathcal{A}_q$ of size q. Then the rate and minimum distance of the LRC are respectively upper bounded by

$$\begin{aligned} \frac{k}{n} &\leq \frac{r}{r + \delta - 1}, \\ d_{\min} &\leq (n - k + 1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\delta - 1). \end{aligned}$$

A proof of the above theorem is given in Section 10.8. Alternative proofs for the case $\delta = 2$ can be found in [68], [168], [228].

10.3 Linear LRCs

The early study of LRCs in the linear case was mostly centered on systematic linear codes, where only the message symbols were guaranteed to be repaired with low degree. These codes were accordingly termed as codes with information-symbol locality. The study was subsequently expanded to include all-symbol locality codes, i.e., LRCs where it was possible to repair all the code symbols with low repair degree. In this section, we begin with information-symbol locality before moving on to discuss all-symbol locality.

The original treatment in [72] was for the case when the local codes have minimum distance $\delta = 2$, corresponding to single-parity-check

codes. The practical usage of LRCs in the Azure code, described in Section 10.6, also involves local codes having minimum distance $\delta = 2$. However, we state and prove the bounds on $d_{\min}$ and code rate $\frac{k}{n}$ here for the more general case $\delta \geq 2$ appearing in [172], as the proof technique remains the same.

Definition 7 (Linear LRC with Information-Symbol Locality)**.** An $[n, k]$ systematic, linear code $\mathcal{C}$ is to be an (r, δ) LRC with information-symbol locality if associated to every message symbol $u_i, 1 \leq i \leq k$, there is a set of ℓ other code symbols $(c_{i_1}, c_{i_2}, \cdots, c_{i_\ell})$ with $\ell \leq r + \delta - 2$ such that the set of $\ell + 1$ code symbols $(u_i, c_{i_1}, c_{i_2}, \cdots, c_{i_\ell})$ forms a code $\mathcal{C}_i$ of block length $= \ell + 1$ and minimum distance $\geq \delta$. We will refer to $\mathcal{C}_i$ as a local code associated to message symbol u_i.

The reason for regarding an (r, δ) LRC as having locality r can be seen from the following. If $(\delta - 1)$ symbols from a local codeword are erased, one is left with $(\ell + 2 - \delta)$ unerased symbols. On the other hand, in any $[n, k, d_{\min}]$ linear code, all message symbols can be recovered from any collection of $(n - d_{\min} + 1)$ code symbols. In the case of a local code of block length $(\ell + 1)$ and minimum distance $\geq \delta$, this works out to $\leq (\ell + 2 - \delta)$. Further, since $\ell + 1 \leq (r + \delta - 1)$, we have $(\ell + 2 - \delta) \leq r$. Thus even in the presence of $(\delta - 1)$ erasures, each local code is always guaranteed to be able to recover the local codeword from any r or less of the remaining code symbols.

Remark 8. We make the following additional observations.

1. The minimum distance $d_{\min}$ of the LRC is $\geq \delta$. This follows from noting that the minimum Hamming weight of codewords in a local code $\mathcal{C}_i$ is $\geq \delta$, hence the same is true of a codeword in the LRC.

2. It is possible for a given local code to be associated to more than one message symbol and conversely, a given message symbol can be associated to more than one local code.

3. By the Singleton bound, the minimum distance of a local code cannot be larger than 1 plus its redundancy. It follows that (i) the redundancy of a local code $\mathcal{C}_i$ of minimum distance $\geq \delta$ must be

$\geq (\delta - 1)$ and consequently that (ii) dimension can be no larger than $\left(r + \delta - 1 - (\delta - 1)\right) \ = \ r$.

4. If the local code has minimum distance δ, it can have redundancy equal to $(\delta - 1)$ iff the code is MDS. If the local code has minimum distance δ and is an MDS code, then the dimension can equal r iff the length $(\ell + 1)$ of the local code is equal to $(r + \delta - 1)$.

10.4 Bounds on $d_{\min}$ and Rate for Linear LRCs

We now present an upper bound on the minimum distance of a linear (r, δ) LRC. This bound was derived for the case $\delta = 2$ of primary interest by Gopalan *et al.* in [72]. The extension to the case of general δ appears in [172]. We begin with a useful Lemma.

Lemma 9. Let $\mathcal{C}$ be an $[n, k, d_{\min}]$ linear code over a finite field $\mathbb{F}_q$. Let

$$G \ = \ \left[\underline{g}_1 \ \underline{g}_2 \ \cdots \ \underline{g}_n\right]$$

be a generator matrix for $\mathcal{C}$. Let s be the largest possible integer such that there exists a subset $S \subset [n]$ of size $|S| = s$ such that the $(k \times s)$ sub-matrix of G associated to the columns whose indices lie in S has rank $= k - 1$. Then $d_{\min} = n - s$.

Proof: Given a subset $S = \{i_1, i_2, \ldots, i_\ell\} \subseteq [n]$, we will mean by $G|_S$, the sub-matrix of G given by

$$G|_S \ = \ \left[\underline{g}_{i_1} \ \underline{g}_{i_2} \ \cdots \ \underline{g}_{i_\ell}\right],$$

and refer to $G|_S$ as the restriction of G to S.

Let $S \subset [n]$ be of largest possible size s such that $\text{rank}(G|_S) = k - 1$. Then there exists $\underline{u} \in \mathbb{F}_q^k$, $\underline{u} \neq 0$, such that $\underline{u}^t G|_S = \underline{0}^t$. Let $\underline{c}^t = \underline{u}^t G$. Clearly, $\underline{c}^t \neq \underline{0}^t$ and $w_H(\underline{c}^t) \leq n - |S|$. It follows that $d_{\min} \leq n - |S| \ = n - s$.

Next, let $\underline{c} \in \mathcal{C}$ have minimum Hamming weight, i.e., $w_H(\underline{c}) = d_{\min}$. Let $T \subseteq [n]$ be the support of $\underline{c}$ and set $S = [n] \setminus T$. The S has size $|S| = n - d_{\min}$. Let $\underline{u} \in \mathbb{F}_q^k$ be the message vector associated to $\underline{c}$. Then clearly, $\underline{u}^t \neq \underline{0}^t$ and $\underline{u}^t G|_S = \underline{0}^t$. It follows that

$$s \geq n - d_{\min} \implies d_{\min} \geq n - s.$$

Hence $d_{\min} = n - s$. □

Theorem 7. The minimum distance $d_{\min}$ of an $[n, k, d_{\min}]$ (r, δ) LRC with information-symbol locality, is upper bounded by:

$$d_{\min} \leq (n - k + 1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\delta - 1). \tag{10.1}$$

Proof: Let $\{u_1, u_2, \cdots, u_k\}$ denote the k message symbols. Let G be a $(k \times n)$ generator matrix for the LRC. Let $\{S_i \subseteq [n] \mid 1 \leq i \leq k\}$, be subsets of indices such that for each i, $(c_j,\ j \in S_i)$ is a local code of length $\leq r + \delta - 1$ and minimum distance $\geq \delta$ that contains the message symbol u_i. The subsets are not necessarily distinct. The first step in our proof is achieving the following goal.

Goal: Use the subsets S_i to construct a set $T \subseteq [n]$ of large size such that $\text{rank}(G|_T) = k - 1$. By Lemma 9, this will establish that $d_{\min} \leq n - |T|$. Thus in the proof below, we will regard each set S_i as a set of column indices associated to the generator matrix G of the LRC.

We will construct the set T recursively and begin with $T_0 = \phi$. Assuming that we have not stopped at the end of iteration a, we will have obtained at the end of iteration a, a set T_a of the form $T_a = \bigcup_{j=1}^{a} S_{i_j}$ with $\text{rank}(G|_{T_a}) < (k-1)$. We will abuse notation and regard the empty set as also having such a representation. We begin iteration $(a+1)$ by picking an index i_{a+1} such that

$$\text{rank}(G|_P) > \text{rank}(G|_{T_a}) \quad \text{where} \quad P = T_a \cup S_{i_{a+1}}. \tag{10.2}$$

Having selected such an index i_{a+1} and having set $P = T_a \cup S_{i_{a+1}}$, we proceed as follows:

1. If $\text{rank}(G|_P) < (k-1)$, we set $T_{a+1} = T_a \cup S_{i_{a+1}}$ and continue the recursion,

2. If $\text{rank}(G|_P) = (k-1)$, we stop, set $T = T_{a+1} = P$ and the flag to $J = 0$.

3. If $\text{rank}(G|_P) = k$, we delete some elements from $S_{i_{a+1}}$ to obtain a set $\hat{S}_{i_{a+1}}$ with $\text{rank}(G|_{\hat{P}}) = (k-1)$ where $\hat{P} = T_a \bigcup \hat{S}_{i_{a+1}}$. Clearly, this can always be done. We then set $T = \hat{P}$ and stop. We set the flag to $J = 1$.

Case (i) Suppose we exited the recursion at the end of the $(a+1)$th iteration with flag $J = 0$. In this case, $T = T_{a+1} = \bigcup_{j=1}^{a+1} S_{i_j}$. The inclusion of each set S_{i_j} can increase the rank by at most r since each local code has dimension at most r. Therefore,

$$(a+1) \quad \geq \quad \left\lceil \frac{k-1}{r} \right\rceil.$$

Next, we claim that each set $S_{i_j}, j = 1, 2, \cdots, a+1$, brings in additional column indices associated to at least $(\delta - 1)$ redundant columns, i.e., indices associated to columns that do not contribute to an increase in rank. This can be explained as follows. Let

$$\Delta_\ell \quad = \quad \text{rank}(G|_{T_\ell}) - \text{rank}(G|T_{\ell-1}), \ \ell = 1, 2, \cdots, a+1.$$

Choose a subset $U_{i_\ell} \subseteq S_{i_\ell}$ of size $|U_{i_\ell}| = \Delta_\ell - 1$ such that if

$$V_\ell \quad = \quad U_{i_\ell} \cup T_{\ell-1},$$

then

$$\text{rank}(G|_{V_\ell}) \quad = \quad \text{rank}(G|_{T_\ell}) - 1.$$

This is clearly possible. (If $\Delta_\ell = 1$, U_{i_ℓ} can be chosen to be the empty set ϕ). Clearly, we can also write $T_\ell = V_\ell \cup S_{i_\ell}$. Since

$$\text{rank}(G|_{T_\ell}) \quad > \quad \text{rank}(G|_{V_\ell}),$$

we claim that

$$| \, S_{i_\ell} \setminus V_\ell \, | \quad \geq \quad \delta,$$

i.e., that while we have increased the rank by 1, we have increased the number of column indices by a quantity $\geq \delta$. In this way, there are always $(\delta - 1)$ column indices associated to redundant columns that are added at every step.

The justification for the claim is as follows: in any $[n, k, d_{\min}]$ code $\mathcal{A}$, any $(k \times m)$ submatrix of a $(k \times n)$ generator matrix $G_{\mathcal{A}}$ for $\mathcal{A}$, must have rank k if $m \geq n - d_{\min} + 1$. Thus if we partition the column indices of $G_{\mathcal{A}}$ according to $[n] = B_1 \cup B_2$, $B_1 \cap B_2 = \phi$, then

$$\text{rank}(G_{\mathcal{A}}) \quad > \quad \text{rank}(G_{\mathcal{A}}|_{B_1})$$

is possible iff $|B_2| \geq d_{\min}$.
It follows that

$$\begin{aligned} |T| &\geq (k-1) + (a+1)(\delta-1) \\ &\geq (k-1) + \left\lceil \frac{k-1}{r} \right\rceil (\delta-1). \end{aligned}$$

We thus have

$$\begin{aligned} d_{\min} &\leq n - |T| \leq n - \{(k-1) + \left\lceil \frac{k-1}{r} \right\rceil (\delta-1)\} \\ \therefore d_{\min} &\leq (n-k+1) - \left\lceil \frac{k-1}{r} \right\rceil (\delta-1). \end{aligned} \tag{10.3}$$

Case (ii) Suppose we exited the recursion at the end of the $(a+1)$th iteration and flag $J = 1$. We then have,

$$(a+1) \geq \left\lceil \frac{k}{r} \right\rceil \text{ and } \operatorname{rank}(G|_{T_a}) < k-1.$$

We also have $T = T_a \bigcup \hat{S}_{i_{a+1}}$, with $\operatorname{rank}(G|_T) = (k-1)$. We can now apply our earlier arguments about increasing the size of the column index set by at least $(\delta-1)$ at each of the first a steps. Since we have replaced $S_{i_{a+1}}$ by $\hat{S}_{i_{a+1}}$, we cannot assert that this last step has introduced any column indices associated to redundant columns at all. Thus we can only assert that

$$\begin{aligned} |T| &\geq (k-1) + a(\delta-1) \\ &\geq (k-1) + \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\delta-1). \end{aligned}$$

This gives us

$$\therefore d_{\min} \leq (n-k+1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\delta-1). \tag{10.4}$$

Claim: $\left\lceil \frac{k-1}{r} \right\rceil \geq \left\lceil \frac{k}{r} \right\rceil - 1$. This can be seen by verifying that

$$\left\lceil \frac{k}{r} \right\rceil - 1 = \left\lfloor \frac{k-1}{r} \right\rfloor, \ \forall k = ar + b, \ 0 \leq b \leq (r-1).$$

Thus the RHS of (10.4) is larger than the RHS of (10.3). Thus (10.4) is the desired upper bound on $d_{\min}$ since we can always be sure that $d_{\min}$ satisfies the upper bound given by (10.4). □

Remark 9. Thus in comparison with an MDS code having the same block length and dimension, we see that the penalty to be paid for requiring locality is

$$\left(\left\lceil \frac{k}{r} \right\rceil - 1\right)(\delta - 1).$$

We note that equation (10.1) reduces in the case when $\delta = 2$ to

$$d_{\min} \leq (n - k + 1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1\right).$$

As shown below, the minimum distance upper bound in Theorem 7 can be turned around to yield an upper bound on the code rate of an LRC.

Corollary 1. The rate $\frac{k}{n}$ of an $[n, k, d_{\min}]$ code having (r, δ) information-symbol locality is upper bounded by

$$\frac{k}{n} \leq \frac{r}{(r + \delta - 1)}.$$

Proof: Clearly, as noted in Remark 8, we must have $d_{\min} \geq \delta$. This along with the upper bound on $d_{\min}$ in Theorem 7 gives us

$$\delta \leq (n - k + 1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1\right)(\delta - 1).$$

It follows that

$$n \geq k + \left\lceil \frac{k}{r} \right\rceil (\delta - 1)$$

leading to the rate bound

$$\frac{k}{n} \leq \frac{r}{(r + \delta - 1)}.$$

□

Table 10.1 provides a listing of the constructions of LRCs presented in this section. These constructions appear in the subsections that follow.

Table 10.1: LRC constructions described in this monograph. All of the constructions appearing in the table are explicit.

Type of LRC	Code	Section
Information-Symbol Locality	Pyramid LRC [102]	10.5
Information-Symbol Locality	Azure LRC [103]	10.6
All-Symbol Locality	Tamo-Barg LRC [228]	10.7

10.5 Pyramid LRC

The pyramid-code construction technique by Huang *et al.* [102], [104] yields LRCs with information-symbol locality that are optimal with respect to the minimum distance bound in (10.1).

Let us assume that it is desired to construct an $[n, k]$ code $\mathcal{C}$ with (r, δ) information symbol locality and minimum distance d_{LRC} attaining the bound in (10.1). Set $s = \left\lceil \frac{k}{r} \right\rceil$ and let n_0 be such that $n_0 = n - (s-1)(\delta-1)$. Note that

$$n_0 \;=\; n-(s-1)(\delta-1) \;=\; d_{\text{LRC}} + k - 1 \;\geq\; k + (\delta - 1),$$

since as noted earlier, $d_{\text{LRC}} \geq \delta$. The starting point of pyramid code construction is the systematic generator matrix $G_0 = [I_k \; P]$ of an $[n_0, k]$ MDS code $\mathcal{C}_0$. Since $n_0 \geq k + \delta - 1$, it follows that the matrix P has at least $(\delta - 1)$ columns. The pyramid code construction proceeds to replace the first $(\delta - 1)$ columns of P by $s(\delta - 1)$ columns that are derived by splitting each column of P into s columns. The resultant matrix is then the generator matrix G of the desired $[n, k]$ code $\mathcal{C}$.

We explain the manner of column splitting through an illustrative example with $(n = 12, k = 5, r = 2, \delta = 3)$. Here $s = \left\lceil \frac{k}{r} \right\rceil = 3$ and $n_0 = n - (s-1)(\delta-1) = 8$. We begin with the systematic generator matrix:

$$G_0 = \left[I_5 \ \underline{p}_1 \ \underline{p}_2 \ \underline{p}_3\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & p_{11} & p_{12} & p_{13} \\ 0 & 1 & 0 & 0 & 0 & p_{21} & p_{22} & p_{23} \\ 0 & 0 & 1 & 0 & 0 & p_{31} & p_{32} & p_{33} \\ 0 & 0 & 0 & 1 & 0 & p_{41} & p_{42} & p_{43} \\ 0 & 0 & 0 & 0 & 1 & p_{51} & p_{52} & p_{53} \end{bmatrix}$$

of an $[n_0 = 8, k = 5]$ MDS code $\mathcal{C}_0$. We will now proceed to split the first $(\delta - 1) = 2$ columns of P into $s = 3$ columns each. Equivalently, we will replace each of the first $(\delta - 1) = 2$ columns $\underline{p}_1, \underline{p}_2$ of P by a $(k \times s)$ sub-matrix as shown below:

$$\begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{41} \\ p_{51} \end{bmatrix} \Longrightarrow \begin{bmatrix} p_{11} & 0 & 0 \\ p_{21} & 0 & 0 \\ 0 & p_{31} & 0 \\ 0 & p_{41} & 0 \\ 0 & 0 & p_{51} \end{bmatrix}, \quad \begin{bmatrix} p_{12} \\ p_{22} \\ p_{32} \\ p_{42} \\ p_{52} \end{bmatrix} \Longrightarrow \begin{bmatrix} p_{12} & 0 & 0 \\ p_{22} & 0 & 0 \\ 0 & p_{32} & 0 \\ 0 & p_{42} & 0 \\ 0 & 0 & p_{52} \end{bmatrix}.$$

In general, if $k = ar$, we split each of the first $(\delta - 1)$ columns of P into a columns, each containing r nonzero elements. If $k = ar + b$, with $0 < b \leq (r-1)$, we split each column into s columns with $(s-1)$ columns having r elements each and the last column containing b elements. In the present case $r = 2$ and $b = 1$. This yields the generator matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & p_{11} & 0 & 0 & p_{12} & 0 & 0 & p_{13} \\ 0 & 1 & 0 & 0 & 0 & p_{21} & 0 & 0 & p_{22} & 0 & 0 & p_{23} \\ 0 & 0 & 1 & 0 & 0 & 0 & p_{31} & 0 & 0 & p_{32} & 0 & p_{33} \\ 0 & 0 & 0 & 1 & 0 & 0 & p_{41} & 0 & 0 & p_{42} & 0 & p_{43} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & p_{51} & 0 & 0 & p_{52} & p_{53} \end{bmatrix}.$$

The next step is to rearrange the columns of the matrix G as shown below:

$$\begin{bmatrix} 1 & 0 & p_{11} & p_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{13} \\ 0 & 1 & p_{21} & p_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{23} \\ 0 & 0 & 0 & 0 & 1 & 0 & p_{31} & p_{32} & 0 & 0 & 0 & p_{33} \\ 0 & 0 & 0 & 0 & 0 & 1 & p_{41} & p_{42} & 0 & 0 & 0 & p_{43} \\ 0 & 0 & \underbrace{0}_{\text{local parity}} & 0 & 0 & 0 & 0 & 0 & 1 & p_{51} & p_{52} & \underbrace{p_{53}}_{\text{global parity}} \end{bmatrix}. \quad (10.5)$$

The rearrangement makes it easy to recognize that the resultant $[12, 5]$ code $\mathcal{C}$ generated by G has $(r = 2, \delta = 3)$ locality. The minimum distance of the $[8, 5]$ MDS code is 4. Hence the minimum Hamming weight of a codeword of $\mathcal{C}_0$ is also 4. A little thought will show that the expansion in the columns in the splitting manner just carried out will not decrease the minimum Hamming weight. Hence $\mathcal{C}$ has $d_{\min} \geq 4$. But by the $d_{\min}$ bound, we have

$$d_{\min} \quad \leq \quad (n - k + 1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\delta - 1) \quad = \quad 4.$$

Hence $\mathcal{C}$ has $d_{\min} = 4$ which is the best $d_{\min}$ possible.

In the general case, the MDS code $\mathcal{C}_0$ is an $[n_0, k, n_0 - k + 1]$ code, and we have chosen n_0 such that the minimum distance of the MDS code $(n_0 - k + 1) = d_{\text{LRC}}$. The column-splitting process results in a code $\mathcal{C}$ with parameters $[n, k, d_{\min} \geq d_{\text{LRC}}]$, by the same argument used in the example. But by the $d_{\min}$ bound in (10.1), the minimum distance can be no larger than d_{LRC}. It follows that $\mathcal{C}$ is an LRC with information-symbol locality that attains the $d_{\min}$ bound in (10.1).

We will refer to code symbols in the pyramid code corresponding to columns that have been split as local parity symbols. The remaining parity symbols will be termed as global parity symbols. Thus in the example pyramid code there are a total of 6 local parities, with 2 local parities associated to each of the three local codes and a single global parity-check as noted in equation (10.5).

10.6 Azure LRC

The Windows Azure storage system employs an $[n = 18, k = 14]$ LRC with $(r = 7, \delta = 2)$ information-symbol locality [101], [103]. This code has a structure similar to the pyramid code. Fig. 10.1 illustrates the structure of this LRC. The dotted boxes identify the code symbols of each of the two local codes, each having a single parity symbol. In addition, there are two global parities. This code has minimum distance $d_{\min} = 4$ and can tolerate erasure of any 3 code symbols. We now compare the Azure code against an $[9, 6]$ RS code that also has $d_{\min} = 4$, and thus is also tolerant to 3 erasures. The repair degree of

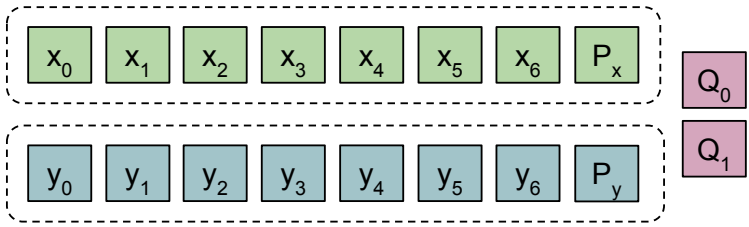

Figure 10.1: The $[18, 14, 4]$ LRC employed in Windows Azure storage. Here P_x and P_y are the local parities, while Q_0, Q_1 represent the two global parities.

the two codes are comparable, at 6 for the $[9, 6]$ RS code, and 7 for the Azure LRC. The primary difference between the two codes lies in the storage overhead. While the $[9, 6]$ RS code has a storage overhead of 1.5, this falls to 1.29 in the case of the Azure LRC. This difference in storage overhead has reportedly resulted in a large cost savings to Microsoft [161].

10.7 Tamo-Barg LRC

Analogous to the definition of a linear LRC with information-symbol locality, we have the definition below of a linear code having all-symbol locality.

Definition 8 (Linear LRC with All-Symbol Locality)**.** An $[n, k]$ linear code $\mathcal{C}$ is to be an (r, δ) LRC with all-symbol locality if associated to every code symbol $c_j, 1 \leq j \leq n$, there is a set of ℓ other code symbols $(c_{i_1}, c_{i_2}, \cdots, c_{i_\ell})$ with $\ell \leq r + \delta - 2$ such that the set of $\ell + 1$ code symbols $(c_j, c_{i_1}, c_{i_2}, \cdots, c_{i_\ell})$ forms a code $\mathcal{C}_j$ of block length $= \ell + 1$ and $d_{\min} \geq \delta$. We will refer to $\mathcal{C}_j$ as a local code associated to code symbol c_j.

Let $\mathcal{C}$ be an $[n, k, d_{\min}]$ linear code with (r, δ) all-symbol locality. Clearly, even under a permutation of code symbols, the code will remain an $[n, k, d_{\min}]$ code with (r, δ) all-symbol locality. One can always construct a systematic generator matrix $G_{\text{sys}} = [I_k \ P]$ either for the code $\mathcal{C}$ or else, a code obtained by permuting code symbols in $\mathcal{C}$. Clearly, when encoded by G_{sys}, one obtains a code that has information-symbol

locality. It follows that the minimum distance bound for an information-symbol locality code given in (10.1) also holds for an all-symbol code. This raises the question as to whether there exist codes with all-symbol locality that achieve the minimum distance bound in (10.1).

The answer is in the affirmative for $(r+\delta-1) \mid n$ and the construction described below for an optimal, all-symbol locality LRC due to Tamo and Barg appears in [228]. Let us assume that it is desired to construct an $[n, k]$ code with all-symbol locality over a finite field $\mathbb{F}_q$ where each code symbol is a code symbol of a local code of block length $\ell \leq r+\delta-1$ and minimum distance $\geq \delta$. We describe the construction as it applies to the case $(r+\delta-1) \mid n$. The construction can however, be generalized [123], [176] to obtain LRCs with all-symbol locality having minimum distance at most one less than the bound in (10.1) for the case $\delta = 2$ that avoids this restriction, and this is described in the notes subsection.

Let $S = \{\theta_i\}_{i=1}^{n}$ be a set of n distinct elements lying in $\mathbb{F}_q$. Set $m = \frac{n}{(r+\delta-1)}$ and let $\{S_j\}_{j=0}^{m-1}$ be a collection of m, pairwise disjoint subsets of size $(r+\delta-1)$ that partition S i.e.,

$$S = \bigcup_{j=0}^{m-1} S_j.$$

Let us assume that it is possible to identify a polynomial $g(x)$ that we will refer to as a "good" polynomial satisfying:

(i) $\deg(g) = (r+\delta-1)$ and

(ii) $g(x) = b_j$ for all $x \in S_j$, i.e., $g(\cdot)$ is constant on each subset S_j.

For simplicity, we describe the construction for the case $r \mid k$. We will then show how this can be extended to the general case. Let $\mathcal{F} \subseteq \mathbb{F}_q[x]$ be the set of polynomials $\{f(\cdot)\}$ over $\mathbb{F}_q$ that can be expressed in the form:

$$f(x) = \sum_{i=0}^{\frac{k}{r}-1} \sum_{j=0}^{r-1} a_{ij}[g(x)]^i x^j.$$

Let $\mathcal{C}$ be the linear code over $\mathbb{F}_q$ given by;

$$\mathcal{C} = \{(f(\theta_i), i = 1, 2, \ldots, n) \mid f \in \mathcal{F}\}.$$

Note that $f(x)\big|_{x\in S_j}$ i.e., the polynomial $f(\cdot)$ restricted to the subset S_j, reduces to a polynomial of degree $\leq (r-1)$. We regard the m subcodes

$$\{(f(x), x \in S_j) \ \mid \ f(\cdot) \in \mathcal{F}\}, \ 0 \leq j \leq (m-1),$$

as the local codes. We see that each local code is an $[r+\delta-1, r, \delta]$ MDS code.

Next, we claim that $\dim(\mathcal{C}) = k$. To see this, we first note that the polynomials $[g(x)]^i x^j$ for different pairs (i,j), $0 \leq i \leq \frac{k}{r}-1$, $0 \leq j \leq r-1$ have different degrees and are hence linearly independent. Hence the subspace of $\mathbb{F}_q[x]$ spanned by the polynomials in $\mathcal{F}$ has dimension $= k$. The maximum degree of a polynomial in $\mathcal{F}$ equals

$$\begin{aligned}(r+\delta-1)\left(\frac{k}{r}-1\right)+(r-1)\\ = \ k-1+\left(\frac{k}{r}-1\right)(\delta-1).\end{aligned}$$

Note that $n = (r+\delta-1)m$. We showed in Corollary 1 that the maximum rate of an LRC is upper bounded by

$$\frac{k}{n} \ \leq \ \frac{r}{r+\delta-1}.$$

It follows from this that $k \leq rm$.

$$\begin{aligned}\therefore k-1+\left(\frac{k}{r}-1\right)(\delta-1) \ &\leq \ rm-1+(m-1)(\delta-1)\\ &= \ (r+\delta-1)m-\delta\\ &< \ (r+\delta-1)m-1 = n-1,\end{aligned}$$

since $\delta \geq 2$. It follows that the mapping $f(x) \in \mathcal{F} \iff (f(\theta), \theta \in S)$ is an injection. This establishes that the code $\mathcal{C}$ has dimension k.

Since the maximum degree of polynomial in $\mathcal{F}$ equals $k-1+\left(\frac{k}{r}-1\right)(\delta-1)$, it follows that

$$d_{\min}(\mathcal{C}) \ \geq \ (n-k+1)-\left(\frac{k}{r}-1\right)(\delta-1).$$

It follows from (10.1) that $\mathcal{C}$ is an optimal code with (r,δ) all-symbol locality.

For the case when $r \nmid k$, let $k = ur + v$, $0 < v \leq (r-1)$. Then simply replace the set $\mathcal{F} \subseteq \mathbb{F}_q[x]$ as follows.

$$f(x) \in \mathcal{F} \iff f(x) = \sum_{i=0}^{u-1} \sum_{j=0}^{r-1} a_{ij}[g(x)]^i x^j + \sum_{j=0}^{v-1} a_{uj}[g(x)]^u x^j.$$

The resultant code can be shown to be an optimal (r, δ) all-symbol locality code having minimum distance

$$d_{\min}(\mathcal{C}) = (n-k+1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\delta - 1).$$

by arguing exactly as for the case $r|k$.

Example Construction

Let H be a multiplicative subgroup of $\mathbb{F}_q^*$ and let $S = \bigcup_{j=0}^{m-1} S_j$ be the union of m distinct cosets of H in $\mathbb{F}_q^*$. Thus we are assuming that $m|H| \leq (q-1)$. Let $m_H(x) = \prod_{h \in H}(x-h)$ be the annihilator polynomial of H.

Claim 1. $m_H(x)$ is constant on each multiplicative coset $S_j = \theta H,\ \theta \in \mathbb{F}_q^*$ of H.

Proof: Let $y \in S_j \implies y = \theta h',\ h' \in H$. Then

$$\begin{aligned} m_H(y) &= \prod_{h \in H} (\theta h' - h) = \prod_{h \in H} h'[\theta - (h')^{-1} h] \\ &= \prod_{h \in H} (\theta - h) = m_H(\theta) \end{aligned}$$

and is hence constant on the coset. □

We can use this fact to construct a good polynomial and hence a code with all-symbol locality by proceeding as follows. Let the field size q and $n, (r, \delta)$ be such that:

$$(r + \delta - 1) \mid n \mid (q-1).$$

Let α be an element of order n in $\mathbb{F}_q^*$ and set

$$\beta = \alpha^m,\ m = \frac{n}{(r+\delta-1)}.$$

Let H be the multiplicative subgroup of $\mathbb{F}_q^*$ given by

$$H \;=\; \{\beta^i \;\mid\; 0 \le i \le r+\delta-2\},$$

and let

$$S_j = \alpha^j H,\; 0 \le j \le m-1,$$

be the m distinct cosets of H in $\mathbb{F}_q^*$ and set

$$S \;=\; \bigcup_{j=0}^{m-1} S_j.$$

The annihilator $m_H(x)$ is given in this case by:

$$m_H(x) \;=\; x^{r+\delta-1} - 1$$

and is hence constant on the cosets $\{S_j\}_{j=0}^{m-1}$ of H. Also, $m_H(x)$ has degree $= (r+\delta-1)$ and hence $m_H(x)$ is a good polynomial. We can simplify the choice of good polynomial in this case by noting that the polynomial

$$g(x) \;=\; x^{r+\delta-1}$$

is a good polynomial as well.

Example: Let $q = 16$, $n = 15$, $r = 3$, $\delta = 3$, $(r+\delta-1) = 5$. Set $g(x) = x^5$. Suppose it is desired to construct an optimal code having dimension $k = 8$. Note that the expansion $k = ur + v$ gives us $u = v = 2$. We set

$$\mathcal{C} \;=\; \{(f(\theta), \theta \in \mathbb{F}_q^*) \;\mid\; f(\cdot) \in \mathcal{F}\},$$

where

$$f(x) \;\in\; \mathcal{F} \iff f(x) \;=\; \sum_{i=0}^{u-1}\sum_{j=0}^{r-1} a_{ij}[g(x)]^i x^j + \sum_{j=0}^{v-1} a_{uj}[g(x)]^u x^j.$$

This yields

$$f(x) \;=\; \sum_{i=0}^{1}\sum_{j=0}^{2} a_{ij} x^{5i+j} + \sum_{j=0}^{1} a_{2j} x^{10+j}.$$

This code has $d_{min} = 4$ which matches with

$$\begin{aligned} d_{\min} &\leq (n-k+1) - \left(\left\lceil \frac{k}{r} \right\rceil - 1\right)(\delta - 1) \\ &= (15-8+1) - \left(\left\lceil \frac{8}{3} \right\rceil - 1\right)(3-1) = 4. \end{aligned}$$

An analogous proof shows that the annihilator $m_H(x)$ of an additive subgroup H of $\mathbb{F}_q$ is also constant on each coset of H in $\mathbb{F}_q$ and may also be used to construct good polynomials and consequently codes with all-symbol locality as well.

10.8 Bounds on $d_{\min}$ and Rate for Nonlinear LRCs

We present below the proof of the bound on rate and minimum distance of a nonlinear LRC appearing in Theorem 6.

Proof: Let $\mathcal{C}$ be an $(n, M, d_{\min})$ code of size $M = q^k$, over an alphabet $\mathcal{A}_q$ of size q, having (r, δ) locality. We will establish the bound on minimum distance appearing in Theorem 6. The bound on code rate will then follow from Corollary 1. Recall from Definition 6, that associated to each code symbol c_i, there is a set $S_i \subseteq [n]$ of size $n_i := |S_i| \leq (r + \delta - 1)$ such that the restriction $\mathcal{C}_i := \mathcal{C}|_{S_i}$ of $\mathcal{C}$ to S_i is a code of block length n_i and minimum distance $\geq \delta$. Note by the Singleton bound that

$$|\, \mathcal{C}_i \,| \;\leq\; q^{n_i - \delta + 1} \;\leq\; q^r.$$

The minimum distance of the code $\mathcal{C}$ can be expressed in the form

$$d_{\min} \;=\; n \;-\; \max_{J \subseteq [n]} \{\; |J| \;\mid\; |\mathcal{C}_J| < q^k \;\}, \tag{10.6}$$

where $\mathcal{C}_J$ is the restriction of $\mathcal{C}$ to the coordinates in J. This follows since the minimum Hamming distance between a pair of distinct codewords is equal to n minus the maximum number of coordinates in which two distinct codewords can agree. We next select a set of $m = \lfloor \frac{k-1}{r} \rfloor$ local codes, which without loss of generality, we may assume to be the local codes $\mathcal{C}_j, j \in [m]$ in such a way that if $F_i = \cup_{j=1}^{i} S_j$, $i \in [m]$, then

$$|\, \mathcal{C}|_{F_i} \,| \;<\; |\, \mathcal{C}|_{F_{i+1}} \,|, \; i = 1, 2, \cdots, (m-1). \tag{10.7}$$

This is possible since,

$$|\,\mathcal{C}|_{F_m}\,| \leq \prod_{i=1}^{m} |\mathcal{C}_i| \;\leq\; q^{rm} \;<\; q^k. \tag{10.8}$$

For $1 \leq i \leq (m-1)$, let $P_i = F_i \cap S_{i+1}$. Note that by (10.7), P_i is a strict subset of S_{i+1}, i.e., $P_i \subsetneq S_{i+1}$. Let $\underline{x} \in \mathcal{C}_{i+1}|_{P_i}$. Consider the following subcode of $\mathcal{C}_{i+1}$

$$\mathcal{C}'_{i+1} \;=\; \{\underline{c} \in \mathcal{C}_{i+1} : \underline{c}|_{P_i} = \underline{x}\}.$$

Among all the possibilities for $\underline{x}$, we choose $\underline{x}$ such that $|\mathcal{C}'_{i+1}|$ is of maximum size. It is possible that $P_i = \phi$ in which case we will have $\mathcal{C}'_{i+1} = \mathcal{C}_{i+1}$. Clearly, we have $d_{\min}(\mathcal{C}'_{i+1}) \geq \delta$, as every local code is assumed to have minimum distance $\geq \delta$. Let us next puncture the code $\mathcal{C}'_{i+1}$ on P_i, i.e, let us pass on to the restriction $\mathcal{C}'_{i+1}|_{S_{i+1}\setminus P_i}$. The restriction will then be a code of the same size, of block length $|F_{i+1}| - |F_i|$ and minimum distance $\geq \delta$. The Singleton bound then gives us:

$$|\mathcal{C}'_{i+1}| \;\leq\; q^{|F_{i+1}|-|F_i|-\delta+1}.$$

From the maximal manner in which $\underline{x}$ was selected, we have:

$$\frac{|C|_{F_{i+1}}|}{|C|_{F_i}|} \;\leq\; |\mathcal{C}'_{i+1}|,$$

leading to:

$$\log_q \frac{|C|_{F_{i+1}}|}{|C|_{F_i}|} \;\leq\; |F_{i+1}| - |F_i| - \delta + 1,$$

i.e.,

$$\log_q \frac{|C|_{F_{i+1}}|}{|C|_{F_i}|} + \delta - 1 \;\leq\; |F_{i+1}| - |F_i|.$$

Summing both sides over the range $i = 1, 2, \cdots, m-1$, we obtain,

$$\log_q \frac{|C|_{F_m}|}{|C|_{F_1}|} + (m-1)(\delta - 1) \;\leq\; |F_m| - |F_1|.$$

Also, by the Singleton bound, we have:

$$\log_q |C|_{F_1}| + \delta - 1 \;\leq\; |F_1|.$$

Adding the two equations above, we get,

$$\log_q |C|_{F_m}| + m(\delta - 1) \ \leq \ |F_m|.$$

Let the integer $j > 0$ be such that $q^{k-j+1} > |\mathcal{C}|_{F_m}| \geq q^{k-j}$. We can then write

$$k - j + m(\delta - 1) \ \leq \ |F_m|.$$

Since $q^{k-j+1} > |\mathcal{C}|_{F_m}|$, it is possible to identify a subset $Q \subset [n] \setminus F_m$ of size $(j-1)$ such that if $F'_m = F_m \cup Q$, we have $|\mathcal{C}|_{F'_m}| < q^k$. This gives us:

$$\begin{aligned} k - j + m(\delta - 1) + j - 1 \ &\leq \ |F_m| + |Q| \ = \ |F'_m| \\ &\implies k - 1 + m(\delta - 1) \ \leq \ |F'_m|. \end{aligned}$$

The equation above, combined with (10.6), gives us

$$d_{\min} \ \leq \ n - |F'_m| \leq n - k + 1 - m(\delta - 1),$$

i.e.,

$$d_{\min} \ \leq \ n - k + 1 - \left(\left\lfloor \frac{k-1}{r} \right\rfloor \right) (\delta - 1)$$

leading to the desired bound

$$d_{\min} \ \leq \ n - k + 1 - \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right) (\delta - 1).$$

□

10.9 Extended Notions of Locality

The theory of LRCs has been extended in various directions and an overview of these different extensions is provided in Fig. 10.2. Availability codes, codes with sequential recovery, codes with hierarchical locality and maximally recoverable codes are discussed in Sections 11, 12, 13 and 14 respectively. A brief discussion of LRCs with cooperative recovery appears in the present section.

A listing of the code constructions appearing in Sections 11, 12, 13 and 14, is provided in Table 10.2.

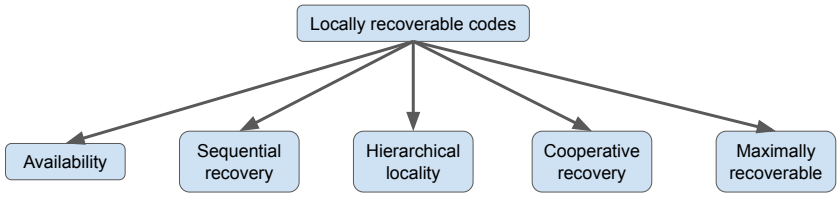

Figure 10.2: Extended notions of locality.

Table 10.2: This table provides a listing of the constructions for availability codes, codes with sequential recovery, codes with hierarchical locality and maximally recoverable codes (MRCs) that appear in Sections 11, 12, 13 and 14 respectively. With the exception of the construction of the MRC based on the Combinatorial Nullstellensatz, all other constructions appearing in the table are explicit.

Type of extended LRC	Code	Section
Codes with Availability	Product Code	11.2.1
Codes with Availability	Wang *et al.* Code [246]	11.2.2
Codes with Sequential Recovery	Near-Regular Graph Code [174]	12.1
Codes with Sequential Recovery	2 Dimensional Product Code [226]	12.1
Codes with Sequential Recovery	Graph based Construction [13] (Example)	12.2.2
Codes with Hierarchical Locality	Chinese Remainder Theorem based Construction [199] (Example)	13.2
Maximally Recoverable Codes	Combinatorial Nullstellensatz based MRC	14.3
Maximally Recoverable Codes	Linearized Polynomials based MRC	14.4
Maximally Recoverable Codes	Linearized Polynomials based MRC with Reduced Field Size [71]	14.5

10.9.1 Cooperative Local Recovery

An $[n, k]$ code with (r, t)-cooperative locality is a code such that if a subset $(c_{i_1}, c_{i_2}, \cdots, c_{i_t})$ of code symbols are erased, then there exists a second subset $\{c_{j_1}, c_{j_2}, \cdots, c_{j_r}\}$ of r other code symbols (i.e., $i_a \neq j_b$ for any pair (a, b)) such that for all $a \in [t]$:

$$c_{i_a} = \sum_{b=1}^{r} \theta_{a,b} c_{j_b}, \quad \theta_{a,b} \in \mathbb{F}_q.$$

In [192], the authors provide the following bound on minimum distance of codes with cooperative locality:

$$d_{\min}(n, k, r, t) \leq n - k + 1 - t \left\lfloor \frac{k-t}{r} \right\rfloor.$$

The same paper also contains the following alphabet-size dependent bound on dimension:

$$k \leq \min_{\gamma \leq \min(\lfloor \frac{n}{r+t} \rfloor, \lfloor \frac{k-1}{r} \rfloor)} r\gamma + \log_q(A_q(n - \gamma(r+t), d)),$$

where $A_q(n, d)$ is the maximum size of a q-ary code of block length n and minimum distance d. Constructions of codes with cooperative locality are also provided in [192].

Notes

Unless otherwise specified, when we speak of an LRC in this notes subsection, we will mean a linear LRC.

1. As mentioned in the introduction, the concept of locality can be found discussed in the early papers [87], [102], [164] as well as in the subsequent paper [72], that explored the topic in greater depth. The treatment in [72] focuses on the linear case, includes upper bounds on the minimum distance, the identification of optimal codes as well as generalizations. The class of optimal linear codes having information-symbol locality includes the class of pyramid codes, that first appeared in [102]. The extension to the case where the codes are linear and where the local codes have minimum distance ≥ 2 appears in [172]. Generalization of LRCs to the non-linear case can be found in [68], [168], [228] as well as in the early paper [87]. The paper [168] also considers the case of LRCs over a vector code-symbol alphabet.

2. Codes achieving the minimum distance bound for general δ:

   (a) The pyramid code [102] discussed in Section 10.5 is an example of a construction for an information-symbol locality code, achieving the upper bound on minimum distance, and having field size that is linear in the block length n.

   (b) Constructions of all-symbol locality codes achieving the upper bound on minimum distance with field size linear in the block length n for the case $(r + \delta - 1)|n$ appear in [38], [172], [228].

A detailed investigation of codes achieving the $d_{\min}$ upper bound can be found in [225].

3. Codes with all-symbol locality for the case $\delta = 2$:

   (a) For the status on constructions for the case when $(r+1)|n$, please see note above and set $\delta = 2$.

   (b) A construction with field size linear in n and minimum distance within 1 of the upper bound on $d_{\min}$ can be found in [228] for the case $r \nmid k$ and $n \neq 1 \mod (r+1)$.

   (c) A construction with minimum distance within 1 of the upper bound on $d_{\min}$ for all parameters, having exponential field size appears in [66].

   (d) A construction achieving the minimum distance bound under the assumption of disjoint repair sets that holds for all $n \leq q$ and all n with $n \mod (r+1) \neq 1$ appears in [123]. The condition $n \mod (r+1) \neq 1$ is removed in [176] and an optimal code construction where optimality is under the assumption of disjoint repair sets is provided in [176], for all $n \leq q$, where q is the field size.

   (e) Upper bounds on $d_{\min}$ tighter than the one given in [72], can be found in [160], [175], [242], [261]. Constructions for codes achieving the tightened bound in [242] for the case of $n_1 > n_2$ where $n_1 = \lceil \frac{n}{r+1} \rceil$, $n_2 = n_1(r+1) - n$ and having exponential field size can also be found there.

In summary, for the case $\delta = 2$, the problem of constructing optimal LRCs with field size linear in block length is completely solved for the cases (a) $(r+1)|n$ and (b) for any $n \leq q$ under the assumption of disjoint repair sets.

Open Problem 10. Construct an optimal LRC for the case ($\delta = 2$) with field size that is linear in the block length n for the general case when $(r+1) \nmid n$ and where the repair sets are not necessarily disjoint.

Open Problem 11. Let $n_1 = \lceil \frac{n}{r+1} \rceil$, and $n_2 = n_1(r+1) - n$. Construct LRCs for the case ($\delta = 2$) with best possible minimum distance for the case when $n_1 \leq n_2$ and where the repair sets are not necessarily disjoint. (There is no constraint on field size here).

4. On the construction of LRCs with large block length for given field size: There is interest in determining the largest possible block length of a code with locality that achieves the upper bound in (10.1) on minimum distance, for a given field size. This is analogous to the problem of determining the maximum possible block length n for which an MDS code having field size q exists. The focus of the research effort here has been on the case $\delta = 2$. Constructions for LRCs with block length exceeding the field size q can be found described in [21], [112], [142]. Bounds on the maximum possible block length n of an LRC having minimum distance achieving (10.1) for a given field size q can be found in [86], [90]. In [34], the authors focus on the general case $\delta > 2$, derive an upper bound on the maximum length possible and provide a construction having length that is super-linear in the size q of the underlying finite field.

5. LRCs with all-symbol locality and small alphabet size for the case $\delta = 2$:

   (a) Upper bounds on $d_{\min}$ for given (n, k, r) or on dimension k for given $(n, d_{\min}, r)$: For fixed alphabet size q, upper bounds can be found in [14], [30], [109]. In the binary ($q = 2$) case, a Hamming-like upper bound on dimension appears in [243]. Upper bounds assuming disjoint repair sets can be found in [1], [74], [155], [243], [259]. Upper bounds assuming that the code is cyclic appear in [231]. Asymptotic upper bounds i.e., upper bounds on code rate as a function of fractional minimum distance $\frac{d_{\min}}{n}$, can be found in [1].

   (b) Constructions: Constructions for cyclic LRCs were introduced in [74] and these codes are for specific values of $(d_{\min}, r)$ and are optimal w.r.t. upper bounds derived in the same paper.

Construction of cyclic LRCs achieving the upper bound on minimum distance given in [72] with field size linear in block length n can be found in [231]. The authors also study cyclic LRCs with smaller field size obtained by looking at sub-field subcodes and trace codes. The locality property of classical binary cyclic codes and codes that can be obtained from them by operations such as shortening can be found in [108]. Construction of cyclic LRCs with local codes that are not MDS codes, can be found in [259] and these codes are optimal w.r.t. upper bounds derived in the same paper. Construction of optimal cyclic LRCs for specific values of $(d_{\min}, r)$ can be found in [119], [154]. Other optimal constructions that are not necessarily cyclic and for specific values of $(d_{\min}, r)$ can be found in [88], [89], [92], [142], [155], [163], [214], [219], [243]. Constructions having good performance with respect to code rate for a fixed alphabet size q and fractional minimum distance can be found in [30], [230].

(c) Constructions of LRCs based on algebraic geometry (AG) codes can be found in [19], [21], [112], [141], [142], [197]. Part of the motivation for exploring the use of AG codes, comes from the fact that for a fixed field size, AG codes can have larger block length in comparison with an MDS code. The authors of [19] were also led to study AG codes as a means of generalizing the construction of LRCs from RS codes in [228], which could be viewed as arising from a simple covering map of the projective line. The idea here is to replace the simple covering map by a covering map from one curve to another, for example from the Hermitian curve to the projective line. In [142], the authors construct optimal LRCs of length larger than the size q of the underlying finite field using elliptic curves. In [141], the authors extend the construction in [19] and make use of the automorphism group of a tower of function fields to derive asymptotically-good LRCs. The construction in [112] is based on automorphism groups of rational function fields.

6. Codes with all-symbol locality for general δ and having small alphabet size: Upper bounds on dimension for a given alphabet size q can be found in [1]. Constructions yielding asymptotic lower bounds on rate for a fixed alphabet size q as a function of fractional minimum distance can be found in [19]. In [91], the authors prove that for $\delta > 2$ there are only two classes of binary codes which achieve the upper bounds given in [172].

7. In a different direction, in [131] the authors explore the use of locality for reducing the decoding complexity of a cyclic code.

11

Codes with Availability

This section, as well as the two sections that follow, may be viewed as providing additional, alternative approaches to handling multiple node failure.

Codes with availability, discussed in the present section, provide multiple, node-disjoint means of accessing the data contained within a particular node, thereby enabling recovery from multiple-node failure. Availability codes have an additional appeal: they are able to handle multiple, simultaneous requests for the data contained within a particular node, a useful feature when storing popular content. The notion of codes with availability was introduced by Wang and Zhang in [241] in the setting of linear codes. As in the case of LRCs, we begin with a discussion of the more general case of nonlinear codes with availability, before specializing to the linear case.

Definition 9 (Availability Code). An $(n, M, d_{\min})$ code $\mathcal{C}$ over an alphabet $\mathcal{A}_q$ of size q is said to be a code with availability with parameters (r, t), if for each code symbol $c_i, i \in [n]$, there are t disjoint repair sets

$$\{\, R_{ij} \subseteq [n] \;\mid\; |R_{ij}| \leq r,\ i \notin R_{ij} \quad j = 1, 2, \cdots, t\},$$

and t associated functions $\{f_{ij} : \mathcal{A}_q^{|R_{ij}|} \to \mathcal{A}_q\}$ such that

$$c_i = f_{ij}((c_\ell, \ell \in R_{ij})), \text{ all } j = 1, 2, \cdots, t.$$

Theorem 8 (Upper Bound on Rate and $d_{\min}$ for Nonlinear Codes [230])**.** Let $\mathcal{C}$ be an $(n, M, d_{\min})$ code over an alphabet $\mathcal{A}_q$ of size q with availability having parameters (r, t) and where the repair sets $\{R_{ij} \mid i \in [n], j \in [t]\}$ are of constant size $|R_{ij}| = r$. Let $k = \lfloor \log_q(M) \rfloor$. The rate and minimum distance of $\mathcal{C}$ then satisfy the upper bounds

$$\frac{k}{n} \leq \frac{1}{\prod_{j=1}^{t}\left(1 + \frac{1}{jr}\right)}, \tag{11.1}$$

$$d_{\min} \leq n - \sum_{i=0}^{t} \left\lfloor \frac{k-1}{r^i} \right\rfloor. \tag{11.2}$$

In our proof of the rate bound, we follow Tamo *et al.* [230] and refer the reader to [230] for a proof of the upper bound on minimum distance.

The rate bound will follow from Lemma 10 and Lemma 11, given below. The proofs adopt a graphical approach that involves associating a directed graph on n nodes with the code, called the recovery graph. The ith node is associated to code symbol c_i. The edges are colored using one of t colors which we associate with elements of the set $[t]$. There is a directed edge bearing color ℓ, $\ell \in [t]$, from node j to node i iff $j \in R_{i\ell}$. Next, a random permutation $\pi(\cdot)$ of the set $[n]$ is chosen and the nodes are linearly ordered from left to right with the ith node appearing in position $\pi(i)$. We then turn to a coloring of the nodes. Node i is assigned color ℓ iff

$$\pi(j) < \pi(i), \ \forall\ j \in R_{i\ell}.$$

It is possible for a node to be assigned up to a maximum of t colors. It is also possible that a node is not assigned any color, i.e., is left uncolored. Clearly for a fixed permutation $\pi(\cdot)$, if the values of the code symbols associated with the uncolored nodes are known, then all the remaining symbols can be determined. It follows that if k_u is the number of uncolored nodes under a given permutation $\pi(\cdot)$, that the file size of the availability code is bounded above by $M \leq q^{k_u}$. This is illustrated in Fig. 11.1 for an example linear code of block length $n = 5$.

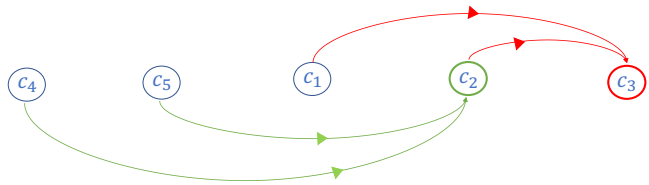

Figure 11.1: Illustrating the recovery graph for a binary linear code, satisfying the parity checks: $c_1 + c_2 = c_3$, $c_2 + c_4 = c_5$. The nodes are ordered in accordance with a random permutation π, so that the node associated to c_i appears in position $\pi(i)$. Here $(\pi(1), \pi(2), \pi(3), \pi(4), \pi(5)) = (3, 4, 5, 1, 2)$. The edges in red are associated to the first p-c equation and the edges in green, with the second (only the edges relevant to node coloring are shown). Under this permutation, node 3 is colored red and node 2 is colored green. The number 3 of uncolored symbols leads to the upper bound $M \leq q^3$ on code size.

Lemma 10. Let $\mathcal{C}$ be an $(n, M, d_{\min})$ code having availability parameters (r, t) and constant repair-set size, i.e., $|R_{ij}| = r$ for all $\{i, j\}$. Then there exists a coordinate permutation $\pi : [n] \to [n]$ such that under the ordering of code symbols determined by π, the number $|U|$ of colored nodes, i.e., nodes that are assigned at least one color as per the method of assigning colors described above, satisfies the lower bound:

$$|U| \quad \geq \quad n\left(1 - \frac{1}{\prod_{j=1}^{t}\left(1 + \frac{1}{jr}\right)}\right).$$

Proof. Let us pick a permutation π randomly from the $n!$ possibilities. We color the nodes of the recovery graph associated to the code as described above, for the ordering defined by π. Let A_{ij} be the event that node i, i.e., the node associated to code symbol c_i, is colored with color j. Let U denote the set of colored nodes. Clearly, U is a function of the particular realization of the random permutation $\pi(\cdot)$. We have

$$P(i \in U) = P(\cup_{j=1}^{t} A_{ij}),$$

where $P(\cdot)$ denotes the probability function. We can employ the inclusion-exclusion principle to calculate the above probability if we know $P(\cap_{j \in S} A_{ij})$ for every subset $S \subseteq [t]$. This latter probability is the probability of the event that node i is colored with all colors in the set S which

implies the event $\pi(\ell) < \pi(i)$ for all $\ell \in R_{ij}, j \in S$. It follows that in the linear ordering determined by π, the code symbol c_i must necessarily appear to the right of all the code symbols $c_m,\ m \in \cup_{j\in S} R_{ij}$. If we restrict attention to the set of code symbols $\{c_m \mid m \in \{i\} \cup_{j\in S} R_{ij}\}$, all orderings of these symbols are equally likely. Hence the probability that code symbol c_i ends up in the rightmost position within this set, is given by:

$$\begin{aligned} P(\cap_{j\in S} A_{ij}) &= \frac{1}{|\cup_{j\in S} R_{ij}| + 1} \\ &= \frac{1}{\sum_{j\in S} |R_{ij}| + 1} = \frac{1}{|S|r + 1}. \end{aligned}$$

By the inclusion-exclusion principle, we have:

$$\begin{aligned} P(i \in U) &= P(\cup_{j=1}^{t} A_{ij}) \\ &= \sum_{j=1}^{t} (-1)^{j-1} \binom{t}{j} P(A_{i1} \cap A_{i2} \cap \ldots \cap A_{ij}) \\ &= \sum_{j=1}^{t} (-1)^{j-1} \binom{t}{j} \frac{1}{jr + 1}. \end{aligned}$$

Through algebraic manipulation, this can be reduced to:

$$P(i \in U) = \left(1 - \frac{1}{\prod_{j=1}^{t} \left(1 + \frac{1}{jr}\right)}\right).$$

The expected value of the number of colored codes is then given by:

$$\begin{aligned} E(|U|) &= \sum_{i} P(i \in U) \\ &= n \left(1 - \frac{1}{\prod_{j=1}^{t} \left(1 + \frac{1}{jr}\right)}\right). \end{aligned}$$

The proof is completed by observing that there exists at least once choice of π for which $|U| \geq E(|U|)$. □

Lemma 11. Let $\mathcal{C}$ be an $(n, M, d_{\min})$ code over an alphabet $\mathcal{A}_q$ of size q with availability parameters (r, t) and repair sets R_{ij} of constant size

$|R_{ij}| = r$ for all $\{i, j\}$. Let π be an arbitrary permutation on $[n]$ and let the nodes of the associated recovery graph be colored as described above, under the ordering of nodes associated to $\pi(\cdot)$. Let U be the set of colored nodes. Then we have the upper bound

$$M \leq q^{n-|U|},$$

on code size.

Proof. Follows from the fact that under any ordering of code symbols, the code symbols associated to colored nodes can be determined given the values of the code symbols associated to uncolored nodes. □

The upper bound on rate given in Theorem 8 then follows from an application of Lemmas 10 and 11.

11.1 Linear Availability Codes

As in the case of LRCs, there is greatest interest in the linear case, and we provide below, a formal definition of a linear availability code.

Definition 10 (Linear Availability Code)**.** An $[n, k, d_{\min}]$ code $\mathcal{C}$ over a field $\mathbb{F}_q$, is said to be a code with availability with parameters (r, t), if for each code symbol c_i there are t disjoint sets

$$\{\ R_{ij} \subseteq [n]\ \mid\ |R_{ij}| \leq r,\ i \notin R_{ij}\ \ j = 1, 2, \cdots, t\ \},$$

such that

$$c_i = \sum_{\ell \in R_{ij}} a_{ij\ell}\ c_\ell,\ a_{ij\ell} \in \mathbb{F}_q, \quad \text{holds for}\ \ j = 1, 2, \cdots, t.$$

11.2 Constructions of Linear Availability Codes

11.2.1 The Product Code Construction

The t-fold product of the $[r+1, r]$ single-parity-check code gives rise to a t-dimensional $[(r+1)^t, r^t]$ product code. It is straightforward to verify that this code is an availability code with availability parameters (r, t) having code rate

$$\frac{k}{n} = \frac{r^t}{(r+1)^t}.$$

11.2.2 A High-Rate Construction

We present here a construction due to Wang *et al.* [246] of an (r,t) availability code $\mathcal{C}$ having parameters

$$\left(n = \binom{r+t}{t}, \quad k = \binom{r+t}{t} - \binom{r+t-1}{t-1} \right).$$

In comparison with the product code, this code not only has a significantly improved rate given by,

$$\frac{k}{n} = \frac{r}{r+t},$$

it also has shorter block length.

Construction 2 (Wang *et al.* [246])**.** Let us define the sets:

$$\begin{aligned} C &= \{S \subseteq [r+t] : |S| = t\} \\ R &= \{S \subseteq [r+t] : |S| = t-1\}. \end{aligned}$$

We will assume that the sets C and R are lexicographically ordered. By lexicographic ordering, we mean the following. In the ordering, a set $E_1 = \{i_1, \ldots, i_t\}$ with $i_1 < i_2 < \ldots < i_t$ appears before a set $E_2 = \{j_1, \ldots, j_t\}$ with $j_1 < j_2 < \ldots < j_t$ iff for some $\ell \in [1,t]$, $i_p = j_p$ for all $p \in [1,\ell]$ and $i_{\ell+1} < j_{\ell+1}$. Clearly, $|C| = \binom{r+t}{t} := n(r,t)$ and $|R| = \binom{r+t}{t-1} := m(r,t)$. Define an $(m(r,t) \times n(r,t))$ binary matrix $H(r,t)$ whose (i,j)th entry h_{ij} is given by:

$$h_{ij} = \begin{cases} 1, & \text{if } R(i) \subseteq C(j), \\ 0, & \text{otherwise,} \end{cases}$$

where $R(i)$ is the ith element in R and $C(j)$ is the jth element in C, both under lexicographic ordering. We define an $[n(r,t), k]$ code $\mathcal{C}$ with parameters (r,t) as the linear, binary code with p-c matrix $H(r,t)$ where $H(r,t)$ is as defined above.

We first establish that the code is an (r,t) availability code as claimed. Following this, we will go on to provide an expression for the dimension k and rate $\frac{k}{n}$ of the code.

Lemma 12. The code $\mathcal{C}$ is a code with availability with parameters (r,t).

Proof. Clearly, each row of H has Hamming weight $(r+1)$ and each column of H has Hamming weight t. We claim that the real inner product of any two rows of H is ≤ 1. Suppose the inner product of two distinct rows i_1 and i_2 of H is ≥ 2. This is possible iff there exist two distinct column indices j_1, j_2 such that

$$\begin{aligned} R(i_1) &\subseteq C(j_1) \cap C(j_2), \\ R(i_2) &\subseteq C(j_1) \cap C(j_2), \end{aligned}$$

But this is impossible since

$$R(i_1) \subseteq C(j_1) \cap C(j_2) \Rightarrow R(i_1) = C(j_1) \cap C(j_2),$$

and as a result, i_1 is uniquely determined from (j_1, j_2). It follows that $\mathcal{C}$ is an availability code having parameters (r, t). □

The two lemmas below will show that the code $\mathcal{C}$ has dimension

$$k = n(r,t) - \binom{r-1+t}{t-1} = \binom{r+t}{t} - \binom{r-1+t}{t-1},$$

and hence rate

$$\frac{k}{n} = \frac{\binom{r+t}{t} - \binom{r-1+t}{t-1}}{\binom{r+t}{t}} = \frac{r}{r+t}.$$

Lemma 13. The matrix $H(r,t)$ has the following recursive structure

$$H(r,t) = \begin{bmatrix} H(r,t-1) & 0 \\ \underbrace{I}_{n(r,t-1) \text{ columns}} & \underbrace{H(r-1,t)}_{n(r,t)-n(r,t-1) \text{ columns}} \end{bmatrix}$$

Proof. There are four blocks in the above recursive structure for $H(r,t)$. We prove the presence of these four blocks separately.

- By lexicographic ordering, the first $n(r,t-1)$ subsets in columns i.e., the sets $C(i)$, $1 \leq i \leq n(r,t-1)$ all contain the element 1. Similarly the first $m(r,t-1)$ subsets in rows i.e., the sets $R(i)$, $1 \leq i \leq m(r,t-1)$ all contain the element 1. Hence 1 is fixed in all these sets and we can think of them as if t has reduced by one. Hence the first $n(r,t-1)$ columns and the first $m(r,t-1)$ rows of $H(r,t)$ has the form: $H(r,t-1)$.

- The rest of the subsets in columns $C(i)$, $n(r,t-1)+1 \leq i \leq n(r,t)$ are sets which does not contain 1 which means these are subsets of $[2, r+t]$ of size t and the rest of the subsets in rows $R(i)$, $m(r,t-1)+1 \leq i \leq m(r,t)$ are sets which does not contain 1 which means these are also subsets of $[2, r+t]$ of size $t-1$ and this equivalent to saying r has reduced by one. Hence $H(r,t)$ in these columns and rows correspond to $H(r-1,t)$.
- The first $m(r,t-1)$ rows of the columns $C(i)$, $n(r,t-1)+1 \leq i \leq n(r,t)$ has zeros as these columns are subsets which does not contain 1 and rows are subsets which contain 1.
- The rest of subsets in rows $R(i)$, $m(r,t-1)+1 \leq i \leq m(r,t)$ are sets which does not contain 1 which means these are subsets of $[2, r+t]$ of size $t-1$. Hence these subsets when added with $\{1\}$ forms a unique subset containing 1 of size t. Hence we get the identity part.

□

Lemma 14. By Lemma 13, the matrix $H(r,t)$ has the recursive structure:

$$H(r,t) = \begin{bmatrix} H(r,t-1) & 0 \\ I & H(r-1,t) \end{bmatrix}$$

In this recursive structure we have that

$$\operatorname{rank}(H(r,t)) = \operatorname{rank}(H'),$$

where

$$H' = \left[\begin{array}{c|c} I & H(r-1,t) \end{array}\right].$$

It follows that the dimension k of the availability code $\mathcal{C}$ is given by:

$$k = n(r,t) - \binom{r-1+t}{t-1}.$$

Proof. The row-reduction process in block-matrix form applied to

$$H(r,t) = \begin{bmatrix} H(r,t-1) & 0 \\ I & H(r-1,t) \end{bmatrix}$$

gives us the matrix

$$\begin{bmatrix} 0 & -H(r,t-1)H(r-1,t) \\ I & H(r-1,t) \end{bmatrix}.$$

We claim that

$$H(r,t-1)H(r-1,t) \;=\; 0.$$

Clearly, if we can show this, this establishes the Lemma. We will show the product $H(r,t-1)H(r-1,t)$ to be the zero matrix by showing that each inner product of a row in $H(r,t-1)$ and a column in $H(r-1,t)$ over $\mathbb{F}_2$ equals 0. Each row of $H(r,t-1)$ is associated to a subset D of size $(t-2)$ drawn from a set of size $(r+t-1)$. Each column of $H(r-1,t)$ is associated to a subset F of size t drawn from a set of size $(r+t-1)$. The inner product is precisely equal to the number of subsets E that satisfy

$$D \subseteq E \subseteq F,$$

given that

$$|D| = (t-2),\ |E| \;=\; (t-1),\ |F| = t.$$

It follows that this number is either 0 or 2. Thus the inner product in either case, is equal to 0 (mod 2). Hence

$$H(r,t-1)H(r-1,t) \;=\; 0.$$

□

11.3 Upper Bounds on $d_{\min}$ of Linear Availability Codes

11.3.1 Bounds Depending upon Field Size q

Most of the upper bounds in the literature on the minimum distance of a linear code $\mathcal{C}$ with availability, and that take into account the size q of the underlying finite field, are based on the following code-shortening approach. Let G be a generator matrix of an (n,k,r,t,q), linear availability code $\mathcal{C}$ with maximum possible minimum distance. Let $S \subseteq [n]$ and let $G|_S$ denote the corresponding sub-matrix of G. Let

$s = |S|$ and ν be an upper bound to the rank of $G|_S$. Let $\mathcal{C}_S = \{\underline{c}|_{[n]\setminus S} : \underline{c} \in \mathcal{C}, \underline{c}|_S = 0\}$ denote the code of block length $(n-s)$ obtained by shortening $\mathcal{C}$ with respect to S. Then we can upper bound the minimum distance of $\mathcal{C}$ via

$$\begin{aligned} d_{\min}(n,k,r,t,q) = d_{\min}(\mathcal{C}) &\leq \min_{\{S:S\subseteq[n],\nu<k\}} d_{\min}(\mathcal{C}_S) \\ &\leq \min_{\{S:S\subseteq[n],\nu<k\}} d_{\min}(n-s,k-\nu,r,t,q), \end{aligned}$$

where $d_{\min}(n,k,r,t,q)$ is the maximum possible minimum distance of an $[n,k]$ linear code over $\mathbb{F}_q$ with availability with parameters (r,t). The bound presented in Theorem 9 below is derived in terms of generalized Hamming weights (GHW) of the dual code $\mathcal{C}^\perp$. GHWs are defined below.

Definition 11. [250] The ith generalized Hamming weight[1] (GHW) d_i of a code $\mathcal{C}$ is the smallest support of an i-dimensional subcode of $\mathcal{C}$. We use $d_i^\perp$ to denote the ith GHW of the dual code $\mathcal{C}^\perp$.

The shortening approach and the approach via GHW are closely connected. Knowing the GHW of the dual code makes it easier to identify candidate sets S to be used in conjunction with the shortening approach. However, in practice, the GHW of the dual code may not be precisely known, while upper bounds to the GHW of the dual code, might be more easily available. For this reason, the bound in Theorem 9 below, is phrased in terms of upper bounds $d_i^\perp \leq e_i,\ i = 1, 2, \cdots, b$ on the first $1 \leq b \leq n-k$ GHWs of the dual code.

Theorem 9. [14] Let $\mathcal{C}$ be an $[n,k]$ linear availability code with parameters (r,t) over the field $\mathbb{F}_q$. Then

$$d_{\min}(n,k,r,t,q) \leq \min_{i\in T} d_{\min}(n-e_i, k+i-e_i, r, t, q)$$

where $T := \{i : e_i - i < k,\ 1 \leq i \leq b\}$ and $b \in [1, n-k]$ and where the integers e_i for $1 \leq i \leq b$, must have the property that they upper bound the corresponding GHW, i.e., $d_i^\perp \leq e_i$.

[1] Generalized Hamming weights are at times referred to in the literature, as minimum support weights, see for example [94].

Proof. As noted above, the $\{e_i\}_{i=1}^{b}$ play the role of known upper bounds on the GHW of the dual code, as in most cases, the exact GHW of the dual code will be unknown. Let the support of a subcode of $\mathcal{C}^{\perp}$ of dimension i be S_i, where $|S_i| = d_i^{\perp}$. Let some arbitrary indices of code symbols be added to S_i so that the augmented set satisfies $|S_i| = e_i$. Next, let $\mathcal{C}$ be shortened at the co-ordinates indexed by S_i i.e.,

$$\mathcal{C}_{\text{shorten}} \quad = \quad \{\underline{c}|_{[n]\setminus S_i} : \underline{c} \in \mathcal{C}, \underline{c}|_{S_i} = \underline{0}\}.$$

It follows that $\mathcal{C}_{\text{shorten}}$ is also an availability code having parameters (r, t) and block length $(n - e_i)$. We claim that $\mathcal{C}_{\text{shorten}}$ has dimension $\geq n - e_i - (n - k - i) \; = \; (k + i - e_i)$. This can be seen as follows. From the definition of GHW, the p-c matrix of the code $\mathcal{C}$ can be written in the following form:

$$H = \begin{array}{c} i \text{ rows} \\ n-k-i \text{ rows} \end{array} \begin{array}{c} \{ \\ \{ \end{array} \left[\begin{array}{cc} H_i & 0 \\ \underbrace{A}_{\text{co-ordinates in } S_i \ (\ e_i \text{ columns})} & \underbrace{H'}_{n-e_i \text{ columns}} \end{array} \right].$$

In the above, H' is the p-c matrix of $\mathcal{C}_{\text{shorten}}$. Clearly, $\text{rank}(H') \leq (n-k-i)$ and it follows that $\mathcal{C}_{\text{shorten}}$ has dimension $\geq n-e_i-(n-k-i) = k+i-e_i$. We also have:

$$\begin{aligned} d_{\min}(\mathcal{C}) &\leq d_{\min}(\mathcal{C}_{\text{shorten}}) \\ &\leq d_{\min}(n - e_i, k + i - e_i, r, t, q). \end{aligned}$$

The bound follows. The restriction $e_i - i < k$ appearing in the definition of the set T in the Theorem is to ensure that the code whose minimum distance appears on the right has dimension ≥ 1. □

For the choice of b and e_i, $1 \leq i \leq b$ appearing in equation (11.5) below, the bound in Theorem 9, is tighter than the corresponding bounds based on the shortening approach, that appear in [109], [133] for $t \geq 2, r \geq 2$. A different upper bound on minimum distance, also based on GHW, appears in [133].

11.3.2 Field-Size-Independent Bounds

The field-size dependent bound appearing in Theorem 9, can be converted into one that is independent of field size, and is given in the corollary below.

Corollary 2. [14] Let $d_{\min}(n,k,r,t)$ be the maximum-possible minimum distance of an $[n,k]$ linear code $\mathcal{C}$ with availability with parameters (r,t). Then

$$\begin{aligned} d_{\min}(n,k,r,t) &\leq \min_{i\in T} d_{\min}(n-e_i, k+i-e_i, r, t) \\ &\leq \min_{i\in T} n-k-i+1, \end{aligned} \tag{11.3}$$

where $T=\{i : e_i - i < k, 1 \leq i \leq b\}$ and $b \in [1, n-k]$ and $d_i^{\perp} \leq e_i$ for $1 \leq i \leq b$.

Proof. From Theorem 9 we have that,

$$\begin{aligned} d_{\min}(n,k,r,t,q) &\leq \min_{i\in T} d_{\min}(n-e_i, k+i-e_i, r, t, q), \\ \Rightarrow \max_q d_{\min}(n,k,r,t,q) &\leq \min_{i\in T} \max_q d_{\min}(n-e_i, k+i-e_i, r, t, q), \\ \Rightarrow d_{\min}(n,k,r,t) &\leq \min_{i\in T} d_{\min}(n-e_i, k+i-e_i, r, t). \end{aligned}$$

Applying the Singleton bound to $d_{\min}(n-e_i, k+i-e_i, r, t)$, we obtain the bound in (11.3). □

Remark 10. Most of the field-size-independent upper bounds in the literature on the minimum distance of a linear code $\mathcal{C}$ with availability, rely first on finding a set $S \subset [n]$ such that $\text{rank}(G|_S) \leq k-1$ where G is the generator matrix and $G|_S$ is the generator matrix restricted to columns indexed by S. This then leads to the upper bound,

$$d_{\min} \leq n - |S|.$$

We show below how the very same bound can be obtained by applying Corollary 2 with $i = |S| - k + 1$, and $e_i = |S|$. The motivation for making this connection, is that this will make it easier to compare prior bounds in the literature, with the bound appearing in Corollary 2.

The dual of the restriction $\mathcal{C}|_S$ of the code $\mathcal{C}$ to the set S, is a shortened version of the dual $\mathcal{C}^{\perp}$ having dimension $\geq |S| - k + 1$. The definition of GHWs of the dual $\mathcal{C}^{\perp}$, allows us to conclude that: $d^{\perp}_{|S|-k+1} \leq |S|$. This now allows us to apply the bound

$$d_{\min} \leq \min_{i\in T} n-k-i+1$$

in Corollary 2 with $i = |S| - k + 1$, and $e_i = |S|$ since the required conditions

$$\begin{aligned} d^{\perp}_{|S|-k+1} &\leq e_i = |S|, \\ e_i - i &< k, \end{aligned}$$

are both satisfied. We end up, as mentioned earlier, with the very same bound:

$$d_{\min} \leq \min_{i \in T} n - k + 1 - (|S| - k + 1) = n - |S|.$$

Remark 11. By the remark above, prior bounds appearing in the literature, correspond to different choices of i in the bound in (11.3). All choices of i can be shown to satisfy the requirements

$$\begin{aligned} d^{\perp}_i &\leq e_i, \\ e_i - i &< k, \end{aligned}$$

for a suitable value of e_i. We identify the value of i employed in the various bounds in the following. Let $t \geq 2, r \geq 2$.

1. The bound in [241] can be obtained by setting:

$$\begin{aligned} i &= \left\lceil \frac{t(k-1)+1}{t(r-1)+1} \right\rceil - 1 = \left\lceil \frac{t(k-r)}{t(r-1)+1} \right\rceil \\ &\leq \left\lceil \frac{k-r}{r-1} \right\rceil \leq \begin{cases} \left\lfloor \frac{k-2}{r-1} \right\rfloor, & \text{if } (r-1) \mid (k-1) \\ \left\lfloor \frac{k-1}{r-1} \right\rfloor, & \text{otherwise.} \end{cases} \end{aligned}$$

2. The bound presented above in Theorem 8 and appearing in [230] is applicable to the case of a general nonlinear code. However, when specialized to the linear case, the same minimum distance bound can be obtained by setting :

$$i = \sum_{i=1}^{t} \left\lfloor \frac{k-1}{r^i} \right\rfloor \leq \begin{cases} \left\lfloor \frac{k-2}{r-1} \right\rfloor, & \text{if } (r-1) \mid (k-1) \\ \left\lfloor \frac{k-1}{r-1} \right\rfloor, & \text{otherwise.} \end{cases}$$

3. A particular bound in [133] can be obtained by setting:

$$i = \left\lfloor \frac{k-2}{r-1} \right\rfloor.$$

4. The tightest known bound for $t \geq 2$, $r \geq 2$ that is derived from this approach appears in [14]. Here, the authors use:

$$b = \lceil n - nR_{\max} \rceil \tag{11.4}$$

$$e_b = n, \quad e_{j-1} = e_j - \left\lceil \frac{2e_j}{j} \right\rceil + r + 1, \forall j \in [2, b] \tag{11.5}$$

$$i = \max_{\{j \,:\, (e_j - j) \,<\, k,\ j \in [b]\}} j$$

where $R_{\max}$ is the maximum rate of a code with availability with parameters (r, t). One can substitute an upper bound for $R_{\max}$ in the event that the precise value of $R_{\max}$ is unknown. For example one can substitute the upper bound on rate given in Theorem 8 in the place of R_{max}.

Constructions achieving any of the known upper bounds on minimum distance for codes with availability are known only for the cases of very small rate [22], [222].

11.4 Strict Availability

Every linear availability code with parameter set (r, t) possesses a p-c matrix H with associated rowspace $\mathcal{H}$, having the property that associated with each coordinate $i \in [n]$, there is a collection $T_i \subseteq \mathcal{H}$ of row vectors of size $|T_i| = t$ where each vector in T_i has Hamming weight $\leq (r + 1)$ and where the intersection of support of any two rows has size $= 1$ and where each row vector has a non-zero value in the ith coordinate.

Linear codes with strict availability, are a further-constrained subset of linear availability codes as defined below. We note that both the product-code construction as well as Construction 2 by Wang *et al.*, are examples of linear codes with strict availability. Imposing the condition of strict availability makes it possible to derive tighter bounds on minimum distance and rate.

Definition 12. An $[n, k, d_{\min}]$ code $\mathcal{C}$ is said to be a code with strict availability with parameters (r, t) and denoted as an $(n, k, r, t)_{sa}$ code, if there exists an $(\frac{nt}{r+1} \times n)$ matrix H_{sa} having the following properties:

- Each row has Hamming weight $r+1$ and each column has Hamming weight t,
- The support of any two rows of H_{sa} intersect in at most one index and
- $\mathcal{C}$ lies in the nullspace of H_{sa}.

With respect to the definition above, we note that for strict availability to hold, we need that $(r+1) \mid nt$. We do not require the rows of the matrix H_{sa} to be linearly independent. It is straightforward to verify that a code with strict availability is also a code with availability. We will now derive the upper bound on the rate of codes with strict availability appearing in [14].

Theorem 10. [14] Let us define

$$\begin{aligned} S_{r,t} &= \{(k,n) : \text{an } (n,k,r,t)_{sa} \text{ code exists over a field } \mathbb{F}_q\}, \\ R(r,t) &= \sup_{\{(k,n) \in S_{r,t}\}} \frac{k}{n}. \end{aligned}$$

Then $R(r,t)$ satisfies the functional equation and upper bound given respectively by:

$$\begin{aligned} R(r,t) &= 1 - \frac{t}{r+1} + \frac{t}{r+1} R(t-1, r+1), \\ R(r,t) &\leq 1 - \frac{t}{r+1} + \frac{t}{r+1} \frac{1}{\prod_{j=1}^{r+1}(1 + \frac{1}{j(t-1)})}. \end{aligned} \tag{11.6}$$

Proof. Let a block length n be fixed. Let us define:

$$\begin{aligned} S_{n,r,t} &= \{k : \text{ an } (n,k,r,t)_{sa} \text{ code exists over some field } \mathbb{F}_q\}, \\ R(r,t,n) &= \max_{\{k \in S_{n,r,t}\}} \frac{k}{n}. \end{aligned}$$

It follows that $R(r,t) = \sup_n R(r,t,n)$. Next, we pick an integer n such that $R(r,t,n) > 0$. Note that by Construction 2, such an n does exist. Let $\mathcal{C}$ be an $(n,k,r,t)_{sa}$ code having rate $R(r,t,n)$. Since our interest is in deriving an upper bound on code rate, we can without loss of generality, assume that $\mathcal{C}$ has a p-c H_{sa} that satisfies the conditions laid

out in Definition 12. It follows that the rank of this p-c matrix satisfies $\text{rank}(H_{sa}) = n - nR(r,t,n)$. Next, we note that H_{sa}^T is the p-c matrix of a code with strict availability having parameters $r' = (t-1),\ t' = (r+1)$ and block length $n' = \frac{nt}{r+1}$. But H_{sa}^T may not define an availability code having maximum possible rate $R(t-1, r+1, n')$. It follows that

$$\begin{aligned}
n' - n'R(r',t',n') &= n' - n'R(t-1,r+1,n') \leq n - nR(r,t,n),\\
\Rightarrow nR(r,t,n) &\leq n - n' + n'R(t-1,r+1,n'),\\
\Rightarrow R(r,t,n) &\leq 1 - \frac{t}{r+1} + \frac{t}{r+1}R(t-1,r+1,n'),\\
\Rightarrow R(r,t,n) &\leq 1 - \frac{t}{r+1} + \frac{t}{r+1}\sup_{n'} R(t-1,r+1,n'),\\
\Rightarrow \sup_{n} R(r,t,n) &\leq 1 - \frac{t}{r+1} + \frac{t}{r+1}\sup_{n'} R(t-1,r+1,n'),\\
\Rightarrow R(r,t) &\leq 1 - \frac{t}{r+1} + \frac{t}{r+1}R(t-1,r+1).
\end{aligned}$$

Next, reversing the roles of H_{sa} and H_{sa}^T, meaning that this time, if H_{sa}^T were instead, the p-c matrix of an availability code having block length $n' = \frac{nt}{r+1}$ and rate $R(r',t',n')$, we would obtain the inequality in the reverse direction:

$$R(r,t) \geq 1 - \frac{t}{r+1} + \frac{t}{r+1}R(t-1,r+1).$$

This gives us the desired functional equation:

$$R(r,t) = 1 - \frac{t}{r+1} + \frac{t}{r+1}R(t-1,r+1).$$

Applying Theorem 8 to upper bound the rate $R(t-1,r+1)$, we obtain the upper bound appearing in (11.6). □

Notes

1. Upper bounds on the rate of binary codes with strict availability: In [114], [115], Kadhe and Calderbank provide the following upper bound on rate of binary codes with strict availability for $t = 3$ and any r:

$$R(r,3) \leq \frac{r-2}{r+1} + \frac{3}{r+1}H_2\left(\frac{1}{r+2}\right),$$

where $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$. For the specific case $t = 3$, any r and $n = \frac{(r+1)(2r+3)}{3}$, they derive the result:

$$R(r,3) \leq 1 - \frac{3}{r+1} + \frac{3\log(2r+4)}{(r+1)(2r+3)} \tag{11.7}$$

The bound in (11.7) can be achieved by constructing a code that makes use of the incidence matrix of a Steiner Triple System as the p-c matrix, a construction pointed out by [12], and [245] in the availability context and thus the upper bound in (11.7) is tight.

In the same paper [115], the authors provide the following upper bound on the rate of binary codes with strict availability for the case $r = 2$ and arbitrary t:

$$R(2,t) \leq H_2\left(\frac{1}{t+1}\right).$$

2. Codes with availability constructed from AG codes can be found in [19], [21], [93], [111]. The constructions in [19], [21], [93] make use of fiber products of curves. The construction in [111] is based on automorphisms of rational function fields.

3. Asymptotic lower bounds on rate as a function of relative minimum distance, with parameters (r,t) appear in [19], [133], [230].

4. High-rate binary constructions of availability codes: The construction given in [246] and described above in this section has rate $\frac{r}{r+t}$ for any (r,t) which is the highest rate among the known general constructions. For some sporadic parameters, codes with rate larger than $\frac{r}{r+t}$ are known:

   (a) The cyclic code construction given in [245] has the following parameters

   $$n = 2^t - 1, k = 2^{(t-1)} - 1, r = t-1$$

   and hence has rate $\frac{2^{(t-1)}-1}{2^t-1}$ which is larger than $\frac{r}{r+t} = \frac{t-1}{2t-1}$.

   (b) The constructions obtained by using the incidence matrix of a balanced incomplete block design (BIBD) as p-c matrix

pointed out in [12] and [245] in the context of availability codes have rate exceeding $\frac{r}{r+t}$. As an example for $t = 3$, the construction by using the incidence matrix of the Steiner triple system (STS), a special case of a BIBD, as p-c matrix has the following parameters

$$n = \frac{(2^s - 1)(2^s - 2)}{6}, k = n - (2^s - 1 - s), r = 2^{s-1} - 2,$$

and hence rate

$$\frac{k}{n} = 1 - \frac{6(2^s - 1 - s)}{(2^s - 1)(2^s - 2)} \geq \frac{r}{r+t} = 1 - \frac{3}{2^{s-1} + 1}$$

for $s \geq 2$. Note that the rate of this code based on STS achieves the upper bound on rate given in equation (11.7). The authors of [245] also generalize this construction to derive codes for other values of r for $t = 3$ by shortening.

(c) In [109], the authors point out a class of majority logic decodable codes as examples of codes with availability. Some of these codes have rate better than $\frac{r}{r+t}$.

5. Constructions with minimum distance $> (t + 1)$: In [222], the authors present constructions obtained by puncturing the generator matrix of the Simplex code in positions indicated by columns of the generator matrix of an anti-code which are optimal w.r.t. the alphabet-dependent bounds given in [30] and the Griesmer bound for linear codes for $r \in \{2, 3\}$. Construction of codes with minimum distance $> (t + 1)$ can also be found in [21], [22], [42], [93], [111], [153], [228], [245], [247], [262].

6. Lower bound on block length: In [12], the authors provide the following lower bound on block length of a code with strict availability,

$$n \geq (r + 1)^2 - \frac{r(r + 1)}{t},$$

and show that the above lower bound can be achieved with equality iff the p-c matrix of the code can be expressed as the incidence matrix of a BIBD.

7. Tight upper bounds on the minimum distance: A tight upper bound on the minimum distance of codes possessing the availability property only for information symbols and that also satisfy an additional restriction on the structure of the code can be found in [193]. An asymptotically tight upper bound on minimum distance of codes with availability under a certain restriction on the structure of the parity checks giving rise to the availability property, can be found in [8].

Open Problem 12. Determine the smallest possible block length n for which an $(n, M \geq q^k)$ availability code over an alphabet of size q exists, having availability parameters (r, t).

Open Problem 13. Determine the maximum rate of an availability code having parameters (r, t).

Open Problem 14. Derive a tight upper bound to the minimum distance $d_{\min}$ of an $(n, M \geq q^k)$ availability code over an alphabet of size q having parameters (r, t).

12

LRCs with Sequential Recovery

LRCs with sequential recovery are a subclass of linear LRCs introduced by Prakash *et al.* in [174]. This class of codes is designed to recover from a set of $t \geq 1$ erasures via a t-step sequential process. In each step, an additional erased code symbol is recovered as a function of r code symbols that have either not been erased, or else, have been recovered in a prior round. We will use the notation seq-LRC to denote an LRC with sequential recovery, more specifically, an (r, t) seq-LRC. In comparison with other classes of LRCs designed for recovery from multiple erasures, such as availability codes or (r, δ) codes, sequential recovery imposes the least restriction on the recovery process. Thus an (r, t) availability code is also an (r, t) seq-LRC and an (r, δ) all-symbol LRC is also a seq-LRC with parameters $(r, \delta - 1)$.

In this section, we begin by formally defining a seq-LRC. We then present a tight upper bound on the rate of an (r, t) seq-LRC. The bound is tight as there is a matching construction by Balaji *et al.* [13], that achieves the upper bound on code rate. We prove the rate bound in this section only for the cases $t = 2, t = 3$ and refer the reader to [13] for the proof in the general case. Similarly, we provide illustrative, example constructions for the parameter sets $(r = 6, t = 4)$ and $(r = 3, t = 5)$.

Definition 13. An $[n, k]$ linear code over $\mathbb{F}_q$ is said to be a seq-LRC with parameters (r, t) if any set $\{c_{i_1}, \ldots, c_{i_t}\}$ of t erased code symbols, can be recovered in a sequential manner, where the jth erased symbol c_{i_j}, $j = 1, 2, \cdots, t$, can be recovered using an equation of the form:

$$c_{i_j} = \sum_{\ell \in S_j} a_{\ell j} c_\ell, \text{ for some } a_{\ell j} \in \mathbb{F}_q,$$

where $S_j \subseteq [n] \setminus \{i_j, i_{j+1}, \cdots, i_t\}$ and $|S_j| \leq r$.

12.1 Recovery from Two or Three Erasures

We now present an upper bound due to Prakash *et al.* [174] on the rate $\frac{k}{n}$ of a seq-LRC for the case $t = 2$. The bound takes on the form of a lower bound on block length for given code dimension k.

Theorem 11. [174] Let $\mathcal{C}$ be an $[n, k]$ seq-LRC over $\mathbb{F}_q$ having parameters $(r, t = 2)$. Then

$$k + \left\lceil \frac{2k}{r} \right\rceil \quad \leq \quad n.$$

Proof. Let $\mathcal{B}$ be the vector space spanned by all codewords with Hamming weight $\leq (r + 1)$ in the dual $\mathcal{C}^\perp$ i.e.,

$$\mathcal{B} \quad = \quad < h_1, \ldots, h_\ell >$$

where each h_i is a codeword in the dual code $\mathcal{C}^\perp$ of Hamming weight $\leq (r+1)$. Without loss of generality, we assume that the set $\{h_1, \ldots, h_\ell\}$ is a linearly independent set of vectors over $\mathbb{F}_q$. Let us set

$$H_0 \quad = \quad \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_\ell \end{bmatrix}.$$

Note that the nullspace of H_0 is a vector space that contains $\mathcal{C}$, but could potentially be of larger dimension. Let w_1 be the number of columns in H_0 having Hamming weight 1. Counting the number of non-zero entries in H_0 in two different ways, row-wise and column-wise we obtain,

$$w_1 + 2(n - w_1) \leq \ell(r + 1). \tag{12.1}$$

Since $\mathcal{B}$ contains all codewords in $\mathcal{C}^{\perp}$ having weight $\leq (r+1)$, the dual of $\mathcal{B}$ is also an $(r, t=2)$ seq-LRC $\mathcal{C}_{seq}$, whose p-c matrix is precisely the matrix H_0. Since $\mathcal{C}_{seq}$ can recover from 2 erasures, it follows that $d_{\min}(\mathcal{C}_{seq}) \geq 3$. It follows that we cannot have two distinct columns of Hamming weight 1 in H_0 having the same support set of size 1. Hence we have that $w_1 \leq \ell$. Substituting this inequality in equation (12.1), we obtain

$$\begin{aligned} 2n - \ell &\leq \ell(r+1) \\ \implies \frac{2n}{r+2} &\leq \ell. \end{aligned} \tag{12.2}$$

Since $\mathcal{B} \subseteq \mathcal{C}^{\perp}$ and the dimension of $\mathcal{C}^{\perp}$ is $n-k$, we have that $\ell \leq n-k$. Substituting this in equation (12.2), we get

$$\begin{aligned} \frac{2n}{r+2} &\leq \ell \leq n-k \\ k &\leq \frac{nr}{r+2} \\ \frac{k(r+2)}{r} &\leq n. \end{aligned}$$

From the above equation, since n and k are integers,

$$k + \left\lceil \frac{2k}{r} \right\rceil \leq n.$$

□

The construction below due to [174] achieves the above lower bound on block length n.

Construction 3. [174] Let $2k = ur + b$, $1 \leq b \leq r$. Let $\mathcal{G}$ be a graph with $u+1$ nodes where u of the nodes have degree r and the remaining node has degree b. Let each edge represent a message symbol. Thus there are a total of k message symbols. Let each node represent a parity symbol storing the binary sum of message symbols corresponding to the edges incident on that node. Thus there are $(u+1)$ parity symbols. The message and parity symbols put together, yield an $[n = k+u+1, k]$ seq-LRC with parameters $(r, t=2)$. Furthermore, since $n = k+u+1 = k + \left\lceil \frac{2k}{r} \right\rceil$, by Theorem 11, this code has the minimum possible block length, and hence maximum possible rate for given parameter set $(k, r, t=2)$.

For any code based on the above construction, it is straightforward to graphically verify that the constructed code is a seq-LRC with parameters $(r, t = 2)$. We do this verification as follows. If two message symbols are erased, then at least one end of the edges corresponding to the erased message symbols is incident on a node that has only one message symbol erased. This allows us to sequentially recover both message symbols. We can similarly recover from the erasure of one parity symbol and one message symbol, or from the erasure of two parity symbols.

We now present the upper bound on the rate of a seq-LRC having parameters $(r, t = 3)$ presented by Song *et al.* in [226]. The derivation presented here, is taken from [13].

Theorem 12. [226] Let $\mathcal{C}$ be an $[n, k]$ seq-LRC over $\mathbb{F}_q$ having parameters $(r \geq 3, t = 3)$. Then,

$$\frac{k}{n} \leq \left(\frac{r}{r+1}\right)^2.$$

Proof. The following proof is based on [13]. As in the proof of Theorem 11, let $\mathcal{B}$ be the span of all codewords with Hamming weight $\leq (r+1)$ in the dual $\mathcal{C}^{\perp}$ i.e.,

$$\mathcal{B} = < h_1, \ldots, h_\ell >$$

where h_i is a codeword in $\mathcal{C}^{\perp}$ of Hamming weight $\leq (r+1)$. Without loss of generality, we assume that $\{h_1, \ldots, h_\ell\}$ form a linearly independent set of vectors. Let

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_\ell \end{bmatrix}.$$

Without loss of generality, we assume that H is of the form:

$$H = \begin{bmatrix} H_1 & H_2 & H_{\text{rest}} \end{bmatrix}.$$

where the submatrix H_1 contains the columns of H having Hamming weight 1, H_2 the columns of weight 2 and H_{rest} the remaining columns.

By permuting the rows of H and by permuting the columns within the matrices H_1 and H_2, the matrix H can be brought into the form:

$$H = \left[\begin{array}{c|c|c|} D_0 & A_1 & 0 \\ \hline 0 & D_1 & \\ \hline 0 & & C \end{array} \; H_{\text{rest}} \right], \tag{12.3}$$

where

1. D_0 is an $(a_0 \times a_0)$ diagonal matrix,
2. A_1 is an $(a_0 \times a_1)$ matrix with each row of weight $\leq r$ and column weight $\in \{1, 2\}$,
3. D_1 is an $(\rho_1 \times a_1)$ matrix with each column of weight ≤ 1,
4. The weight of each column of the matrix $\left[\frac{A_1}{D_1}\right]$ is exactly 2.
5. C is an $((\rho_1 + p) \times a_2)$ matrix with each column of weight exactly 2.

We now draw some conclusions relating to the submatrices of H. Each column in A_1 must have weight exactly equal to 1. If a column in A_1 had Hamming weight 2, this would imply that a column of D_1 had Hamming weight 0. But then it would be possible to find a set of three columns in H including the column in H corresponding to this column in D_1 that are linearly dependent, contradicting the fact that $d_{\min} \geq 4$. It follows that each column in D_1 also has Hamming weight 1. Counting the non-zero entries in matrix A_1 row-wise and column-wise, we obtain

$$a_1 \quad \leq \quad a_0 r. \tag{12.4}$$

Counting the non-zero entries in matrix $\left[\frac{D_1}{0} | C\right]$ row-wise and column-wise, we get

$$a_1 + 2a_2 \quad \leq \quad (\rho_1 + p)(r + 1). \tag{12.5}$$

By equating the number of rows in H to the sum of the number of rows in D_0 and C, we get

$$\ell \quad = \quad a_0 + \rho_1 + p. \tag{12.6}$$

Substituting equation (12.6) in (12.5), we get,

$$a_1 + 2a_2 \quad \leq \quad (\ell - a_0)(r+1). \tag{12.7}$$

Counting the non-zero entries in matrix D_1 row-wise and column-wise,

$$a_1 \quad \leq \quad \rho_1(r+1).$$

Substituting the above equation in equation (12.6),

$$\ell \;=\; a_0 + \rho_1 + p \quad \geq \quad a_0 + \frac{a_1}{(r+1)} + p.$$

Substituting equation (12.4) in the above equation,

$$\ell \quad \geq \quad \frac{a_1}{r} + \frac{a_1}{(r+1)} + p. \tag{12.8}$$

All the above inequalities hold even if any of D_0, A_1, D_1, C are empty matrices. Counting the non-zero entries in matrix H row-wise and column-wise,

$$a_0 + 2(a_1 + a_2) + 3(n - (a_0 + a_1 + a_2)) \quad \leq \quad \ell(r+1).$$

Substituting equation (12.7) in the above equation we have,

$$\begin{aligned} 3n - 2a_0 - \left(a_1 + \frac{(\ell - a_0)(r+1) - a_1}{2}\right) \quad &\leq \quad \ell(r+1) \\ 3n + a_0\left(\frac{(r+1)}{2} - 2\right) - \frac{a_1}{2} \quad &\leq \quad \frac{3\ell(r+1)}{2}. \end{aligned}$$

Since $r \geq 3$, we have $\left(\frac{(r+1)}{2} - 2\right) \geq 0$. Hence substituting equation (12.4) in the above equation we have,

$$\begin{aligned} 3n + \frac{a_1}{r}\left(\frac{(r+1)}{2} - 2\right) - \frac{a_1}{2} \quad &\leq \quad \frac{3\ell(r+1)}{2} \\ 3n - \frac{a_1}{r}\left(\frac{3}{2}\right) \quad &\leq \quad \frac{3\ell(r+1)}{2}. \end{aligned}$$

Substituting equation (12.8) in the above equation we have,

$$3n - \left(\frac{3\ell(r+1)}{2(2r+1)} - \frac{3p(r+1)}{2(2r+1)}\right) \quad \leq \quad \frac{3\ell(r+1)}{2}.$$

Using $p \geq 0$ in the above equation,

$$\begin{aligned} n &\leq \frac{\ell(r+1)}{2} + \frac{\ell(r+1)}{2(2r+1)} \\ \ell &\geq \frac{n(2r+1)}{(r+1)^2}. \end{aligned}$$

Using $\ell \leq n-k$ in the above equation, we get the bound given in the Theorem. □

We now present a code [226] achieving the above upper bound on rate.

Construction 4 (Product-Code Construction [226])**.** Let $n = (r+1)^2$. Arrange the n code symbols in the form of an $(r+1) \times (r+1)$ array. We impose the constraint that the sum of code symbols in any row or column equals 0. The code symbols in the first r rows and r columns can be chosen arbitrarily from $\mathbb{F}_q$. Given these r^2 values, the value of code symbols in the rest of the array is determined. Thus the code has dimension $k = r^2$. The fact that this is a seq-LRC with parameters $(r, t = 3)$ can be seen as follows. If there are 3 erased symbols then there must exist either a row or a column with exactly 1 erased code symbol which can be recovered from the remaining entries in that row or column. The remaining symbols can be similarly recovered thereafter. By Theorem 12, this code has the maximum possible rate $\left(\frac{r}{r+1}\right)^2$. This code is commonly known as the product code in two dimensions.

Although 2-dimensional product code is well known in the literature, [226] was the first paper to point out this construction in the context of seq-LRCs.

12.2 The General Case

An upper bound for the case of general (r, t) derived in Balaji *et al.* [13] is presented below in Theorem 13. Matching constructions establishing that this bound is tight are also given in the same paper. The bound also establishes the correctness of a conjecture due to Song *et al.* appearing in [226] and that is stated in the notes subsection.

12.2.1 Rate Bound

Theorem 13. [13] Let $\mathcal{C}$ be an $[n, k]$ seq-LRC over $\mathbb{F}_q$ having parameters $(r \geq 3, t)$, over the finite field $\mathbb{F}_q$. Then

$$\frac{k}{n} \leq \begin{cases} \frac{r^{s+1}}{r^{s+1}+2\sum_{i=0}^{s} r^i}, & \text{for } t \text{ even}, \\ \frac{r^{s+1}}{r^{s+1}+2\sum_{i=1}^{s} r^i+1}, & \text{for } t \text{ odd}, \end{cases} \tag{12.9}$$

where $s = \lfloor \frac{t-1}{2} \rfloor$.

The proof is along the same lines of the proof used to bound the rate for the cases $t = 2$ and $t = 3$. The bound is tight as it is possible to construct seq-LRCs that achieve this rate bound. Details including code constructions, can be found in [13].

12.2.2 Example Constructions

The seq-LRC construction provided in [13] that achieves the rate bound in (12.9), takes on a slightly different form for the cases t even and t odd. We present below, illustrative examples of code constructions corresponding to parameter sets $(r = 6, t = 4)$ representing the case of t even and $(r = 3, t = 5)$ representing the case t odd. Additional details can be found in [13].

Illustrative Example for t Even

Let $\mathcal{C}$ be a binary code with binary p-c matrix:

$$H = \left[\begin{array}{c|c|c} D_0 & A_1 & 0 \\ \hline 0 & D_1 & C \end{array}\right], \tag{12.10}$$

where

1. D_0 is an $(a_0 \times a_0)$ diagonal matrix with nonzero diagonal entries,
2. D_1 is an $(a_0 r \times a_0 r)$ diagonal matrix with nonzero diagonal entries,
3. A_1 is an $(a_0 \times a_0 r)$ matrix with each row of weight r and each column of weight 1,

4. C is an $(a_0 r \times \frac{a_0 r^2}{2})$ matrix with each row of weight r and each column of weight 2.

Hence the block length is given by $n = a_0(1+r+\frac{r^2}{2})$. Since the diagonal entries of D_0, D_1 are nonzero, it follows that the rank of H is equal to the number of rows. It follows that the dimension k of the code k equals $\frac{a_0 r^2}{2}$.

Let us form an augmented matrix H_∞ by adding a row to H at the very top, this row is the binary sum of rows in H. Thus H_∞ is given by:

$$H_\infty = \left[\begin{array}{c|c|c} \underline{1} & 0 & 0 \\ \hline D_0 & A_1 & 0 \\ \hline 0 & D_1 & C \end{array}\right], \tag{12.11}$$

where

1. $\underline{1}$ is an $(1 \times a_0)$ row vector with each coordinate equal to 1,
2. the matrices D_0, D_1, A_1, C remain as before.

Clearly, H_∞ is also a valid p-c matrix for the code $\mathcal{C}$. Each column of H_∞ has Hamming weight exactly 2. Hence this matrix H_∞ can be interpreted as the edge-vertex incidence matrix of a graph G_∞ having n edges and $(1 + a_0 + a_0 r)$ nodes (the number of rows in H_∞). Fig. 12.1, called the Moore graph, shows the graph G_∞ corresponding to the values $(a_0 = 7, r = 6)$ for a certain choice of matrices D_0, D_1, A_1, C ensuring that the girth of G_∞ is $\geq t + 1 = 5$.

Each edge in G_∞ represents a distinct code symbol while each vertex represents a parity-check on the code symbols represented by edges attached to the vertex. Thus each vertex is associated to a row in the p-c matrix H_∞ and each edge to a column of the p-c matrix. Each column of the p-c matrix H_∞ has Hamming weight 2 and the location of the two 1s within the column indicates the vertices to which the edge is connected. In Fig. 12.1, the edges at the very top, which are colored in red, correspond to the first $a_0 = 7$ columns of H_∞. The edges which are colored in black and blue, correspond respectively, to the columns of H_∞ corresponding to the sub-matrices

$$\begin{bmatrix} 0 \\ A_1 \\ D_1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ 0 \\ C \end{bmatrix}.$$

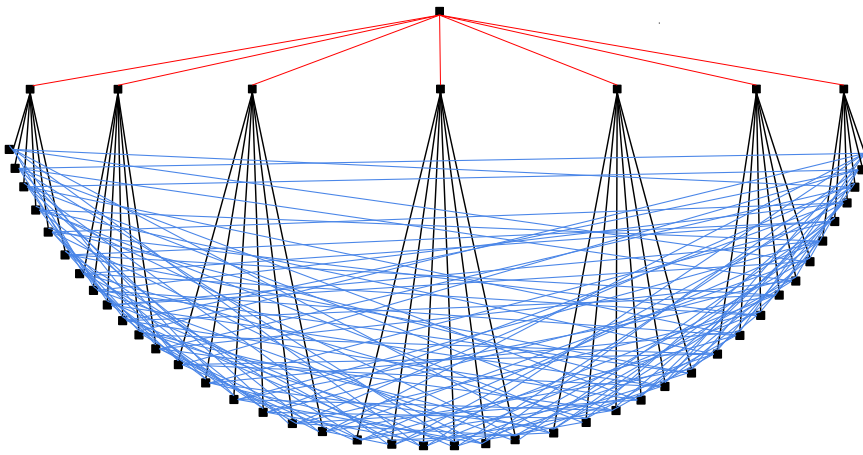

Figure 12.1: The figure shows the graphical interpretation of a binary, rate-optimal seq-LRC $\mathcal{C}$ having parameter set $(n, k, r, t) = (175, 126, 6, 4)$. Each of the 175 edges of the graph represents a distinct code symbol and each of the 50 vertices represents a parity-check of the code symbols represented by edges incident on it. This is a regular graph with a total of 50 vertices, each of degree $r + 1 = 7$ and is an example of a Moore graph called the Hoffman-Singleton graph. This graph has girth 5, which is a necessity for the associated binary code to be able to recover from $t = 4$ erasures. The code has redundancy 49 and not 50 since it turns out that the overall parity-check at the very top is redundant.

The sequential recovery property of this binary code can be deduced from the girth of G_∞ and the fact that all nodes in G_∞ have degree exactly $r + 1$. It turns out that the girth of our example G_∞ graph is equal to 5. Hence if there are any ≤ 4 erased symbols and if in G_∞ only edges corresponding to erased symbols are retained, there will be at least one vertex or parity-check with degree 1 and hence the erased symbols can be recovered one by one.

This connection between girth and sequential recovery, namely that a girth $\geq t + 1$ guarantees sequential recovery from t erasures, was to our knowledge, first pointed out in the context of LRCs by [192] and the graphs discussed in [192] can also be used to construct LRCs with sequential recovery.

It can be verified that the rate of the binary code $\mathcal{C}$ achieves the upper bound on rate given in Theorem 13 for $(r = 6, t = 4)$.

Illustrative Example for t Odd

We now present an illustrative example of a seq-LRC with $(r = 3, t = 5)$ Let $\mathcal{C}$ be a binary code having binary p-c matrix given by:

$$H = \left[\begin{array}{c|c|c} D_0 & A_1 & 0 \\ \hline 0 & D_1 & A_2 \\ \hline 0 & 0 & P \end{array}\right], \tag{12.12}$$

where

1. D_0 is an $(a_0 \times a_0)$ diagonal matrix with non-zero diagonal entries,
2. A_1 is an $(a_0 \times a_0 r)$ matrix with each row of weight r and each column of weight 1,
3. D_1 is an $(a_0 r \times a_0 r)$ diagonal matrix with non-zero diagonal entries,
4. A_2 is an $(a_0 r \times a_0 r^2)$ matrix with each row of weight r and each column of weight 1,
5. P is an $(\frac{a_0 r^2}{r+1} \times a_0 r^2)$ matrix with each row of weight $r+1$ and each column of weight 1.

Hence the block length is given by $n = a_0(1 + r + r^2)$. Since D_0, D_1 are diagonal and each column of A_1, A_2, P has Hamming weight exactly 1, we have that the rank of the above p-c matrix is equal to the number of rows. Hence $k = a_0 r^2 - \frac{a_0 r^2}{r+1}$.

As in the case t even, let us form an augmented matrix H_∞ by adding a row to H at the very top, this row is the binary sum of rows in H. Thus H_∞ is given by:

$$H_\infty = \left[\begin{array}{c|c|c} \underline{1} & 0 & 0 \\ \hline D_0 & A_1 & 0 \\ \hline 0 & D_1 & A_2 \\ \hline 0 & 0 & P \end{array}\right], \tag{12.13}$$

where

1. $\underline{1}$ is an $(1 \times a_0)$ vector with each coordinate equal to 1,

2. the matrices D_0, A_1, D_1, A_2, P remain as before.

Clearly, as in the case t even, H_∞ is also a valid p-c matrix for the code $\mathcal{C}$. Each column of H_∞ has Hamming weight exactly 2. Hence this matrix H_∞ can be interpreted as the edge-vertex incidence matrix of a graph with n edges and $(1 + a_0 + a_0 r + \frac{a_0 r^2}{r+1})$ nodes (number of rows in H_∞). The graph G_∞ with H_∞ as the node-edge incidence matrix is shown in Fig. 12.2, corresponding to the values $(a_0 = 4, r = 3)$ for a certain choice of the matrices $\{D_0, D_1, A_1, A_2, P\}$ such that girth of G_∞ is $\geq t + 1 = 6$.

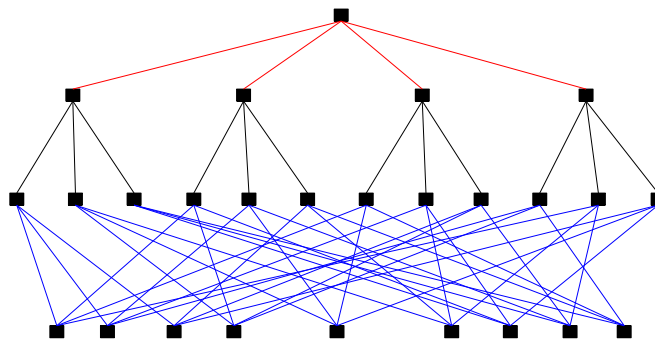

Figure 12.2: The figure shows the graphical interpretation of a binary, rate-optimal seq-LRC $\mathcal{C}$ having parameter set $(n, k, r, t) = (52, 27, 3, 5)$. Each of the 52 edges of the graph represents a distinct code symbol and each of the 26 vertices represents a parity-check on the code symbols represented by edges incident on it. This is a regular graph with a total of 26 vertices, each of degree $r + 1 = 4$ and is an example Moore graph for $(r = 3, t = 5)$ corresponding to the projective plane of order $r = 3$. This graph has girth 6, which is a necessity for the associated binary code to be able to recover from $t = 5$ erasures. The code has redundancy 25 and not 26 since it turns out that the overall parity-check at the very top is redundant.

As in the example case of $t = 4$ even above, each edge in G_∞ represents a distinct code symbol while each vertex represents a parity-check on the code symbols represented by edges attached to the vertex. Thus each vertex is associated to a row in the p-c matrix H_∞ and each edge to a column of the p-c matrix. Each column of the p-c matrix H_∞ has Hamming weight 2 and the location of the two 1s within the column indicates the vertices to which the edge is connected. In Fig. 12.2, the edges at the very top, which are colored in red, correspond to

the first a_0 columns of H_∞. The edges which are colored in black and blue, correspond respectively, to the columns of H_∞ corresponding to the sub-matrices

$$\begin{bmatrix} 0 \\ A_1 \\ D_1 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ 0 \\ A_2 \\ P \end{bmatrix}.$$

The sequential recovery property follows from by noting that the girth of G_∞ is ≥ 6 and that all nodes in G_∞ have degree exactly $r+1$. It can be seen that the rate of this code achieves the upper bound on rate given in Theorem 13 for $(r = 3, t = 5)$.

Notes

1. Conjecture on code rate by Song *et al.*: As discussed above, the tight upper bound on code rate appearing in Theorem 13 for general $t, r \geq 3$ establishes correctness of the conjecture below:

   Conjecture 1. [226] Let $\mathcal{C}$ be an $[n, k]$ seq-LRC having parameters $(r \geq 3, t)$, over the finite field $\mathbb{F}_q$. Then

   $$\frac{k}{n} \leq \frac{1}{1 + \sum_{i=1}^{m} \frac{a_i}{r^i}}$$

   where $m = \lceil \log_r(k) \rceil$, and the integers $\{a_i\}$ satisfy the conditions $a_i \geq 0$, $\sum_{i=1}^{m} a_i = t$.

2. High rate codes having smaller block length: The construction of seq-LRCs for any (r, t) having smaller block length and high rate, including some codes that are rate optimal, can be found in [12], [226]. Constructions of codes with smaller block length possessing both the sequential recovery and availability properties can be found in [252].

3. Upper bound on $d_{\min}$ for seq-LRCs: An upper bound on the minimum distance of an $[n, k]$ seq-LRC for $t = 2$ and any r can be found in [174]. Codes achieving this upper bound are also given

in [174] for $n = \frac{(r+\beta)(r+2)}{2}$, $\beta \mid r$. Constructions of seq-LRCs for general (r, t) with large minimum distance, but which are not necessarily minimum distance optimal, can be found in [192].

4. Source of upper bound on $d_{\min}$ for codes with availability: The upper bound on minimum distance for codes with availability obtained from equation (11.3) by applying (11.5) is based on the upper bound on minimum distance for seq-LRCs with $t = 2$ mentioned above and given in [174].

5. In Fig. 12.3, we compare the tight bound in (12.9) on the rate of a seq-LRC with the upper bound in (11.1), due to Tamo *et al.* on the rate of an availability code. The plots suggest that codes with sequential recovery offer a significant rate advantage. Availability codes, of course have the advantage of offering multiple disjoint repair sets.

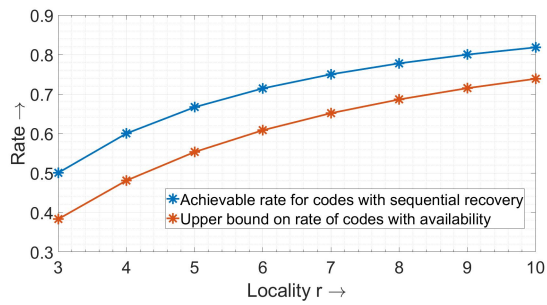

Figure 12.3: Comparison of rate bounds on codes with sequential recovery (12.9) and codes with availability (11.1) for $t = 12$.

Open Problem 15. The tight upper bound on rate presented in this section does not depend on block length n and depends only on the pair (r, t). It can be shown that the block length n of a rate-optimal code must necessarily satisfy $n \geq r^{\frac{t-2}{2}}$ on account of the tree-like structure forced upon the graphical representation of these codes as discussed in [13]. The open problem in this context, is to derive an upper bound on the dimension k of a seq-LRC for given (n, r, t) and identify constructions achieving the upper bound.

Open Problem 16. Clearly, the minimum distance $d_{\min}$ of a seq-LRC designed to recover from t erasures satisfies the lower bound $d_{\min} \geq (t+1)$. The open problem here is to derive an upper bound on the minimum distance of seq-LRCs for general (n, k, r, t) and obtain constructions achieving this upper bound.

13

Hierarchical Locality

Discussions of hierarchical codes can be found in the early papers by Huang *et al.* [102], [104] and Duminuco and Biersack [52], [53] and the notes section provides additional details on these papers. In a more recent paper by Sasidharan *et al.* [199], an upper bound on the minimum distance of hierarchical codes is derived and optimal constructions provided for certain parameter sets. In the present section, we present the bound on minimum distance appearing in [199], [201] as well as an example of a construction of a hierarchical code that is optimal with respect to the distance bound, also drawn from the same papers. We restrict our attention throughout, to the case of linear hierarchical codes.

Motivation To reduce the repair degree while maintaining a relatively low value of storage overhead, the Windows Azure Storage system employs an $[18, 14, 4]$ pyramid-like code with information-symbol locality, and locality parameters $(r = 7, \delta = 2)$. This code is illustrated in Fig. 13.1. Every code symbol except the global parities P_1, P_2 can be recovered accessing $r = 7$ other code symbols. While the code is well-suited to handle single node-failures, the failure for example, of two

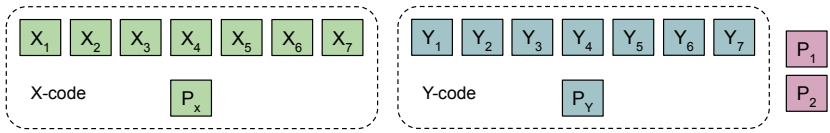

Figure 13.1: Illustrating the $[18, 14, 4]$ code with information-symbol locality and locality parameters $(r = 7, \delta = 2)$ employed in the Windows Azure storage system.

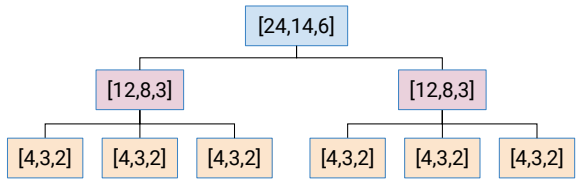

Figure 13.2: A $[24, 14, 6]$ code having 2-level hierarchical locality. The top-level $[24, 14, 6]$ code is a code that is a subcode of the disjoint union of two $[12, 8, 3]$ codes. The $[12, 8, 3]$ codes are in turn, subcodes of the disjoint union of three $[4, 3, 2]$ codes. As a result, the $[24, 14, 6]$ code can recover from any single-node failure with repair degree 3 by making use of the codes at the bottom level and any double-node failure with repair degree $r = 8$ by making use of the middle codes.

nodes within the support of the local X-code, will mean that local node repair is no longer possible and hence, the repair degree will jump from $r = 7$ to $k = 14$. In such a scenario, codes with hierarchical locality can step in to provide a more gradual degradation in repair degree.

An example of a code with hierarchical locality is presented in Fig. 13.2. The figure shows an $[24, 14, 6]$ code having 2-level hierarchical locality. The top-level $[24, 14, 6]$ code is a code that is a subcode of the disjoint union of two $[12, 8, 3]$ codes. The $[12, 8, 3]$ codes, called middle codes, are in turn, subcodes of the disjoint union of three $[4, 3, 2]$ codes. As a result, the $[24, 14, 6]$ code can recover from any single-node failure with repair degree 3 by making use of the codes at the bottom level and any double-node failure with repair degree $r = 8$ by making use of the middle codes. It is only with 3 or more node failures, that the repair degree jumps to 14. We note that the Windows Azure code has information-symbol locality, while the $[24, 14, 6]$ code has all-symbol locality.

The $[24, 14, 6]$ code is also an example of a code with two-level (all-symbol) hierarchical locality. Clearly, this has a natural extension to multi-level hierarchical locality, encompassing more than 2 layers. In this context, the conventional LRC may be regarded as codes having a single level of hierarchy. In the present section, we restrict our attention for simplicity, to codes with two-level, all-symbol, hierarchical locality and refer the reader to [199], [201] for an extension to the case of multi-level hierarchical locality.

We present below a formal definition of two-level, all-symbol hierarchical locality. When we speak in this section of a code with hierarchical locality, we will mean a linear code with all-symbol, two-level hierarchical locality.

Definition 14. [199] An $[n, k, d]$ linear code $\mathcal{C}$ is a code with hierarchical locality having locality parameters $[(r_1, \delta_1), (r_2, \delta_2)]$ if associated to each symbol $c_i, 1 \leq i \leq n$, there exists a code C_i obtained by puncturing $\mathcal{C}$ such that $c_i \in \text{Supp}(C_i)$ and the following conditions hold:

1. C_i has block length $\leq (r_1 + \delta_1 - 1)$,
2. $d_{\min}(C_i) \geq \delta_1$,
3. C_i is itself, a code with (r_2, δ_2)-locality.

As in the example, the code C_i associated to c_i will be referred to as the middle code associated to c_i. The local codes that are part of each middle code will simply be referred to as local codes.

Remark 12. Each local code (of a middle code), is a code with (r_2, δ_2)-locality, meaning that it is a code of block length $\leq (r_2 + \delta_2 - 1)$ and minimum distance $\geq (\delta_2 - 1)$. By the Singleton bound, this means that the dimension of each local code is $\leq r_2$. Similarly, the dimension of each middle code C_i is $\leq r_1$.

Remark 13. We can scale the definition recursively to h-level hierarchical locality by modifying the last constraint. In the case of h-level hierarchy, the code C_i would be required to be a code with $(h-1)$-level hierarchical locality having locality parameters $[(r_2, \delta_2), (r_3, \delta_3), \cdots, (r_h, \delta_h)]$.

13.1 An Upper Bound on $d_{\min}$

The upper bound on the minimum distance of a (linear) code with hierarchical locality derived by Sasidharan *et al.* in [199], [201] will now be presented. While the result given in is for the general, h-level hierarchy case, for simplicity, we present it here only for the case $h = 2$.

Theorem 14. Let $\mathcal{C}$ be an $[n, k, d]$-linear code with hierarchical locality having locality parameters $[(r_1, \delta_1), (r_2, \delta_2)]$. Then

$$d_{\min} \leq n - k + 1 - \left(\left\lceil \frac{k}{r_2} \right\rceil - 1\right)(\delta_2 - 1) - \left(\left\lceil \frac{k}{r_1} \right\rceil - 1\right)(\delta_1 - \delta_2). \quad (13.1)$$

Proof. The proof will identify a code $\mathcal{C}_S$ obtained by restricting the code $\mathcal{C}$ to a subset $S \subseteq [n]$, where S has large size and $\dim(\mathcal{C}_S) = (k - 1)$. The bound

$$d_{\min} \leq n - |S|, \quad (13.2)$$

then follows by invoking Lemma 9. We note that if G is a generator matrix for $\mathcal{C}$, then $\dim(\mathcal{C}_S)$ can alternately be described as $\text{rank}(G\,|_S)$. Algorithm 1 given below, is used to identify a candidate punctured code $\mathcal{C}_S$ having sufficiently large support $|S|$. We will use L_i and M_j to denote the supports of the local and middle codes respectively. We will assume an ordering of indices such that the algorithm below picks the L_i and M_j in order of their index. By this, we mean that in the jth iteration of the outer loop indexed by j, the algorithm identifies a middle code $\mathcal{C}_{M_j}$, that accumulates additional rank. Within the inner loop, indexed by the variable i, the algorithm picks up the local code $\mathcal{C}_{L_i}$, whose support L_i is contained in the support M_j of the middle code, i..e, $L_i \subseteq M_j$ and that accumulates additional rank. Thus, the inner loop index keeps incrementing without resetting, regardless of whether or not the outer loop index associated to the middle code, has been incremented. The algorithm terminates once the accumulated rank equals k.

Let i_{end} and j_{end} respectively denote the values of the indices i and j at which the algorithm terminates. We will use S to denote a running support set that is incremented whenever either index i or j is incremented. When we speak of the rank associated with support set S,

we will mean the rank of the matrix $G\,|_S$. Let a_i denote the incremental rank and s_i denote the incremental support size when adding the support of a local code L_i to the existing support set S by replacing S by the union $S \cup L_i$. Then we have $s_i \geq a_i + (\delta_2 - 1), \quad 1 \leq i \leq i_{\text{end}}$, since the rank condition (i.e., Line 3 in Algorithm 1) ensures that $a_i > 0$ in every iteration.

Let V_i denote the column space of the matrix $G|_{L_i}$. Let i_j denote the index of the last local code $\mathcal{C}_{L_{i_j}}$ added for fixed value j of the outer loop index. As noted above, the support L_{i_j} of the code $\mathcal{C}_{L_{i_j}}$ is contained in M_j. Within the jth outer iteration, if there are no more local codes having support contained in the support M_j of the current middle code, and which will result in an increase in rank of the associated matrix $G|_S$, then the support L_{i_j} of the last local code added with $L_{i_j} \subseteq M_j$ is deleted from S and the support set is incremented by taking instead, the union with M_j. Thus, in place of replacing the existing support set S by $S \cup L_{i_j}$, we replace S by $S \cup M_j$. This has the effect of increasing the support size during the i_jth inner iteration by an amount

$$\begin{aligned} s_{i_j} &\geq a_{i_j} + (\delta_1 - 1), \\ &= a_{i_j} + (\delta_2 - 1) + (\delta_1 - \delta_2), \quad 1 \leq j \leq j_{\text{end}}. \end{aligned}$$

Thus we have that for any j, $1 \leq j \leq j_{\text{end}}$, we have

$$s_i \geq \begin{cases} a_i + (\delta_2 - 1), & i \neq i_j, \\ a_{i_j} + (\delta_2 - 1) + (\delta_1 - \delta_2), & i = i_j. \end{cases}$$

The rank accumulates to k during iteration i_{end} of the inner loop, corresponding to value j_{end} of the outer loop's index. This can only happen if

$$i_{\text{end}} \geq \left\lceil \frac{k}{r_2} \right\rceil \quad \text{and} \quad j_{\text{end}} \geq \left\lceil \frac{k}{r_1} \right\rceil. \tag{13.3}$$

After adding $(i_{\text{end}} - 1)$ local codes, we would have arrived at a support set S having accumulated rank

$$\text{rank}(G\,|_S) = \sum_{i=1}^{i_{\text{end}}-1} a_i \leq (k-1).$$

Algorithm 1 (for the proof of Theorem 14)

Let $j = 0, i = 0, S = \phi$.
while ($\exists$ a middle code $\mathcal{C}_{M_j}$ having support $M_j \subseteq [n]$ such that $\text{rank}(G|_{S\cup M_j}) > \text{rank}(G|_S)$) **do**
    while ($\exists$ a local code $\mathcal{C}_{L_i}$ having support $L_i \subseteq M_j$ such that $\text{rank}(G|_{S\cup L_i}) > \text{rank}(G|_S)$) **do**
        $S = S \cup L_i$
        $i = i + 1$
    end while
    $S = (S \setminus L_{i-1}) \cup M_j$
    $j = j + 1$
end while

Clearly, we can augment S by adding a set J containing $(k-1) - \sum_{i=1}^{i_{\text{end}}-1} a_i$ indices to the support of S to ensure that the rank of $G\,|_{S\cup J}$ equals $(k-1)$. This will ensure that

$$|J| \;+\; \sum_{i=1}^{i_{\text{end}}-1} a_i \;=\; (k-1).$$

In a final step, we replace S by $S \cup J$. With this we have

$$\begin{aligned} |S| \;&\geq\; |J| + \sum_{i=1}^{i_{\text{end}}-1} s_i \\ &\geq\; |J| + \sum_{i=1}^{i_{\text{end}}-1} (a_i + \delta_2 - 1) \;+\; \sum_{j=1}^{j_{\text{end}}-1} (\delta_1 - \delta_2) \\ &=\; (k-1) \;+\; \sum_{i=1}^{i_{\text{end}}-1} (\delta_2 - 1) \;+\; \sum_{j=1}^{j_{\text{end}}-1} (\delta_1 - \delta_2). \end{aligned}$$

It follows from our estimates of i_{end} and j_{end} that

$$|S| \geq (k-1) \;+\; \left(\left\lceil \frac{k}{r_2} \right\rceil - 1\right)(\delta_2 - 1) + \left(\left\lceil \frac{k}{r_1} \right\rceil - 1\right)(\delta_1 - \delta_2),$$

leading to the bound on minimum distance appearing in the theorem. □

13.2 Optimal Constructions

Constructions for codes with hierarchical locality for any arbitrary level h of hierarchy can be found in [199], [201]. Our focus here is on the case of 2-level hierarchy. Let (n_1, n_2) denote block lengths of the middle and local codes respectively. Then the construction presented in [199] is optimal when $n_2 \mid n_1 \mid n$ and $r_2 \mid r_1 \mid k$. The construction is shown to be optimal under certain other numerical constraints as well. In [18], the authors provide optimal constructions based on algebraic curves and elliptic curves. The constructions provided in [18] also assume numerical conditions such as for example, $\delta_2 = 2$ or $r_2 \mid r_1 \mid k$. In [260], the authors first construct a family of generalized RS-based optimal LRCs and then use the resulting LRCs to construct optimal codes with hierarchical locality. The constructions presented in [260], are less restrictive in their choice of parameters in comparison with the constructions in [199] and [18]. We now present an illustrative example of the optimal construction in [199] for 2-level hierarchy.

Example Construction of an Optimal Code with 2-Level Hierarchy
The code presented here has parameters given by:

$$[n = 24,\ k = 14,\ (r_1, \delta_1) = (8, 3),\ (r_2, \delta_2) = (3, 2)].$$

As will be seen, the code will turn out to be optimal despite the fact that $r_2 \nmid r_1$. We choose $n_1 = 12$ and $n_2 = 4$ as the block lengths of the middle and local codes respectively and note that $n_2 \mid n_1 \mid n$. In the construction, there are two middle codes having disjoint support sets M_1 and M_2. In turn, each middle code contains three support-disjoint local codes, i.e., $M_i = L_{i1} \cup L_{i2} \cup L_{i3}, i = 1, 2$ where L_{ij} denotes the support of a local code.

The underlying finite field $\mathbb{F}_q$ in the construction is selected to be the field $\mathbb{F}_{25}$, which ensures that $n \mid (q-1)$. Let G, H with $H \subseteq G$ be subgroups of $\mathbb{F}_q^*$ with sizes given by $|G| = n_1 = 12$, $|H| = n_2 = 4$. Let α be a primitive element of $\mathbb{F}_q$, thus α has order 24. We set

$$G = \{1, \alpha^2, \alpha^4, \ldots, \alpha^{22}\}, \qquad H = \{1, \alpha^6, \alpha^{12}, \alpha^{18}\}.$$

We have the two coset decompositions:

$$\mathbb{F}_q^* = G \cup \alpha G \quad \text{and} \quad G = H \cup \alpha^2 H \cup \alpha^4 H.$$

The annihilator polynomials associated with each of these cosets are identified below:

$$P_{\alpha^j H}(x) \;=\; \prod_{\theta \in \alpha^j H} (x - \theta) \;=\; x^4 - \alpha^{4j}, \quad 0 \le j \le 5.$$

and

$$P_{\alpha^i G}(x) \;=\; \prod_{\theta \in \alpha^i G} (x - \theta) \;=\; x^{12} - \alpha^{12i}, \quad i = 0, 1.$$

For $j = 0, 1, 2$, let $f_j(x)$ and $g_j(x)$ denote the message polynomials of degree $(r_2 - 1) = 2$ associated to the local codes $C_{L_{1j}}$ and $C_{L_{2j}}$ respectively. This will ensure that each of the local codes obtained by evaluating these message polynomials has minimum distance $\delta_2 = 2$. The polynomials can thus be expressed in the form :

$$\begin{aligned} f_j(x) &= a_{j2}x^2 + a_{j1}x + a_{j0}, \\ g_j(x) &= b_{j2}x^2 + b_{j1}x + b_{j0}. \end{aligned}$$

Next, let $f(x), g(x)$ be polynomials satisfying:

$$\begin{aligned} f(x) &= f_j(x) \pmod{P_{\alpha^{2j} H}(x)}, \; 0 \le j \le 2, \\ g(x) &= g_j(x) \pmod{P_{\alpha^{2j+1} H}(x)}, \; 0 \le j \le 2. \end{aligned}$$

From Chinese Remainder Theorem theory, we have the following closed-from expressions for f and g:

$$f(x) = f_0(x)Q_{00}(x) + f_1(x)Q_{10}(x) + f_2(x)Q_{20}(x), \qquad (13.4)$$

$$g(x) = g_0(x)Q_{01}(x) + g_1(x)Q_{11}(x) + g_2(x)Q_{21}(x), \qquad (13.5)$$

where

$$Q_{00}(x) = \frac{(x^4 - \alpha^8)(x^4 - \alpha^{16})}{(1 - \alpha^8)(1 - \alpha^{16})}, \qquad Q_{01}(x) = \frac{(x^4 - \alpha^{12})(x^4 - \alpha^{20})}{(\alpha^4 - \alpha^{12})(\alpha^4 - \alpha^{16})},$$

$$Q_{10}(x) = \frac{(x^4 - 1)(x^4 - \alpha^{16})}{(\alpha^8 - 1)(\alpha^8 - \alpha^{16})}, \qquad Q_{11}(x) = \frac{(x^4 - \alpha^4)(x^4 - \alpha^{20})}{(\alpha^{12} - \alpha^4)(\alpha^{12} - \alpha^{20})},$$

$$Q_{20}(x) = \frac{(x^4 - 1)(x^4 - \alpha^8)}{(\alpha^{16} - 1)(\alpha^{16} - \alpha^8)}, \qquad Q_{21}(x) = \frac{(x^4 - \alpha^4)(x^4 - \alpha^{12})}{(\alpha^{20} - \alpha^4)(\alpha^{20} - \alpha^{12})}.$$

From (13.4), (13.5), it can be seen that the monomials in both $f(\cdot)$ and $g(\cdot)$ above have degree belonging to the set $\{0, 1, 2, 4, 5, 6, 8, 9, 10\}$. Since the middle code is required to have minimum distance $\delta_1 = 3$, we would like to make sure that the coefficient of x^{10} in both $f(x)$ and $g(x)$ equals zero. This can be ensured by making sure that the message coefficients $\{a_{ij}, b_{ij}, \mid i, j \in \{0, 1, 2\}\}$ are pre-coded to satisfy the conditions:

$$\begin{aligned} \frac{a_{02}}{(1-\alpha^8)(1-\alpha^{16})} + \frac{a_{12}}{(\alpha^8-1)(\alpha^8-\alpha^{16})} + \frac{a_{22}}{(\alpha^{16}-1)(\alpha^{16}-\alpha^8)} &= 0 \\ \frac{b_{02}}{(\alpha^4-\alpha^{12})(\alpha^4-\alpha^{20})} + \frac{b_{12}}{(\alpha^{12}-\alpha^{20})(\alpha^{12}-\alpha^4)} + & \\ \frac{b_{22}}{(\alpha^{20}-\alpha^4)(\alpha^{20}-\alpha^{12})} &= 0. \end{aligned}$$

We now proceed to identify an overall message polynomial $m(x)$ satisfying

$$m(x) = \begin{cases} f(x) & (\text{mod } P_G(x)), \\ g(x) & (\text{mod } P_{\alpha G}(x)). \end{cases}$$

Again by Chinese remainder theorem, we have that

$$m(x) = f(x)T_0(x) + g(x)T_1(x),$$

where

$$\begin{aligned} T_0(x) &= \frac{(x^{12}-\alpha^{12})}{(1-\alpha^{12})}, \\ T_1(x) &= \frac{(x^{12}-1)}{(\alpha^{12}-1)}. \end{aligned}$$

The monomials in the polynomial $m(x)$ have degrees belonging to the set $\{0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14, 16, 17, 18, 20, 21\}$. However, we are interested in constructing an overall block code having dimension $k = 14$. We have already imposed two constraints on the 18 message coefficients $\{a_{ij}, b_{ij}\}_{i,j=0}^{2}$. Thus we are in a position to impose two further constraints. In the interest of ensuring that the minimum distance as large a value as possible, we restrict $m(x)$ to have degree 18, by setting the

corresponding coefficients to zero, which turns out to correspond to imposing the precoding constraints

$$\left(\frac{1}{1-\alpha^{12}}\right)\sum_{j=0}^{2}\left[\frac{a_{j1}}{(\alpha^{8(j-1)}-\alpha^{8j})(\alpha^{8(j-1)}-\alpha^{8(j+1)})}-\frac{b_{j1}}{(\alpha^{8(j-1)+4}-\alpha^{8j+4})(\alpha^{8(j-1)+4}-\alpha^{8(j+1)+4})}\right] = 0$$

$$\left(\frac{1}{1-\alpha^{12}}\right)\sum_{j=0}^{2}\left[\frac{a_{j0}}{(\alpha^{8(j-1)+4}-\alpha^{8j+4})(\alpha^{8(j-1)+4}-\alpha^{8(j+1)+4})}-\frac{b_{j0}}{(\alpha^{8(j-1)+4}-\alpha^{8j+4})(\alpha^{8(j-1)+4}-\alpha^{8(j+1)+4})}\right] = 0.$$

In this way, we have ensured that the overall code has minimum distance ≥ 6. Turns out from the bound in (13.1) that this is the best possible.

Notes

1. Early work on hierarchical codes: The idea of hierarchical codes was introduced by Huang *et al.* [102], [104], with the help of an example of a $[20, 12]$ code having a two-level hierarchy and two global parities. The authors refer to the code as a multi-hierarchical extension of the Pyramid code, and note that this class of codes permits decoding at the lowest level of local codes, gradually moving up the hierarchy of local codes, and at the final step, making use of global parities. In the papers [52], [53], published subsequently, Duminuco and Biersack present hierarchical codes as a means of achieving reduced value of average repair degree. They investigate the average repair degree for a hierarchical code of block length 64, with probabilistically varying number of erased nodes.

Open Problem 17. Provide constructions of optimal codes having h-level hierarchical locality for all possible parameter sets $(n, k, (r_1, \delta_1), (r_2, \delta_2), \cdots, (r_h, \delta_h))$ without being restricted by numerical constraints involving code parameters.

14

Maximally Recoverable Codes

Background The notion of a maximally recoverable code (MRC) was introduced by Chen *et al.* [40]. Subsequent, early papers on the topic include those by Gopalan *et al.* [71], [72] and Blaum *et al.* [27]. There is considerable variation in the definition of an MRC within the literature. In this section, we introduce and treat MRCs in what is perhaps the most basic setting. More general settings are discussed in the notes subsection. In the basic setting, there is a parent $[n, k_L]$ code $\mathcal{C}_L$ over a field $\mathbb{F}_q$, and the goal is to identify a k-dimensional subcode $\mathcal{C}$ of $\mathcal{C}_L$ that is maximal in the following sense: $\mathcal{C}$ should be capable of recovering from all erasure patterns that it is possible to do so, given that $\mathcal{C}$ is a subcode of $\mathcal{C}_L$ of dimension k. It turns out that if the underlying field size q is large enough, such an MRC is guaranteed to exist.

More general definitions of an MRC do not assume a fully-specified parent code $\mathcal{C}_L$. They may assume for instance, that $\mathcal{C}_L$ is defined by an $(n - k_L \times n)$ p-c matrix of rank $(n - k_L)$ that is only partially identified. For instance, the location of the non-zero elements within the p-c matrix could be partially or fully specified, but not the entries themselves. A simple example of this is when the desired MRC $\mathcal{C}$ is required to have disjoint locality, corresponding to a specific partitioning

of the coordinate set $[n]$. From a geometric point of view, one could say that the topology of the parity-checks (by which we mean the support sets of the parity-checks) has been identified, but not the specific parity-checks themselves. The rest of the definition remains unchanged and an $[n, k]$ MRC is defined in this more general topological setting, as any subcode of $\mathcal{C}_L$ that is capable of recovering from any erasure pattern that it is possible for an $[n, k]$ subcode of such a code $\mathcal{C}_L$ to do so.

Motivation All of the discussion in this section will be restricted to linear codes. We begin with an informal description of an MRC. Let $\mathcal{C}_L$ be a linear $[n, k_L]$ code over a finite field $\mathbb{F}_q$, where each codeword satisfies locality constraints imposed by the linearly independent rows of an $((n - k_L) \times n)$ p-c matrix H_L. Consider a situation where one would like to impose additional parity constraints so as to arrive at a subcode $\mathcal{C}$ of $\mathcal{C}_L$ having dimension $k < k_L$, that is capable of recovering from a larger number of erasure patterns. How should one go about designing these additional p-c equations?

It turns out that for a given reduced code dimension k, there are certain excluded erasure patterns from which the subcode $\mathcal{C}$ cannot possibly recover, simply by virtue of being a subcode of the parent code $\mathcal{C}_L$. If q is sufficiently large, it is possible to identify a subcode $\mathcal{C}$ of $\mathcal{C}_L$ that is capable of recovering from every erasure pattern that is not an excluded erasure pattern. Such a subcode $\mathcal{C}$ is called an MRC with respect to the parent code $\mathcal{C}_L$. As noted at the start of this section, there are more general definitions of an MRC in the literature, and the notes subsection discusses some of these. A more formal definition of MRCs (for the basic setting), appears below.

14.1 Recoverable Erasure Patterns

Throughout this section, we will identify an erasure pattern with the corresponding subset $E \subseteq [n]$ that specifies the coordinates of erased code symbols. We will use S to denote the complement $S = [n] \setminus E$ of E.

Definition 15. A subset $E \subseteq [n]$ is called a recoverable erasure pattern of an $[n, k]$ code $\mathcal{C}$, if a codeword $\underline{c} \in \mathcal{C}$ is uniquely determined by its restriction $\underline{c}|_S$ to the complement $S = [n] \setminus E$ of E.

Theorem 15. Let $\mathcal{C}$ be an $[n, k]$ linear code and let G, H of size $(k \times n)$ and $(n - k \times n)$ respectively, be a generator and p-c matrix for $\mathcal{C}$. Let E be an erasure pattern and let $S = [n] \setminus E$ denote its complement. Let $G|_S$ and $H|_E$ denote the restrictions of G and H to the index sets S and E respectively. Then

(a) E is a recoverable erasure pattern iff $\text{rank}(G|_S) = k$,

(b) E is a recoverable erasure pattern iff $\text{rank}(H|_E) = |E|$.

Proof: Follows from the encoding and p-c equations given by

$$\begin{aligned} \underline{u}^T G &= \underline{c}^T, \\ H\underline{c} &= \underline{0}, \end{aligned}$$

where $\underline{u}, \underline{c}$ denote the message and code vector respectively. □

The set of recoverable erasure patterns of a block code $\mathcal{C}$ can be partially ordered by inclusion. Recoverable erasure patterns that are maximal with respect to this partial ordering, will be termed as maximal recoverable erasure patterns. Clearly, knowledge of the maximal recoverable erasure patterns from which $\mathcal{C}$ can recover, characterizes the set of all erasure patterns from which the code can recover. In the case of an $[n, k]$ linear block code over a finite field $\mathbb{F}_q$, by Theorem 15 above, recoverable erasure patterns are in 1-1 correspondence with subsets of the columns of the p-c matrix H that form linearly independent sets. Since the p-c matrix has rank $(n - k)$ it follows that all maximal recoverable erasure patterns are of size $(n - k)$.

14.1.1 Excluded Erasure Patterns

Let $\mathcal{C}$ be an $[n, k]$ subcode of an $[n, k_L]$ code $\mathcal{C}_L$, with $k < k_L$. By the discussion above, the subcode $\mathcal{C}$ cannot recover from any erasure pattern of size $> (n - k)$. There are certain other erasure patterns that $\mathcal{C}$ cannot possibly recover from, simply by virtue of being a subcode of the parent code $\mathcal{C}_L$. We will refer to these latter erasure patterns as excluded erasure patterns. In the theorem below, we characterize these

in two different ways, from the perspective of the generator and p-c matrices of the code $\mathcal{C}_L$.

Theorem 16 (Excluded Erasure Patterns)**.** Let $\mathcal{C}_L$ be a linear $[n, k_L]$ code over a finite field $\mathbb{F}_q$, having generator matrix G_L of size $(k_L \times n)$ and p-c matrix H_L of size $(n - k_L \times n)$. Let $\mathcal{C}$ be a subcode of $\mathcal{C}_L$ of dimension $k < k_L$. Then it is not possible for $\mathcal{C}$ to recover from an erasure pattern E of size $|E| \leq (n - k)$ if either of the following two equivalent conditions are satisfied:

(a) $\text{rank}(G_L|_S) < k$, where $S = [n] \setminus E$,

(b) $\text{rank}(H_L|_E) < |E| - (k_L - k)$.

Proof: We assume without loss of generality, that $E = \{1, 2, \cdots, |E|\}$, i.e., that the first $|E|$ code symbols have been erased. The proof of (a) is straightforward. To see that (b) is equivalent to (a), we begin by partitioning G_L as below:

$$G_L \quad = \quad [G_L|_E \; G_L|_S].$$

We will first assume (a) and show that (a) implies (b). Since $|S| = n - |E| \geq k$ and $\text{rank}(G_L|_S) < k$, it follows that we can replace the p-c matrix H_L by a row-reduced version $H_L^{'}$ that takes on the following form

$$\begin{aligned} H_L^{'} &= [H_L^{'}|_E \;\; H_L^{'}|_S] \\ &= \begin{bmatrix} A & B \\ [0] & D \end{bmatrix}, \end{aligned}$$

with $\text{rank}(D) = |S| - \text{rank}(G_L|_S) > |S| - k \geq 0$, and where the rows of A are linearly independent. It follows that

$$\begin{aligned} \text{rank}(H_L^{'}|_E) &= (n - k_L) - \text{rank}(D) \;<\; (n - k_L) - (|S| - k) \\ &= |E| - (k_L - k). \end{aligned}$$

One can reverse the arguments above to show that $\text{rank}(H_L^{'}|_E) < |E| - (k_L - k)$ implies $\text{rank}(G_L|_S) < k$. We thus obtain

$$\text{rank}(G_L|_S) < k \;\; \text{iff} \;\; \text{rank}(H_L|_E) < |E| - (k_L - k),$$

where we have used the fact that $\text{rank}(H_L|_E) = \text{rank}(H_L^{'}|_E)$. □

In the corollary below, we single out the case when an erasure pattern has maximal size $(n - k)$.

Corollary 3. In the setting of Theorem 16 above, it is not possible for $\mathcal{C}$ to recover from an erasure pattern E of size $|E| = (n-k)$ if either of the following two equivalent conditions are satisfied:

(a) $\text{rank}(G_L|_S) < k$, where $S = [n] \setminus E$,

(b) $\text{rank}(H_L|_E) < (n - k_L)$.

14.2 Defining Maximally Recoverable Codes

Motivated by the theorem and corollary above, we make the following definition:

Definition 16 (Maximally Recoverable Codes)**.** Let $\mathcal{C}_L$ be a linear $[n, k_L]$ code over a finite field $\mathbb{F}_q$, having generator matrix G_L of size $(k_L \times n)$ and p-c matrix H_L of size $(n - k_L \times n)$. Let $\mathcal{C}$ be a subcode of $\mathcal{C}_L$ of dimension $k < k_L$. We define an erasure pattern E to be an excluded erasure pattern (EEP) for $\mathcal{C}$ if E is such that the equivalent conditions (a) and (b) of Theorem 16 above are satisfied. If further, E is of size $|E| = (n-k)$, we will say that E is a maximal EEP (m-EEP). We will say that $\mathcal{C}$ is an MRC (with respect to parent code $\mathcal{C}_L$) iff $\mathcal{C}$ is able to recover from any erasure pattern of size $\leq (n-k)$ that is not an EEP.

Remark 14 (Observations on MRCs)**.** We make the following observations concerning an MRC:

1. An MRC is maximal with respect to the property of recoverability from erasure patterns.

2. As noted in the discussion above, knowledge of the maximal recoverable erasure patterns from which $\mathcal{C}$ can recover, characterizes the set of all erasure patterns from which the code can recover. It follows that we may also define an MRC as follows: $\mathcal{C}$ is an MRC (with respect to parent code $\mathcal{C}_L$) iff $\mathcal{C}$ is able to recover from any erasure pattern of size equal to $(n-k)$ that is not an m-EEP.

3. An MDS code can recover the entire codeword $\underline{c}$ given access to any restriction $\underline{c}|_S$ where S is of size k. Analogously, by Corollary 3 above, an MRC can recover the entire codeword $\underline{c}$ given access to

any restriction $\underline{c}|_S$ where S is of size k and where in addition, S is such that $\text{rank}(G_L|_S) = k$. In this sense, an $[n, k]$ MRC is "as MDS as possible" (given that it is a subcode of a parent code $\mathcal{C}_L$ having generator matrix G_L).

The corollary below presents a test for an MRC that follows from Theorem 15, Corollary 3, Definition 16 and Remark 14.

Corollary 4 (Test for MRCs)**.** In the setting of Theorem 16 above, an $[n, k]$ code $\mathcal{C}$ is an MRC with respect to $[n, k_L]$ code $\mathcal{C}_L$ iff $\mathcal{C}$ has the ability to recover from any erasure pattern E of size $|E| = (n - k)$ that satisfies either of the equivalent conditions:

(a) $\text{rank}(H_L|_E) = (n - k_L)$,

(b) $\text{rank}(G_L|_S) = k$, where $S = [n] \setminus E$.

Consequently, a subcode $\mathcal{C}$ of $\mathcal{C}_L$ is an MRC with respect to $\mathcal{C}_L$ iff $\mathcal{C}$ has a p-c matrix H satisfying:

$$\text{rank}(H|_E) \;=\; (n-k) \quad \text{whenever} \quad \text{rank}(H_L|_E) \;=\; (n-k_L).$$

The definition of an MRC does not make it clear whether or not MRCs exist for any given choice of $[n, k_L]$ parent code $\mathcal{C}_L$ and desired subcode dimension k. We answer this in the affirmative in this section. We begin with an example.

Example 14.1 (An Example MRC)**.** Let $\mathcal{C}_L$ be the parent $[n = 15, k_L = 12]$ code satisfying the locality constraints associated to p-c matrix

$$H_L \;=\; \left[\begin{array}{ccccc|ccccc|ccccc} 1&1&1&1&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&1&1&1&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1 \end{array}\right].$$

Let $k = 10$, so that it is desired to construct a $[15, 10]$ subcode $\mathcal{C}$ of $\mathcal{C}_L$ that is an MRC. Let $\{\theta_i\}_{i=1}^{15}$ be a set of 15 distinct elements in the finite field $\mathbb{F}_q$, partitioned into the three sets:

$$A_m \;=\; \{\theta_{5m+i} \mid i = 1, 2, 3, 4, 5\}, \quad m = 0, 1, 2.$$

Let the $\{\theta_i\}$ be further chosen so that

$$\theta_i + \theta_j \quad \neq \quad \theta_k + \theta_l,$$

for any two pairs $(\theta_i \in A_u, \theta_j \in A_u)$, $(\theta_k \in A_v, \theta_l \in A_v)$, with $u \neq v$. It is possible to identify such a set of 15 elements for example, in the finite field $\mathbb{F}_q$ with $q = 2^8$. It turns out that under these conditions, the subcode $\mathcal{C}$ defined by the augmented p-c matrix

$$H \quad = \quad \left[\begin{array}{ccccc|ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7 & \theta_8 & \theta_9 & \theta_{10} & \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} \\ \theta_1^2 & \theta_2^2 & \theta_3^2 & \theta_4^2 & \theta_5^2 & \theta_6^2 & \theta_7^2 & \theta_8^2 & \theta_9^2 & \theta_{10}^2 & \theta_{11}^2 & \theta_{12}^2 & \theta_{13}^2 & \theta_{14}^2 & \theta_{15}^2 \end{array}\right]$$

is an MRC. This can be viewed as an instance of constructing an MRC over a field extension, (see Remark 15 below), if the parent code $\mathcal{C}_L$, prior to field extension, is interpreted as a code over $\mathbb{F}_2$.

14.3 Existence of MRCs

Theorem 17. Let $\mathcal{C}_L$ be an $[n, k]$ code over $\mathbb{F}_q$. Let the field size q satisfy $q > \binom{n-1}{n-k-1}$. Then there exists an $[n, k]$ subcode code $\mathcal{C} \subseteq \mathcal{C}_L$ over $\mathbb{F}_q$ which is an MRC with respect to $\mathcal{C}_L$.

Proof. Let H be the p-c matrix of an $[n, k]$ code $\mathcal{C} \subseteq \mathcal{C}_L$ of the form:

$$H = \left[\begin{array}{c} H_L \\ H_{\text{Glob}} \end{array}\right],$$

where H_L is an $(n - k_L \times n)$ p-c matrix for $\mathcal{C}_L$ and where H_{Glob} is a $(k_L - k \times n)$ matrix. Let x_{ij} denote the entry in the ith row and jth column of H_{Glob}. Let E be an erasure pattern of size $(n - k)$. The restriction of H to E is of the form:

$$H|_E = \left[\begin{array}{c} H_L|_E \\ H_{\text{Glob}}|_E \end{array}\right].$$

Let the erasure pattern E also satisfy the property that $\text{rank}(H_L|_E) = (n - k_L)$. The theorem will then follow from Corollary 4 if we can establish that $\text{rank}(H|_E) = (n - k)$ for every such erasure pattern. For any given

fixed erasure pattern E of size $(n-k)$ satisfying $\text{rank}(H_L|_E) = (n-k_L)$, $H|_E$ can be made to have rank $(n-k)$ by suitably assigning values to the variables $\{x_{ij}\}$. This follows since clearly, there exists a choice of values for the variables $\{x_{ij}\}$ that ensures that the $(|E| \times |E|)$ matrix $H|_E$ is of full rank as one can always extend the linearly independent rows of $H_L|_E$ to a basis for $\mathbb{F}_q^{|E|}$. It follows that $H|_E$ is an $(|E| \times |E|)$ square matrix whose determinant $p_E(\{x_{ij}\}) = \det(H|_E)$ is a non-zero polynomial in the variables $\{x_{ij}\}$. Note that $p_E(\{x_{ij}\})$ is a polynomial that is of degree ≤ 1 in each of the variables $\{x_{ij}\}$.

Our aim is to identify an assignment of values to $\{x_{ij}\}$ such that the determinants $p_E(\{x_{ij}\})$ evaluate to a non-zero value for all $E \subseteq [n]$ of size $(n-k)$ such that $\text{rank}(H_L|_E) = n - k_L$. Towards this, we form the product polynomial:

$$P(\{x_{ij}\}) = \prod_{\{E:\ E\subseteq[n],\ |E|=(n-k),\ \text{rank}(H_L|_E)\ =\ n\ -\ k_L\}} p_E(\{x_{ij}\}).$$

Clearly, by our argument above, each of the constituent polynomials in this product is a nonzero polynomial. The next step is to identify a set of values to the variables $\{x_{ij}\}$ so that the product polynomial $P(\{x_{ij}\})$ is non-zero. We note that the product polynomial $P(\{x_{ij}\})$ is a polynomial in the variables $\{x_{ij}\}$ such that the degree of $P(\{x_{ij}\})$ in any of the individual variables is $\leq \binom{n-1}{n-k-1}$. This is because there are at most $\binom{n-1}{n-k-1}$ subsets E of size $|E| = (n-k)$ such that $H|_E$ contains a given variable x_{ij}. By the Combinatorial Nullstellensatz theorem [5], an assignment of values for $\{x_{ij}\}$ can always be found if $q > \binom{n-1}{n-k-1}$. □

Remark 15 (Necessity of Field Extension)**.** It can happen that the locality constraints represented by the p-c matrix H_L are over a field $\mathbb{F}_b$, so that $\mathcal{C}_L$ is an $[n, k_L]$ code over the field $\mathbb{F}_b$, but that a maximally-recoverable subcode having dimension k over the very same field $\mathbb{F}_b$ either does not exist or else, is hard to identify. It is possible in such situations, to pass on to an extension field $\mathbb{F}_q$, $q = b^m$ of $\mathbb{F}_b$ with $m > 1$, such that if one now replaces $\mathcal{C}_L$ with the code $\hat{\mathcal{C}}_L$ over the extension field $\mathbb{F}_q$ defined by the same p-c matrix H_L, a maximally-recoverable subcode $\mathcal{C}$ of $\hat{\mathcal{C}}_L$ over the field $\mathbb{F}_q$ can be found.

14.4 MRCs Constructed using Linearized Polynomials

We now present a method of explicitly identifying the variables $\{x_{ij}\}$ using linearized polynomials, in place of calling upon the Combinatorial Nullstellensatz. We illustrate for the case when the p-c matrix H_L and the code $\mathcal{C}_L$ are both defined over a finite field $\mathbb{F}_b$ having characteristic 2 and of size $b = 2^\ell$, but this approach extends to other characteristic as well. We claim that the $(k_L - k) \times n$ matrix H_{Glob} appearing in the proof of Theorem 17 can be selected to be of the form:

$$\begin{aligned} H_{\text{Glob}} &= L_p(\alpha_1, \ldots, \alpha_n, k_L - k) \\ &\triangleq \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^{2^\ell} & \alpha_2^{2^\ell} & \ldots & \alpha_n^{2^\ell} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{2^{\ell(k_L-k-1)}} & \alpha_2^{2^{\ell(k_L-k-1)}} & \ldots & \alpha_n^{2^{\ell(k_L-k-1)}} \end{bmatrix} \end{aligned} \tag{14.1}$$

where $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ are a set of n elements that are drawn from a degree-n field extension $\mathbb{F}_q$, $q = b^n$, of the field $\mathbb{F}_b$, that are linearly independent over $\mathbb{F}_b$. Thus, we are in need of field extension here as described in Remark 15 above. Thus under this construction technique, the required field size is exponential in the block length n.

We now proceed to explain the claim made above, namely that the submatrix H_{Glob} providing global parity, can be taken to have the form in (14.1). Let E be an erasure pattern of size $|E| = (n - k)$ such that $\text{rank}(H_L|_E) = (n - k_L)$. Without loss of generality, we assume that $E = \{1, 2, \ldots, (n - k)\}$. We have:

$$\begin{aligned} H_{\text{Glob}}|_E &= L_p(\alpha_1, \ldots, \alpha_{|E|},\ k_L - k) \\ &= \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_{|E|} \\ \alpha_1^{2^\ell} & \alpha_2^{2^\ell} & \ldots & \alpha_{|E|}^{2^\ell} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{2^{\ell(k_L-k-1)}} & \alpha_2^{2^{\ell(k_L-k-1)}} & \ldots & \alpha_{|E|}^{2^{\ell(k_L-k-1)}} \end{bmatrix} \end{aligned}$$

and after row reduction of $H_L|_E$, $H|_E$ can be written as,

$$
\begin{aligned}
H|_E &= \begin{bmatrix} H_L|_E \\ H_{\text{Glob}}|_E \end{bmatrix} \\
&= \left[\begin{array}{ccccccc}
1 & 0 & \dots & 0 & a_{1,1} & \dots & a_{1,|E|-m} \\
0 & 1 & \dots & 0 & a_{2,1} & \dots & a_{2,|E|-m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 1 & a_{m,1} & \dots & a_{m,|E|-m} \\
\hline
\alpha_1 & \alpha_2 & \dots & \alpha_m & \alpha_{m+1}^{2^\ell} & \dots & \alpha_{|E|} \\
\alpha_1^{2^\ell} & \alpha_2^{2^\ell} & \dots & \alpha_m^{2^\ell} & \alpha_{m+1}^{2^\ell} & \dots & \alpha_{|E|}^{2^\ell} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\alpha_1^{2^{\ell(k_L-k-1)}} & \alpha_2^{2^{\ell(k_L-k-1)}} & \dots & \alpha_m^{2^{\ell(k_L-k-1)}} & \alpha_{m+1}^{2^{\ell(k_L-k-1)}} & \dots & \alpha_{|E|}^{2^{\ell(k_L-k-1)}}
\end{array}\right]
\end{aligned}
$$

where $a_{i,j} \in \mathbb{F}_{2^\ell}$ and where we have set $m = (n - k_L)$. If we prove that $H|_E$ is full rank then the above choice of H_{Glob} yields an MRC from Corollary 4.

Suppose $H|_E$ has rank $< |E|$, then there is a non-zero vector $v = [v_1, ..., v_{|E|}]$ in the left null space of $H|_E$. Hence $vH|_E = 0$ and this implies that for $1 \leq i \leq (n - k_L)$,

$$v_i = \sum_{j=0}^{k_L-k-1} v_{j+m+1}\alpha_i^{2^{\ell j}} = f(\alpha_i)$$

where we have defined $f(\cdot)$ to be the linearized polynomial given by

$$f(x) = \sum_{j=0}^{k_L-k-1} v_{j+m+1}x^{2^{\ell j}}.$$

Then for $n - k_L + 1 \leq i \leq n - k$,

$$f(\alpha_i) = \sum_{j=1}^{n-k_L} a_{ji}v_j = \sum_{j=1}^{n-k_L} a_{ji}f(\alpha_j).$$

Since $a_{ji} \in \mathbb{F}_{2^\ell}$, by the properties of linearized polynomials we have,

$$f\left(\alpha_i + \sum_{j=1}^{n-k_L} a_{ji}\alpha_j\right) = 0, \quad n - k_L + 1 \leq i \leq n - k.$$

Since $\alpha_1, \dots, \alpha_{|E|}$ are linearly independent over $\mathbb{F}_{2^\ell}$, this implies that $f(x)$ has at least $2^{\ell(k_L-k)}$ zeros, since any linear combination of the

set $\{\alpha_i + \sum_{j=1}^{n-k_L} a_{ji}\alpha_j : n - k_L + 1 \leq i \leq (n-k)\}$ of $(k_L - k)$ linearly independent elements over $\mathbb{F}_{2^\ell}$ is also a zero of $f(x)$. There are $2^{\ell(k-k_L)}$ such linear combinations. However, the degree of $f(x)$ is $\leq 2^{\ell(k-k_L-1)}$. It follows that $f(x) \equiv 0$ i.e., $f(x)$ must be the all-zero polynomial. This implies that the vector $v = 0$, a contradiction. Hence $H|_E$ is full rank and this choice of H_{Glob} yields an MRC.

The above described construction has large field size primarily because of the choice of H_{Glob} that contains n elements $\{\alpha_1, \ldots, \alpha_n\}$ that are linearly independent over $\mathbb{F}_{2^\ell}$. It turns out that for the same choice of H_{Glob}, it is possible to reduce the field size for a special case of H_L by selecting the set $\{\alpha_1, \ldots, \alpha_n\}$ more intelligently. This is described below.

14.5 Reduced Field-Size Construction for the Disjoint Locality Case

We present in this subsection, a construction due to Gopalan *et al.* [71], and along the lines of the linearized polynomial construction appearing in the previous subsection, for a specific case of all-symbol locality. The reduced field size comes about through an intelligent choice of the $\{\alpha_i\}_{i=1}^n$, lying in an extension field $\mathbb{F}_q$ of the ground field $\mathbb{F}_b$ of suitable size, that does not require all of them to be linearly independent. We retain the notation of the previous subsection.

We assume in this construction that $(r+1)|n$. More formally, let $\mathcal{C}_L$ be an $[n, k_L]$, $(r, \delta = 2)$ LRC with all-symbol locality, having disjoint repair sets and p-c matrix given by:

$$H_L = \begin{bmatrix} 1_{r+1} & 0 & \ldots & 0 \\ 0 & 1_{r+1} & \ldots & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & \ldots & 1_{r+1} \end{bmatrix}, \tag{14.2}$$

where 1_{r+1} is a row vector of length $r+1$ with all components equal to 1. Note that the matrix H_L is a binary matrix, so that in reference to the previous subsection, we have that the ground field here is $\mathbb{F}_b = \mathbb{F}_2$.

Let the matrix H_{Glob} identifying global parity-checks be given by $H_{\text{Glob}} = L_p(\alpha_1, \ldots, \alpha_n, k_L - k)$. Set

$$\begin{aligned} m &= \frac{n}{r+1} = (n - k_L), \\ S_g &= \{(r+1)(g-1)+1, \ldots, (r+1)g\}, \quad \text{for } \ 1 \le g \le m. \end{aligned}$$

It follows that $(n-k) = m + (k_L - k)$. From Corollary 4, it follows that in determining whether or not a code is an MRC, it suffices to focus attention on those erasure patterns of size $(n-k)$ that are such that $\text{rank}(H_L|_E) = n - k_L = m$.

From equation (14.2), it can be seen that

$$\text{rank}(H_L|_E) \;=\; |\{\ j : |S_j \cap E| > 0,\ j \in [m]\ \}|,$$

for any $E \subseteq [n]$. It follows that $\text{rank}(H_L|_E) = n - k_L = m$ iff we choose the erasure pattern E such that there is at least one erasure within the support of each local code. Accordingly, we restrict our attention from here on to erasure patterns of size $(n-k)$ that have at least one element in the support S_g of each local code.

We now show how it is possible to select the $\{\alpha_i\}$ in such a way that $H|_E$ is invertible for any such E, i.e., has rank $(n-k)$. This will ensure that the code having H as p-c matrix is an MRC. Hence without loss of generality, we can assume that the erasure pattern E has the structure shown below for some pairs $\{(i_1, j_1),\ \ldots,\ (i_m, j_m)\}$ satisfying

$$\begin{gathered} [i_g, j_g] \ \subseteq\ S_g,\ g \in [m], \\ i_1 \le j_1,\ i_2 \le j_2,\ \cdots,\ i_m \le j_m, \\ E \ = \ [i_1, j_1] \ \cup \ \cdots \ \cup [i_m, j_m], \\ (j_1 - i_1 + 1) \ + \ \cdots \ + \ (j_m - i_m + 1) = |E| = m + k_L - k. \end{gathered}$$

The restriction $H|_E$ of H to E takes on the form:

$$H|_E = \begin{bmatrix} H_L|_E \\ H_{\text{Glob}}|_E \end{bmatrix}$$

$$= \left[\begin{array}{ccccccccccc} 1 & \dots & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 1 & \dots & 1 \\ \hline & & & & & & & & & \\ \alpha_{i_1} & \dots & \alpha_{j_1} & \alpha_{i_2} & \dots & \alpha_{j_2} & \dots & \alpha_{i_m} & \dots & \alpha_{j_m} \\ \alpha_{i_1}^2 & \dots & \alpha_{j_1}^2 & \alpha_{i_2}^2 & \dots & \alpha_{j_2}^2 & \dots & \alpha_{i_m}^2 & \dots & \alpha_{j_m}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{i_1}^{2^{(k_L-k-1)}} & \dots & \alpha_{j_1}^{2^{(k_L-k-1)}} & \alpha_{i_2}^{2^{(k_L-k-1)}} & \dots & \alpha_{j_2}^{2^{(k_L-k-1)}} & \dots & \alpha_{i_m}^{2^{(k_L-k-1)}} & \dots & \alpha_{j_m}^{2^{(k_L-k-1)}} \end{array}\right].$$

Note that for some values of $u \in [m]$, it could be that $i_u = j_u$. It follows that for some $0 < \ell \leq m$, we can without loss of generality, write:

$$\begin{aligned} j_u - i_u &> 0, \forall u \in [\ell], \\ j_u - i_u &= 0, \forall u \in [m] \setminus [\ell], \\ \sum_{u=1}^{\ell} (j_u - i_u) &= k_L - k. \end{aligned}$$

The last equation implies that $\ell \leq k_L - k$, a fact we will make use of below. Next, we perform some column operations on $H|_E$. We add the column i_u to each of the columns $(i_u + 1, \dots, j_u)$, for $u = 1, \dots, \ell$. Clearly this column operation does not change the rank of $H|_E$. The resulting matrix after column operations is given by:

$$H'_E = \begin{bmatrix} H'_{\text{L},E} \\ H'_{\text{glob},E} \end{bmatrix}$$

where

$$\begin{aligned} H'_{\text{glob},E} = \; & L_p(\alpha_{i_1}, (\alpha_{i_1} + \alpha_{i_1+1}), \dots, (\alpha_{i_1} + \alpha_{j_1}), \\ & \alpha_{i_2}, (\alpha_{i_2} + \alpha_{i_2+1}), \dots, (\alpha_{i_2} + \alpha_{j_2}), \\ & \dots, \alpha_{i_\ell}, (\alpha_{i_\ell} + \alpha_{i_\ell+1}), \dots, (\alpha_{i_\ell} + \alpha_{j_\ell}), \\ & \alpha_{i_{\ell+1}}, \alpha_{i_{\ell+2}}, \dots, \alpha_{i_m}, k_L - k) \end{aligned}$$

and

$$H'_{\mathrm{L,E}} = \begin{bmatrix}
1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 & & & & \\
0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & I_{m-\ell} & \\
0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 & & & &
\end{bmatrix}.$$

Clearly, the rank of above matrix H'_E is $m + \mathrm{rank}(J)$ where, J is the square $((k_L - k) \times (k_L - k))$ matrix given by

$$\begin{aligned} J &= L_p((\alpha_{i_1} + \alpha_{i_1+1}), \dots, (\alpha_{i_1} + \alpha_{j_1}), (\alpha_{i_2} + \alpha_{i_2+1}), \dots, (\alpha_{i_2} + \alpha_{j_2}), \\ &\quad \dots, (\alpha_{i_\ell} + \alpha_{i_\ell+1}), \dots, (\alpha_{i_\ell} + \alpha_{j_\ell}), \quad k_L - k). \end{aligned}$$

For H'_E to be of full rank, we need that J have full rank $k_L - k$. Let

$$\begin{aligned} T &\triangleq \{(\alpha_{i_1} + \alpha_{i_1+1}), \dots, (\alpha_{i_1} + \alpha_{j_1}), (\alpha_{i_2} + \alpha_{i_2+1}), \dots, (\alpha_{i_2} + \alpha_{j_2}), \\ &\quad \dots, (\alpha_{i_\ell} + \alpha_{i_\ell+1}), \dots, (\alpha_{i_\ell} + \alpha_{j_\ell})\}. \end{aligned}$$

It follows from the definition in (14.1) that it suffices to select the $\{\alpha_i\}_{i=1}^n$ in such a way that the $(k_L - k)$ elements in T are linearly independent over $\mathbb{F}_2$. To achieve this, we define for $1 \le i \le m$,

$$\begin{aligned} \{\alpha_{(r+1)(i-1)+1}, \dots, \alpha_{(r+1)i}\} &= \lambda_i \{\zeta_1, \dots, \zeta_{r+1}\} \\ &:= \{\lambda_i \zeta_1, \ \dots, \ \lambda_i \zeta_{r+1}\}, \end{aligned}$$

where the scale factors

$$\{\lambda_1, \lambda_2, \dots, \lambda_m\} \subseteq \mathbb{F}_q$$

are chosen to be a set of m elements from $\mathbb{F}_q$ such that any subset of size $(k_L - k)$ is a linearly independent set over $\mathbb{F}_{2^r}$, where $\mathbb{F}_{2^r}$ is a subfield of the code symbol alphabet $\mathbb{F}_q$ and where further,

$$\{\zeta_1, \dots, \zeta_r\} \subseteq \mathbb{F}_{2^r}$$

are a set of linearly independent elements over $\mathbb{F}_2$ and where we set $\zeta_{r+1} = 0$. Thus we must have that r divides the degree $[\mathbb{F}_q : \mathbb{F}_2]$ of

the extension $\mathbb{F}_q/\mathbb{F}_2$, i.e., $r \mid [\mathbb{F}_q : \mathbb{F}_2]$. The precise value of q will be indicated at a later point in the proof. We will now show that the elements in T are linearly independent over $\mathbb{F}_2$. Let us assume to the contrary, that the elements in T are linearly dependent. This would imply that for some choice of coefficients $\{b_{us} \in \mathbb{F}_2\}$ with at least one of them being non-zero, we have

$$\sum_{u=1}^{\ell} \sum_{s=1}^{j_u - i_u} b_{us}(\alpha_{i_u} + \alpha_{i_u+s}) = \sum_{u=1}^{\ell} \lambda_u \sum_{s=1}^{j_u - i_u} b_{us}(\zeta_1 + \zeta_{s+1}) = 0.$$

Since

$$\sum_{s=1}^{j_u - i_u} b_{us}(\zeta_1 + \zeta_{s+1}) \in \mathbb{F}_{2^r}$$

and $\lambda_1, \ldots, \lambda_\ell$ are linear independent over $\mathbb{F}_{2^r}$ as $0 < \ell \leq (k_L - k)$, it must be that:

$$\sum_{s=1}^{j_u - i_u} b_{us}(\zeta_1 + \zeta_{s+1}) \quad = \quad 0,$$

for all u. However, since $0 < (j_u - i_u) \leq r$ for $u \in [\ell]$, and $\zeta_1, \ldots, \zeta_r$ are linearly independent over $\mathbb{F}_2$, this is possible only if $b_{us} = 0$, for all $1 \leq u \leq \ell$ and $1 \leq s \leq j_u - i_u$, which contradicts the linear-dependence assumption. Hence for the choice of

$$\{\alpha_{(r+1)(i-1)+1}, \ldots, \alpha_{(r+1)i}\} = \lambda_i \{\zeta_1, \ldots, \zeta_{r+1}\},$$

we have that the code constructed is maximally recoverable. We now present an explicit choice for λ_i. We choose the $\{\lambda_i\}$ to be drawn from a finite field of size

$$q \quad = \quad 2^{g(k_L - k)},$$

where the integer g is chosen to satisfy $r \mid g$ and $m \leq 2^g$. We present the choice of $\{\lambda_i\}$ in the form of a vector with $(k_L - k)$ components over the field $\mathbb{F}_{2^g}$:

$$\lambda_i \quad := \quad [1, \beta_i, \beta_i^2, , \ldots, , \beta_i^{(k_L - k)-1}]^T$$

where $\beta_1, \beta_2, \ldots, \beta_m$ are m distinct elements from $\mathbb{F}_{2^g}$. Now $\{\zeta_1, \ldots, \zeta_r\}$ are any set of r elements in $\mathbb{F}_{2^r} \subseteq \mathbb{F}_{2^g}$ which are linearly independent over $\mathbb{F}_2$. Hence the required field size for this choice of α_i's is $q = 2^{g(k_L - k)}$.

Notes

1. MRC constructions for the case of uniform, disjoint locality: the discussion here, pertains to a more general definition of an MRC, corresponding to the setting where the parent code is only partially specified as described below.

   Let $(r+\delta-1) \mid n$ and let $\mathcal{C}_L$ be an $[n, k_L]$, (r, δ) LRC over the finite field $\mathbb{F}_q$ having all-symbol locality, in which the n code symbols are divided into disjoint groups containing $(r+\delta-1)$ code symbols each. The locality constraint in this case, is that each local code must be an $[r+\delta-1, r]$ MDS code. There is freedom however, in selecting the particular MDS code employed. The goal as in the basic setting, is to construct an $[n, k]$ subcode $\mathcal{C}$ of $\mathcal{C}_L$ that is an MRC, i.e., a subcode $\mathcal{C}$ that can recover from any erasure pattern that it is possible to do so under the given constraints.

   Such codes were introduced in [27] under the name of partial-MDS (P-MDS) codes. Field extensions are permitted meaning that it is allowed to identify a subcode $\mathcal{C}$ over an extension field $\mathbb{F}_{q^e}$ of $\mathbb{F}_q$, see Remark 15. An important goal in this thread of research, is to identify the smallest size of extension field $\mathbb{F}_{q^e}$ for which it is possible to find an MRC. Some papers in the literature, assume a specific p-c matrix for the parent code $\mathcal{C}_L$ in which case, the setting reverts to the basic setting addressed in this section.

   Reduced field-size constructions of MRCs for general (r, δ) can be found in [35], [36], [70], [78], [83], [158], [159], [232], apart from the existential result appearing in Theorem 17. These constructions yield upper bounds on the required field size. Lower bounds on the field size required to construct an MRC in this setting, are presented in [79], [80]. The explicit construction described in Section 14.5 based on [71], corresponds to the case $\delta = 2$. Apart from the constructions in [35], [36], [70], [78], [83], [158], [159], [232], additional constructions of MRCs, including constructions for the case $\delta = 2$, can be found in [20], [27], [96], [147]. Construction of MRCs corresponding to specific values of parameters r, h, where $h := (k_L - k)$ represents the number of global parities, can be

found in [15], [24], [28], [29], [39]. Weight enumerators, GHWs and higher support weights for MRCs with $\delta = 2$ can be found in [135].

2. MRCs for the case when the locality constraints correspond to a grid-like topology: In [73], the authors initiated the study of MRCs under a grid-like topology. Under the grid-like topology framework, each codeword of a code $\mathcal{C}$ over a field $\mathbb{F}$ of characteristic 2 is expressed in the form of an $(m \times n)$ matrix. Each row satisfies parity-checks imposed by an $(a \times n)$ p-c matrix H_a; each column satisfies parity-checks imposed by a second $(b \times m)$ p-c matrix H_b and a third $(h \times mn)$ matrix H_{gl} imposes global parities. An erasure pattern E is said to be recoverable, if there is a code (i.e., a set of matrices H_a, H_b, H_{gl}, all with entries drawn from $\mathbb{F}$) which is capable of recovering from E. An MRC is then a code that can recover from all possible recoverable erasure patterns. In [73], the authors derive a super-polynomial lower bound on the size of the finite field $\mathbb{F}$ required to construct an MRC with respect to the grid-like topology. They also derive a necessary and sufficient condition, termed as the regularity condition, for an erasure pattern to be recoverable for the case when $(a = 1, h = 0)$ and arbitrary b. In [217], the authors extend the study to consider $a \in \{1, 2\}, h = 0$, and arbitrary b, and characterize a subset of recoverable erasure patterns using an alternate proof technique.

There are many open problems on the topic of MRCs that one could list. A basic open problem is listed below.

Open Problem 18. Determine the minimum field size required to construct an MRC with respect to parent code $\mathcal{C}_L$ having disjoint-locality specified by the p-c matrix H_L given in (14.2).

More generally, one can raise the same question as above with the single-parity-check local codes associated to (14.2) replaced by MDS codes constructed say, using Vandermonde matrices. One could generalize this further by leaving the p-c matrices of the individual MDS codes unspecified. One could also ask similar questions with respect to other topologies.

15

Codes with Combined Locality and Regeneration

In the previous sections, we have seen that RGCs minimize the repair bandwidth, whereas LRCs have low repair degree. A natural question to ask is, do there exist codes that simultaneously have low repair bandwidth as well as low repair degree? Working independently, Kamath *et al.* [117], [134] and Rawat *et al.* [189], [220] arrived at the same class of codes that answered this question in the affirmative. These codes have the property that the local codes are RGCs and for this reason, are termed as locally regenerating codes (LRGCs). It follows that LRGCs share the same vector symbol alphabet $\mathbb{F}_q^\alpha$ as RGCs.

In this section, we present a minimum distance bound that applies to LRGCs. We also describe in brief, constructions that achieve the bound. We follow the approach adopted in [117].

15.1 Locality of a Code with Vector Alphabet

In this section, we will refer to a linear code having a vector alphabet as a vector code.

Definition 17 (Vector Codes)**.** A vector code $\mathcal{C}$ over $\mathbb{F}_q^\alpha$, is a linear code, i.e., a code that is closed under vector addition and multiplication

by scalars from $\mathbb{F}_q$, and where each codeword $\mathbf{c} \in \mathcal{C}$ is of the form $\mathbf{c} = (\underline{c}_0\ \underline{c}_1\ \ldots\ \underline{c}_{n-1})$, with $\underline{c}_i \in \mathbb{F}_q^{\alpha}$, for all i.

Clearly, every linear RGC is an example of a vector code. Note that every codeword can also be viewed as an $(\alpha \times n)$ array and for this reason, such codes are also referred to as array codes [23], [26] (see also the notes subsection of Section 2).

We associate with the vector code $\mathcal{C}$, a scalar code $\mathcal{C}_s$ of length $n\alpha$, obtained from $\mathcal{C}$ by concatenating the code symbols $\underline{c}_i$ of each codeword $(\underline{c}_0\ \underline{c}_1\ \ldots\ \underline{c}_{n-1})$ as shown below,

$$(\underline{c}_0^T\ \underline{c}_1^T\ \ldots\ \underline{c}_{n-1}^T),$$

to obtain a codeword in $\mathcal{C}_s$. We use K to denote the dimension of $\mathcal{C}_s$, i.e., the dimension of $\mathcal{C}_s$ when viewed as a vector space over $\mathbb{F}_q$.

Thin and Thick Columns

Let $G = [G_0\ G_1\ \ldots\ G_{n-1}]$ be a generator matrix for $\mathcal{C}_s$, where each sub-matrix G_i, is of size $(K \times \alpha)$. We will refer to the ordered set of α columns making up the ith sub-matrix G_i, as the ith thick column. Each of the $n\alpha$ columns of G will be called a thin column. Thus, the matrix G can be viewed as being comprised of $n\alpha$ thin columns; it can also be viewed as being comprised of n thick columns, where each thick column consists of α thin columns.

We will use $[[n, K, d_{\min}, \alpha]]$ to denote the parameters of the vector code $\mathcal{C}$, where $d_{\min}$ is the minimum distance of $\mathcal{C}$ when viewed as a code over the vector symbol alphabet $\mathbb{F}_q^{\alpha}$. While the underlying finite field is not identified within the notation, we will assume throughout, that the underlying finite field is $\mathbb{F}_q$. Given a subset $S \subseteq [0, n-1]$, we use $\mathcal{C}|_S$ to denote the restriction of $\mathcal{C}$ to the coordinates in S. We use $G||_S$ to denote the $(K \times |S|\alpha)$ matrix obtained by restricting G to the thick columns associated to the coordinates in S. The double bars indicate that the restriction is to a set of thick columns.

Locality

Definition 18. Let $\mathcal{C}$ be a vector code of block length n over $\mathbb{F}_q^{\alpha}$. The ith vector code symbol $\underline{c}_i$ for $0 \leq i \leq (n-1)$, is said to have (r, δ) locality,

if there exists a subset $S_i \subseteq [0, n-1]$ such that $i \in S_i$, $|S_i| \leq (r+\delta-1)$ and the minimum distance of the code $\mathcal{C}|_{S_i}$ is greater than or equal to δ. Any such code $\mathcal{C}|_{S_i}$, will be referred to as a local code.

Definition 19 ((r, δ) Information-Symbol Locality)**.** An $[[n, K, d_{\min}, \alpha]]$ vector code is said to have (r, δ) information-symbol locality if there exists a subset $\mathcal{I} \subseteq [0, n-1]$ such that:

- $\text{rank}(G||_{\mathcal{I}}) = K$, and
- for any $i \in \mathcal{I}$, the ith vector code symbol $\underline{c}_i$, has (r, δ) locality.

One can further extend the definition as follows. A vector code is said to have (r, δ) all-symbol locality, if for all $i \in [0, n-1]$, the ith vector symbol $\underline{c}_i$, has (r, δ) locality. Furthermore, if for a code having (r, δ) all-symbol locality, the subsets $\{S_i \mid i \in [0, n-1]\}$ are either identical or else, disjoint, i.e., $S_i = S_j$ or else, $|S_i \cap S_j| = 0$, for $i \neq j$, $0 \leq i, j \leq n-1$, then the vector code will be said to have all-symbol disjoint locality. All the code constructions discussed in this section will have the disjoint-locality property.

15.2 Codes with MSR/MBR Locality

Both MSR and MBR codes belong to a class of vector codes that we term here as uniform rank accumulation (URA) codes.

15.2.1 Uniform Rank Accumulation Property

Definition 20 (Uniform Rank Accumulation Codes)**.** An $[[n, K, d_{\min}, \alpha]]$ vector code having associated generator matrix G of size $(K \times n\alpha)$ is said to be an uniform rank accumulation code if there exists a sequence of n non-negative integers $\{a_1, a_2, \ldots, a_n\}$, referred to as the rank profile of the code, having the following properties:

(i) $a_1 = \alpha$, and

(ii) $\text{rank}(G||_{\mathcal{I}}) = \sum_{j=1}^{i} a_j$,

for all $\mathcal{I} \subseteq [0, n-1]$ such that $|\mathcal{I}| = i$.

The rank profile of an $\{(n,k,d),(\alpha,\beta),K,\mathbb{F}_q\}$ MSR code is given by (see for example, [211]):

$$a_i \;=\; \begin{cases} \alpha & 1 \leq i \leq k \\ 0 & (k+1) \leq i \leq n \end{cases}. \tag{15.1}$$

Note that

$$\sum_{i=1}^{n} a_i \;=\; k\alpha \;=\; K,$$

as expected.

In the case of an $\{(n,k,d),(\alpha,\beta),K,\mathbb{F}_q\}$ MBR code, the rank profile is given by [211]:

$$a_i = \begin{cases} \alpha-(i-1)\beta & 1 \leq i \leq k \\ 0 & (k+1) \leq i \leq n \end{cases}. \tag{15.2}$$

Once again, we see that

$$\sum_{i=1}^{n} a_i \;=\; k\alpha - \binom{k}{2}\beta \;=\; \left(dk - \binom{k}{2}\right)\beta \;=\; K,$$

as expected.

MSR and MBR Locality

An LRGC with MSR (similarly, MBR) locality [117], [189] is an $[[n,K,d_{\min},\alpha]]$ vector code with (r,δ) locality, where the local codes are MSR (similarly, MBR) codes having identical parameters:

$$\{(n_\ell,r,d),\ (\alpha,\beta),\ K_\ell,\ \mathbb{F}_q\},$$

where $n_\ell \;:=\; (r+\delta-1)$ and $K_\ell \leq K$. Note that the local codes will have identical rank profiles as the parameters of either an MSR or an MBR code determine its rank profile uniquely.

We now present the minimum distance upper bound appearing in [117], for a class of vector codes having URA codes as local codes. Minimum distance upper bounds for codes with MSR and MBR locality follow from this bound.

Minimum Distance Bound

We restrict our attention here to $[[n, K, d_{\min}, \alpha]]$ vector codes $\mathcal{C}$ with (r, δ) information-symbol locality, where the local codes are URA codes having identical parameters $[[n_\ell = r + \delta - 1, K_\ell, \delta, \alpha]]$ and identical rank profile $\{a_1, a_2, \ldots, a_{n_\ell}\}$. It follows that the rank profile $\{a_1, a_2, \ldots, a_{n_\ell}\}$ of each local code has the property that $a_i = 0$ for $i \geq (r+1)$.

Next, let us construct the semi-infinite, periodic sequence $b_1, b_2, b_3, \ldots$, where $b_{i+jn_\ell} \triangleq a_i$, for $1 \leq i \leq n_\ell$ and $j \geq 0$. For $s \geq 1$, we set

$$P(s) \;=\; \sum_{i=1}^{s} b_i. \tag{15.3}$$

For $y \geq 1$, set $P^{(\text{inv})}(y) = x$, where x is the smallest integer such that $P(x) \geq y$. The minimum distance of $\mathcal{C}$ is then upper bounded by the following theorem (see Theorem 4.1 in [117]):

Theorem 18. Let $\mathcal{C}$ be an $[[n, K, d_{\min}, \alpha]]$ code with (r, δ) information-symbol locality, where the local codes are URA codes having identical $[[n_\ell = r + \delta - 1, K_\ell, \delta, \alpha]]$ parameters and identical rank profile $\{a_1, \ldots, a_{n_\ell}\}$. Then, we have:

$$d_{\min} \;\leq\; n - P^{(\text{inv})}(K) + 1. \tag{15.4}$$

With respect to Theorem 18, we make the following definitions:

- A code satisfying (15.4) with equality is said to be minimum-distance-optimal. Note that in this case, we will have $K \leq P(n - d_{\min} + 1)$.
- A code that is minimum-distance-optimal is said to be rate-optimal if $K = P(n - d_{\min} + 1)$.

For a code with MSR locality, one can simplify (15.4) using (15.1) to obtain ([117], [189]):

$$d_{\min} \;\leq\; n - \left\lceil \frac{K}{\alpha} \right\rceil + 1 - \left(\left\lceil \frac{K}{\alpha r} \right\rceil - 1 \right)(\delta - 1).$$

15.2.2 Constructions for Codes with MSR Locality

In [189], the authors present an explicit construction of minimum-distance-optimal LRGCs with MSR all-symbol locality having parameters

$$[[n = \nu n_\ell,\ K \leq \nu K_\ell,\ d_{\min},\ \alpha]].$$

Here, ν is an integer satisfying $\nu \geq 2$ and each local code is an MSR code having common parameters $\{(n_\ell = (r + \delta - 1), r, d), (\alpha, \beta), K_\ell = r\alpha\}$, where δ is the minimum distance of the local MSR code. The construction requires a field-size that is exponential in n. In [117], the authors establish the existence of a minimum-distance-optimal LRGC with MSR all-symbol locality, having the reduced field-size requirement $\binom{n}{\mu}$, with all parameters remaining the same, apart from placing the additional constraint that $K = \mu\alpha$ for some integer μ such that $r \leq \mu \leq \nu r$.

15.2.3 Constructions for Codes with MBR Locality

Local Codes are MBR codes with General (n_ℓ, r, d) Parameters Given an arbitrary MBR code $\mathcal{C}_{\text{MBR}}$ having block length n_ℓ, an explicit construction of a minimum-distance-optimal LRGC with MBR all-symbol locality and block length n that is a multiple of n_ℓ is presented in [116], in which each local code is the same MBR code, namely the code $\mathcal{C}_{\text{MBR}}$. The construction makes use an approach based on pre-coding using Gabidulin codes [49], [69] and has a field-size requirement that is exponential in the block length n.

The product-matrix construction for MBR codes described in Section 4.2, yields MBR codes for any parameter sets (n_ℓ, r, d) and requires an $O(n_\ell)$ field size. In [129], the authors present an explicit construction of a minimum-distance-optimal LRGC having block length n and MBR all-symbol locality, in which the local codes are identical and correspond to a product-matrix MBR code of block length n_ℓ, with $n_\ell | n$. This construction makes use of the scalar Tamo-Barg all-symbol locality code construction described in Section 10.7, and has only a linear field-size requirement.

Local Codes are Polygonal MBR Codes The polygonal MBR code construction described in Section 4.1.1, yields MBR codes having parameter sets of the form

$$\left\{(n_\ell, r, d = n_\ell - 1),\ (\alpha = n_\ell - 1, \beta = 1),\ K_\ell = r\alpha - \binom{r}{2}\right\}$$

for any pair (n_ℓ, r) with $n_\ell > r \geq 1$. The construction makes use of a scalar MDS code precoder and the resultant MBR code possesses the RBT property. In [117], the authors present the construction of a minimum-distance-optimal LRGC with MBR all-symbol locality where the local MBR codes are polygonal MBR codes. Interestingly, this LRGC construction may be regarded as replacing the scalar MDS precoder appearing in the construction of the polygonal MBR code, with a scalar all-symbol locality code having optimal minimum distance such as the Tamo-Barg code (see Section 10.7). The resultant LRGC code has parameters

$$[[n = \nu n_\ell,\ K \leq \nu K_\ell,\ d_{\min},\ \alpha = n_\ell - 1]]$$

where $\nu \geq 2$, and the local codes are polygonal MBR codes having parameters

$$\left\{(n_\ell, r, d = n_\ell - 1), (\alpha = n_\ell - 1, \beta = 1), K_\ell = r\alpha - \binom{r}{2}\right\}.$$

This explicit construction has an $O(n^2)$ field-size requirement.

An Example LRGC with Polygonal MBR Locality In the example, we illustrate the above construction for the case $\nu = 3$. The parameters of the overall LRGC are given by:

$$[[n = 15, K = 20, d_{\min} = 5, \alpha = 4]]$$

and the local polygonal MBR codes have parameters given by

$$\{(n_\ell = 5, r = 3, d = 4), (\alpha = 4, \beta = 1), K_\ell = 9\}.$$

The construction proceeds as follows. We begin by using the pentagon MBR construction to realize each local code (see Fig. 15.1). Each

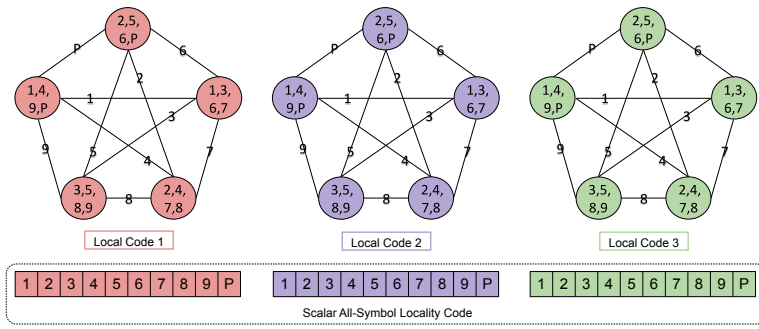

Figure 15.1: The upper portion of the figure shows an example LRGC $\mathcal{C}$ having parameters $[[n = 15, K = 20, d_{\min} = 5, \alpha = 4]]$ that is optimal with respect to (15.4). There are 3 disjoint local codes, each of which is a pentagon-MBR code having parameter set $\big\{(5, 3, 4), (4, 1), 9, \mathbb{F}_{31}\big\}$. The set of 30 scalar symbols shown in the bottom portion of the figure form a $[30, 20, 9]$ LRC with all-symbol locality (optimal with respect to (10.1)), where each local code is a $[10, 9, 2]$ MDS code. The 30 symbols of the LRC are used to label the $(3 \times 10) = 30$ edges of the 3 pentagons.

pentagon MBR code is made up of $N_\ell := \binom{n_\ell}{2} = 10$ scalar symbols over $\mathbb{F}_{31}$. Owing to the data collection property, the contents of each pentagon should be decodable from the contents of any $r = 3$ nodes. This calls for the 10 scalar symbols populating the MBR code to form a $[10, 9, 2]$ MDS code. By concatenating three codewords of the $[10, 9, 2]$ MDS code, we can populate the three pentagons and we will in this way, have satisfied the node repair and data collection properties required to say that each local code is an MBR code having the desired parameters given above. The periodic sequence $\{b_i\}$ in this case is given by

$$\big(\underbrace{4, 3, 2, 0, 0}_{\text{first period}}, \underbrace{4, 3, 2, 0, 0}_{\text{second period}}, \underbrace{4, 3, 2, 0, 0}_{\text{third period}}\big).$$

The associated sum sequence $\{P(s)\}$ is given by

$$\begin{aligned}&(P(1), P(2), \cdots, P(15))\\ &= (4, 7, 9, 9, 9, 13, 16, 18, 18, 18, 22, 25, 27, 27, 27).\end{aligned}$$

Since the desired minimum distance of the LRGC is $d_{\min} = 5$, we have that $n - d_{\min} + 1 = 15 - 5 + 1 = 11$. Hence, equivalently, we should be able to recover the data file of size $K = 20$ from the contents of any

11 among the 15 nodes. If we are able to construct such a code, it will be optimal with respect to the minimum distance since the minimum distance bound (15.4) states that

$$d_{\min} \quad \leq \quad n - P^{(\mathrm{inv})}(20) + 1 \quad = \quad 15 - 11 + 1 = 5.$$

Note that the number of distinct scalar symbols obtained by contacting any set of 11 nodes is no smaller than $(4{+}3{+}2{+}1{+}0{+}4{+}3{+}2{+}1{+}0{+}4) = 24$. Let the set of 30 scalar symbols form a Tamo-Barg code of length 30 and dimension $K = 20$ that is comprised of three support-disjoint, $[10, 9, 2]$ scalar local codes. The symbols of each of these local codes, populate the nodes associated to the three MBR codes as shown in Fig. 15.1. The minimum distance of such a Tamo-Barg code (which meets the bound (10.1)) is given by

$$d_{\min} \quad = \quad (30 - 20 + 1) \quad - \quad \left(\left\lceil \frac{20}{9} \right\rceil - 1 \right)(2 - 1) \quad = \quad 11 - 2 = 9.$$

Hence if one has access to any set of $30 - 9 + 1 = 22$ scalar symbols, one can recover all the data. On the other hand, we have access to 24 scalar symbols. Thus the data file can be recovered by contacting any 11 nodes and decoding the Tamo-Barg code. It follows that this construction is optimal with respect to the minimum distance bound in Theorem 18.

Open Problem 19. Construct minimum-distance-optimal, linear field-size LRGCs with MSR all-symbol locality, for general (n, K, n_ℓ, r, d).

16

Repair of Reed-Solomon Codes

In this section we consider the repair of a Reed-Solomon (RS) code. We will depart slightly from the notation employed thus far and assume that the symbol alphabet of the code is a finite field $\mathbb{F}$ of size q^t, i.e. $\mathbb{F} = \mathbb{F}_{q^t}$. We will use $\mathbb{B}{=}\mathbb{F}_q$ to denote the subfield of $\mathbb{F}$ of size q, which we will refer to as the base field. Thus $\mathbb{F}$ is a vector space over $\mathbb{B}$ of dimension t.

16.1 Vectorization Approach

The conventional repair of a scalar $[n, k, n-k+1]$ MDS code over $\mathbb{F}$ regards each code symbol lying in $\mathbb{F}$ as an indivisible unit, leading to a total repair bandwidth of k times the amount of data stored in the failed node, where k is the dimension of the code. RGCs, introduced in Section 3, enable node repair with significantly reduced repair bandwidth. This is made possible by the fact that RGCs have a vector symbol alphabet of the form $\mathbb{F}_q^\alpha$. This suggests that the repair bandwidth of an RS code can perhaps be reduced by regarding an RS code over the field $\mathbb{F} = \mathbb{F}_{q^t}$, instead, as a code over the vector symbol alphabet $\mathbb{F}_q^t$. Since code construction and repair schemes presented have all been linear in nature, we will use the isomorphism of $\mathbb{F} = \mathbb{F}_{q^t}$ and $\mathbb{F}_q^t$ as vector spaces of dimension t over $\mathbb{F}_q$ in this re-interpretation of an RS code.

Remark 16. A vectorized RS code may be viewed as a special instance of a concatenated code, where the outer code is the scalar $[n, k, n-k+1]$ RS code over $\mathbb{F} = \mathbb{F}_{q^t}$, and where the inner code is the trivial $[t, t, 1]$ code over the base field $\mathbb{B} = \mathbb{F}_q$.

As an example of the savings in repair bandwidth that can be achieved, consider the example of a $[16, 8, 9]$ RS code over the field $\mathbb{F} = \mathbb{F}_{2^4}$. Traditional repair requires the downloading of 8 symbols over $\mathbb{F} = \mathbb{F}_{2^4}$, corresponding to a repair bandwidth of 32 bits. As we will see later in this section, by regarding the same code instead, as a code over the vector alphabet $\mathbb{F}_2^4$, corresponding to the choice of base field $\mathbb{B} = \mathbb{F}_2$, it is possible to perform single-node repair by downloading just 1 bit each from the 15 surviving nodes, for a total repair bandwidth of 15 bits.

In general, traditional repair bandwidth of an RS code equals kt symbols over the base field $\mathbb{B}$. On the other hand, the repair bandwidth of an MSR code with $d = n - 1$ and sub-packetization level $\alpha = t$ is given by:

$$d\beta = \frac{(n-1)t}{n-k}, \tag{16.1}$$

which is in general significantly smaller. In this comparison, we have chosen the RGC to be an MSR code to match the MDS property of an RS code.

Since the field size of an RS code is typically on the order of its length n, it follows that the vector-code-symbol viewpoint corresponds to a sub-packetization level that is logarithmic in the length n. On the other hand, there are bounds (see notes subsection) showing that an exponential sub-packetization level is needed to achieve the repair bandwidth of an MSR code indicated in (16.1). Thus a study of the minimum repair bandwidth needed to repair an RS code is not only of practical interest, it also provides some insight into how repair bandwidth scales with sub-packetization level.

16.2 Tools Employed

GRS Codes We recall from Section 2 that an $[n, k]$ GRS code $\mathcal{C}$ over $\mathbb{F}$ is a code of the form:

$$\mathcal{C} \quad = \quad \{ \ (u_1 f(\alpha_1), \cdots, u_n f(\alpha_n)) \ \mid \ f(x) \in \mathbb{F}[x], \ \deg(f) < k \ \},$$

where the evaluation set $E = \{\alpha_i \mid 1 \leq i \leq n\}$, is a subset of $\mathbb{F}$ of size n, and where the $\{u_i\}_{i=1}^n$ are a set of n nonzero elements, not necessarily distinct, over $\mathbb{F}$. We will refer to the set $\{u_i\}$ as the scaling set. An RS code is a GRS code where no scaling takes place, i.e., $u_i = 1$, for all i. We will refer to an RS code where the evaluation set E is all of $\mathbb{F}$ as a full RS code. Thus the block length n of a full RS code equals the field size $|\mathbb{F}|$.

As explained in Section 2 the dual $\mathcal{C}^\perp$ of a GRS code $\mathcal{C}$ is also a GRS code, having the same evaluation set E. The scaling elements $(v_1, \cdots, v_n)$ of the dual code $\mathcal{C}^\perp$ are given by

$$v_i \quad = \quad u_i^{-1} \prod_{j=1, j \neq i}^{n} (\alpha_i - \alpha_j)^{-1}.$$

It may be noted that the dual of a full RS code is a full RS code.

16.2.1 Trace-Dual Basis

The trace function $\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}$ is the mapping from the extension field $\mathbb{F} = \mathbb{F}_{q^t}$ to the base field $\mathbb{B} = \mathbb{F}_q$ given by

$$\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(x) \quad = \quad x + x^q + x^{q^2} + \cdots + x^{q^{t-1}}.$$

It is straightforward to verify that the trace function is linear over $\mathbb{B}$. Let $(\gamma_1, \cdots, \gamma_t)$ be a basis for $\mathbb{F}$ over $\mathbb{B}$ and let x be an element of $\mathbb{F}$. Then x can be uniquely recovered from knowledge of the t trace values:

$$\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(x\gamma_i) \quad = \quad a_i \in \mathbb{B}, \quad i = 1, 2, \cdots, t.$$

This can be seen using the trace-dual basis. Associated to every basis, $(\gamma_1, \cdots, \gamma_t)$ for $\mathbb{F}$ over $\mathbb{B}$, there is a second basis, $(\gamma_1^*, \cdots, \gamma_t^*)$ for $\mathbb{F}$ over

$\mathbb{B}$, known as the trace-dual basis (for instance, see [156, Ch. 4]) that satisfies:

$$\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(\gamma_i \gamma_j^*) \quad = \quad \begin{cases} 1, & i = j, \\ 0, & \text{else.} \end{cases}$$

Using the trace-dual basis, we can recover x from $\{a_i\}_{i=1}^t$ via:

$$x \quad = \quad \sum_{i=1}^{t} a_i \ \gamma_i^*.$$

16.2.2 Repair Polynomials

We continue in the setting as above, where the field $\mathbb{F} = \mathbb{F}_{q^t}$ of definition of the RS code is a degree-t extension of a base field $\mathbb{B} = \mathbb{F}_q$. We will show in this subsection, that if one can find a set of t polynomials in $\mathbb{F}[x]$ of degree $< n-k$ satisfying certain properties, then node repair in a GRS code with smaller bandwidth is possible. We term these polynomials as *repair polynomials*. The lemma below identifies the desired properties of the set of t repair polynomials and also shows how they can be used in node repair.

Given a set of m elements $\{w_i\}_{i=1}^m$, with all $w_i \in \mathbb{F}$, we will use $\langle w_1, \cdots, w_m \rangle$ to denote the vector space over the base field $\mathbb{B}$ spanned by the m elements.

Lemma 15 (Recovery Using a Given Set of Repair Polynomials [85])**.** Let $\mathcal{C}$ be an $[n, k]$ GRS code having evaluation set $E = \{\alpha_1, \cdots, \alpha_n\}$ and scaling set $\{u_i\}_{i=1}^n$. Thus each codeword in $\mathcal{C}$ is of the form $(u_1 f(\alpha_1), \cdots, u_n f(\alpha_n))$ for some polynomial f over $\mathbb{F}$ of degree $< k$. Suppose it is possible to identify a set $\{g_1(x), \cdots, g_t(x)\}$ of t polynomials over $\mathbb{F}$, each of degree $< (n-k)$ that satisfy:

$$\dim_{\mathbb{B}} \Big(\ \langle \ g_1(\alpha_i), \ g_2(\alpha_i), \cdots, \ g_t(\alpha_i) \rangle \ \Big) \quad = \quad \begin{cases} t & i = i_0 \\ b_i & i \neq i_0. \end{cases} \tag{16.2}$$

Then the i_0th code symbol $u_{i_0} f(\alpha_{i_0})$ for $i_0 \in [n]$, can be recovered by downloading $b = \sum_{i \in [n] \setminus \{i_0\}} b_i$ symbols over the base field $\mathbb{B}$.

Proof: Let the dual code of the $[n,k]$ GRS code $\mathcal{C}$ associated to evaluation set E, and scaling set $\underline{u}=(u_1,\cdots,u_n)$ be the $[n,n-k]$ GRS code having scaling set vector $\underline{v}=(v_1,\cdots,v_n)$ (and the same evaluation set E). Since each of the repair polynomials has degree $<(n-k)$, for every $j\in[t]$ we have

$$\begin{aligned}
&(v_1g_j(\alpha_1),\cdots,v_ng_j(\alpha_n)) &&\in \mathcal{C}^{\perp},\\
\implies &\sum_{i=1}^{n} u_iv_if(\alpha_i)g_j(\alpha_i) &&= 0,\\
\implies &u_{i_0}v_{i_0}f(\alpha_{i_0})g_j(\alpha_{i_0}) &&= -\sum_{i\in[n]\setminus\{i_0\}} u_iv_if(\alpha_i)g_j(\alpha_i).
\end{aligned}$$

Applying the trace function on both sides:

$$\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}\left(u_{i_0}v_{i_0}f(\alpha_{i_0})g_j(\alpha_{i_0})\right) = -\sum_{i\in[n]\setminus\{i_0\}} \mathrm{Tr}_{\mathbb{F}/\mathbb{B}}\left(u_iv_if(\alpha_i)g_j(\alpha_i)\right). \tag{16.3}$$

Note that since

$$\dim_{\mathbb{B}}\Big(\ \langle\ g_1(\alpha_{i_0}),\ g_2(\alpha_{i_0}),\cdots,\ g_t(\alpha_{i_0})\rangle\ \Big) = t,$$

it follows that the elements $\{g_j(\alpha_{i_0})\}_{j=1}^{t}$ form a basis for $\mathbb{F}$ over $\mathbb{B}$. Hence, the values of $u_{i_0}v_{i_0}f(\alpha_{i_0})$ and hence of $f(\alpha_{i_0})$, can be determined by making use of (16.3), for all $j\in\{1,2,\cdots,t\}$. This approach requires that the ith node, $i\neq i_0$, provides the t values:

$$\left\{\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}\left(u_iv_if(\alpha_i)g_j(\alpha_i)\right)\mid j=1,2,\cdots,t\right\}. \tag{16.4}$$

However since the space

$$W_i := \langle g_1(\alpha_i),\ g_2(\alpha_i),\ \cdots,g_t(\alpha_i)\rangle,$$

has dimension b_i over $\mathbb{B}$, it follows that in place of t, b_i scalars from the base field $\mathbb{B}$ suffice to provide the information content contained in (16.4). More specifically, if the set $\{\theta_{i1},\cdots,\theta_{ib_i}\}$ is a basis for W_i, it suffices for the ith node to supply the b_i symbols:

$$\left\{\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}\ \left(u_iv_if(\alpha_i)\theta_{ij}\right)\mid j=1,2,\cdots,b_i\right\}.$$

It follows that node i_0 can be repaired using the repair bandwidth associated to a set of $b=\sum_{i\in[n]\setminus\{i_0\}} b_i$ symbols over the base field $\mathbb{B}$. $\square$

Remark 17. It turns out that the converse of Lemma 15 is also true. Let $\mathcal{C}$ be an $[n, k]$ GRS code and $(u_1 f(\alpha_1), \cdots, u_n f(\alpha_n))$ be a codeword in $\mathcal{C}$. Linear repair of the i_0th code symbol $u_{i_0} f(\alpha_{i_0})$ by downloading b_i symbols over the base field $\mathbb{B}$ from node i, for all $i \in [n] \setminus \{i_0\}$, is possible only if there exists a set of t polynomials $\{g_1(x), \cdots, g_t(x)\}$ over $\mathbb{F}$, each of degree $< (n-k)$ satisfying (16.2). We refer the reader to [85] for a proof.

16.3 Guruswami-Wootters Repair Scheme

In [85], the authors identify a set $\{g_j(x) \mid j \in [t]\}$, of repair polynomials leading to a repair scheme for an RS code having parameters $(n \leq q^t, k \leq n - q^{t-1})$ where remarkably, the repair bandwidth $b = (n-1)$, measured in number of symbols over the base field $\mathbb{B} = \mathbb{F}_q$, is as small as possible.

16.3.1 Repair Polynomials

Let $\{\gamma_1, \cdots, \gamma_t\}$ be the basis of $\mathbb{F}$ over $\mathbb{B}$. The Guruswami-Wootters (GW) scheme makes use of the following set of t repair polynomials for the repair of node i_0:

$$g_j(x) \quad = \quad \frac{\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(\gamma_j (x - \alpha_{i_0}))}{(x - \alpha_{i_0})}, \ j \in [t]. \tag{16.5}$$

Note that as required, the degree of each repair polynomial $g_j(x)$ equals $(q^{t-1} - 1) < (n-k)$. The Lemma below will establish that in the GW scheme, we have

$$b_i \quad = \quad 1, \quad 1 \leq i \leq n, \quad i \neq i_0,$$

so that the repair bandwidth of the GW scheme is $b = (n-1)$ symbols over $\mathbb{B}$. Thus, it suffices for each of the $d = (n-1)$ helper nodes to pass on just a single symbol over $\mathbb{B}$.

Lemma 16.

$$\dim_{\mathbb{B}} \Big(\langle g_1(\alpha_i), g_2(\alpha_i), \cdots, g_t(\alpha_i) \rangle \Big) \quad = \quad \begin{cases} t & i = i_0 \\ 1 & i \neq i_0. \end{cases}$$

Proof: Note that $\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(x) = x + x^q + \cdots + x^{q^{t-1}}$. It follows that $g_j(\alpha_{i_0}) = \gamma_j$ and therefore,

$$\dim_{\mathbb{B}}\Big(\langle g_1(\alpha_{i_0}), \cdots, g_t(\alpha_{i_0})\rangle\Big) \;\; = \;\; t.$$

For $i \in [n] \setminus \{i_0\}$ we clearly have:

$$\langle g_1(\alpha_i), \cdots, g_t(\alpha_i)\rangle = \left\langle \frac{1}{\alpha_i - \alpha_{i_0}} \right\rangle \text{ as } \mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(\gamma_j(\alpha_i - \alpha_{i_0})) \in \mathbb{B}.$$

In this we have used that

$$g_j(\alpha_i) \;\; = \;\; \frac{\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(\gamma_j(\alpha_i - \alpha_{i_0}))}{(\alpha_i - \alpha_{i_0})} \;\; \neq \;\; 0,$$

for at least one value of $j, j \in [t]$. It follows that

$$\dim_{\mathbb{B}}\left(\langle g_1(\alpha_i), \cdots, g_t(\alpha_i)\rangle\right) = 1.$$

□

16.4 Dau-Milenkovic Repair Scheme

The Dau-Milenkovic (DM) scheme [48] generalizes the GW scheme by making it applicable to the following larger set of RS code parameters:

$$(\; n \leq q^t, \quad k \leq n - q^s \;),$$

for $s \in [t-1]$. The repair bandwidth b achieved by this scheme satisfies: $b \leq (n-1)(t-s)$ measured once again, in units of symbols over the base field $\mathbb{B}$. When $(t-s) = 1$, the DM scheme achieves the same repair bandwidth as the GW scheme. The generalization is carried out by replacing the trace function by the larger class of linearized polynomials.

16.4.1 Repair Polynomials

We begin by introducing linearized polynomials, these polynomials are called subspace polynomials in [48].

Definition 21 (Linearized Polynomials)**.** A (monic) linearized polynomial over the field $\mathbb{F} := \mathbb{F}_{q^t}$ is a polynomial of the form

$$L(z) \quad = \quad \sum_{i=0}^{h} \ell_i z^{q^i},$$

where $\ell_i \in \mathbb{F}$, all i and $\ell_h = 1$.

Linearized polynomials are so-called as they exhibit linear behavior over the base field $\mathbb{B} = \mathbb{F}_q$:

$$L(cx+y) \quad = \quad cL(x) + L(y), \quad c \in \mathbb{B}, \ x, y \in \mathbb{F}.$$

The zeros of $L(z)$ form a subspace W over $\mathbb{B}$ of dimension h. Conversely, it can be shown that any polynomial over $\mathbb{F}$ whose zeros form a subspace W over $\mathbb{B}$ is a linearized polynomial [156]. We will use $L_W(\cdot)$ to denote the linearized polynomial whose zeros are precisely the elements of the subspace W. The trace function $\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}$ encountered earlier, is a linearized polynomial:

$$\mathrm{Tr}_{\mathbb{F}/\mathbb{B}}(z) \quad = \quad \sum_{i=0}^{t-1} z^{q^i}.$$

We now introduce a set of t repair polynomials that are based on linearized polynomials. Let $i_0 \in [n]$ be the index of the node that we wish to repair, so the aim is to recover $f(\alpha_{i_0})$. Let W be a subspace over $\mathbb{B}$ of dimension s. The t repair polynomials in the DM scheme are defined as given below:

$$g_j(x) \ = \ \frac{L_W(\gamma_j(x-\alpha_{i_0}))}{(x-\alpha_{i_0})} \ = \ \gamma_j \prod_{w \in W \setminus \{0\}} (\gamma_j(x-\alpha_{i_0}) - w), \quad (16.6)$$

for all $j \in [t]$, where $\{\gamma_1, \gamma_2, \cdots, \gamma_t\}$ is a basis for $\mathbb{F}$ over $\mathbb{B}$. Note that the $\deg(g_j(x)) = q^s - 1 < n - k$, as needed. The lemma below shows how this choice of repair polynomial set permits recovery of $f(\alpha_{i_0})$ by downloading $\leq (n-1)(t-s)$ symbols from $\mathbb{B}$.

Lemma 17.

$$\dim_{\mathbb{B}}\Big(\langle g_1(\alpha_i), \cdots, g_t(\alpha_i)\rangle\Big) \quad = \quad \begin{cases} t & i = i_0 \\ \leq t-s & i \neq i_0, \end{cases}$$

Proof: By the definition of repair polynomial in equation (16.6)

$$g_j(\alpha_{i_0}) \;=\; \gamma_j(-1)^{|W|-1} \prod_{w\in W\setminus\{0\}} w$$

Clearly, $\prod_{w\in W\setminus\{0\}} w \neq 0$ and it follows from this that

$$\dim_{\mathbb{B}}\Big(\langle g_1(\alpha_{i_0}),\cdots,g_t(\alpha_{i_0})\rangle\Big) \;=\; t,$$

as $\{\gamma_1,\cdots,\gamma_t\}$ is a basis for $\mathbb{F}$ over $\mathbb{B}$. For the case $i\in[n]\setminus\{i_0\}$, we have

$$g_j(\alpha_i) \;=\; \frac{L_W(\gamma_j(\alpha_i-\alpha_{i_0}))}{(\alpha_i-\alpha_{i_0})}.$$

As a result,

$$\begin{aligned}&\dim_{\mathbb{B}}\Big(\langle g_1(\alpha_i),\cdots,g_t(\alpha_i)\rangle\Big)\\ =\;&\dim_{\mathbb{B}}\Big(\langle L_W(\gamma_1(\alpha_i-\alpha_{i_0})),\cdots,L_W(\gamma_t(\alpha_i-\alpha_{i_0}))\rangle\Big) \;\leq\; t-s.\end{aligned}$$

The last inequality follows by regarding $L_W(\cdot)$ as a linear mapping from a t-dimensional space $\mathbb{F}$ back to $\mathbb{F}$, while having a kernel W of dimension s. □

16.5 Bounds on Repair-Bandwidth

Let $\mathcal{C}$ be an $[n,k]$ GRS code over $\mathbb{F}$. We claim [48], [85] that any linear repair scheme for $\mathcal{C}$ will necessarily incur a repair bandwidth of at least:

$$b \;\geq\; (n-1)\log_q \frac{q^t(n-1)}{(n-k-1)(q^t-1)+n-1} \tag{16.7}$$

units, measured in terms of number of symbols over the subfield $\mathbb{B}$.

Proof. From Remark 17, we know that given an evaluation point $\alpha_{i_0}\in\mathbb{F}$, there exists a set $\{g_1(x),g_2(x),\ldots,g_t(x)\}$, $g_j(x)\in\mathbb{F}[x]$, of t repair polynomials, each having degree $<(n-k)$, such that:

$$\dim_{\mathbb{B}}\Big(\langle g_1(\alpha_{i_0}),g_2(\alpha_{i_0}),\cdots,g_t(\alpha_{i_0})\rangle\Big) \;=\; t.$$

Let

$$\dim_{\mathbb{B}}\Big(\langle g_1(\alpha_i), g_2(\alpha_i), \cdots, g_t(\alpha_i)\rangle\Big) \;=\; d_i,$$

for all $i \in [n] \setminus \{i_0\}$. It follows that the repair bandwidth needed to recover the code symbol corresponding to the evaluation point α_{i_0} using (16.3) is now given by $\sum_{i\in[n]\setminus\{i_0\}} d_i$. For $i \in [n] \setminus \{i_0\}$, let the subspace $S_i \subseteq \mathbb{B}^t$ over $\mathbb{B}$ be defined as follows:

$$S_i \;:=\; \left\{ s \;=\; (s_1, s_2, \ldots, s_t) \in \mathbb{B}^t \;\Big|\; \sum_{j\in[t]} s_j g_j(\alpha_i) = 0 \right\}.$$

We have $\dim_{\mathbb{B}}(S_i) = t - d_i$ and hence the cardinality of the set of nonzero elements in S_i is given by $q^{t-d_i} - 1$. As the next step, we determine the average number ρ of sets $\{S_i, i \in [n] \setminus \{i_0\}\}$ that a nonzero element in $\mathbb{B}^t$ belongs to:

$$\begin{aligned}
\rho \;&:=\; \frac{1}{(q^t-1)} \sum_{s\neq 0, s\in\mathbb{B}^t} |\{i \in [n] \setminus \{i_0\} : s \in S_i\}| \\
&=\; \frac{1}{(q^t-1)} \sum_{i\in[n]\setminus\{i_0\}} |\{s \in S_i,\; s \neq 0\}| \\
&=\; \frac{1}{(q^t-1)} \sum_{i\in[n]\setminus\{i_0\}} (q^{t-d_i} - 1). \qquad (16.8)
\end{aligned}$$

Clearly, there exists a t-tuple $s^* := (s_1^*, s_2^*, \ldots, s_t^*) \in \mathbb{B}^t \setminus \{0\}$, such that the polynomial $g^*(x) = \sum_{j\in[t]} s_j^* g_j(x)$ vanishes on at least ρ evaluation points. Furthermore, $g^*(\alpha_{i_0}) \neq 0$ as

$$\dim_{\mathbb{B}}\Big(\langle g_1(\alpha_{i_0}), g_2(\alpha_{i_0}), \cdots, g_t(\alpha_{i_0})\rangle\Big) = t.$$

This tells us that $g^*(x)$ is a non-zero polynomial of degree $< (n-k)$ that has at least ρ zeros. It follows that

$$\rho \;\leq\; n - k - 1. \qquad (16.9)$$

From (16.8) and (16.9), one obtains

$$\sum_{i\in[n]\setminus\{i_0\}} q^{-d_i} \;\leq\; \frac{(n-k-1)(q^t-1)+(n-1)}{q^t} \;:=\; \rho'. \qquad (16.10)$$

The repair bandwidth b is lower bounded by the quantity

$$\min_{\{d_i \in [0,t]\}} \sum_{i \in [n], i \neq i_0} d_i$$

subject to (16.10). It turns out that the minimum occurs when $\{d_i\}$ are balanced and this results in the following lower bound on repair bandwidth:

$$\begin{aligned} b &\geq (n-1) \log_q \left(\frac{n-1}{\rho'} \right) \\ &= (n-1) \log_q \frac{q^t (n-1)}{(n-k-1)(q^t - 1) + n - 1}. \end{aligned}$$

□

Corollary 5. If $n = q^t$ and $n - k = q^s$ for some $s \in [t-1]$, any linear repair scheme for an $[n, k]$ GRS code requires a repair bandwidth of at least (measured over $\mathbb{B}$):

$$\begin{aligned} b &\geq (n-1) \log_q \left(\frac{q^t}{q^s} \right) \\ &= (n-1)(t-s). \end{aligned} \tag{16.11}$$

16.5.1 Optimality of Repair Schemes

It can be verified that

- when $n = q^t$ and $n - k = q^{t-1}$ the GW scheme achieves the lower bound presented in (16.11) and
- when $n = q^t$ and $n - k = q^s$, for $s \in [t-1]$, the DM scheme achieves the lower bound presented in (16.11).

Notice that the sub-packetization $t = \log_q n$ in both schemes.

Notes

1. An early paper: The line of work in which scalar MDS codes are vectorized by treating each code symbol belonging to a field $\mathbb{F}$, as a vector over a base field $\mathbb{B}$, for the purpose of reducing

repair bandwidth, began with the work of Shanmugam *et al.* in [215]. Here, the authors showed the existence of an efficient repair scheme for systematic node repair, that improves upon the repair bandwidth incurred under traditional repair, for the case $k = n-2$.

2. Achieving the cut-set bound: The Tamo-Ye-Barg RS repair scheme in [234] has sub-packetization level $t = e^{(1+o(1))(n \log n)}$ and achieves the cut-set bound $b \geq \frac{t(n-1)}{(n-k)}$ on the minimum possible repair bandwidth b of an MDS code. It is also shown in [234], that the sub-packetization level required for linear repair of a scalar MDS code, having minimum possible repair bandwidth, must satisfy $t \geq e^{(1+o(1))(k \log k)}$. In [43], the authors show that given a scalar MDS code that achieves the cut-set bound on minimal repair bandwidth, it is possible to replace the repair scheme employed here, by an optimal-access repair scheme.

3. Trading increased sub-packetization level for reduced repair bandwidth: The sub-packetization level t appearing in the GW and DM schemes is of logarithmic order with respect to block length n, i.e., $t = \log_q(n)$. The RS repair scheme presented in [254], has exponential sub-packetization level given by $t = (n-k)^n$, but a smaller repair bandwidth b satisfying $b < \frac{t(n+1)}{(n-k)}$ and this value of repair bandwidth is optimal in the limit as $n \to \infty$. In [44], this result is refined, resulting in a scheme having smaller sub-packetization $t = u^{m+n-1}$ where $n-k = u^m$ for integers u, m, but which continues to asymptotically achieve the cut-set bound. The tradeoff between sub-packetization and repair bandwidth of RS codes is further explored in [82], [140].

4. Multiple node repair: In [47], the authors extend the GW scheme and formulate RS repair schemes for the case of two or three node failures. The authors of [157] provide a general framework for efficiently handling multiple erasures for scalar MDS codes. In [234], an RS repair scheme achieving the cut-set bound for multiple node failures is presented, that has large sub-packetization level.

5. Improved repair of the $[14, 10]$ HDFS RS code: In [58], Duursma and Dau consider the $[14, 10, 5]$ RS code over $\mathbb{F}_{2^8}$ employed in

the Hadoop Distributed File System, and present a repair scheme having reduced repair bandwidth of 54 bits in comparison with the 80 bits required under conventional repair.

Open Problem 20. Determine the minimum-possible sub-packetization level of an RS repair scheme that achieves the cut-set bound $b \geq \frac{t(n-1)}{(n-k)}$ on repair bandwidth with equality.

Open Problem 21. Determine the smallest repair bandwidth permitted by an RS repair scheme for given parameters $\{n, k, q, t\}$.

17

Codes in Practice

MDS Codes

Distributed systems such as Hadoop, Google File System and Windows Azure have evolved to support erasure codes so as to derive the benefits of improved storage efficiency in comparison with simple replication. MDS codes in general, and RS codes in particular, are the most common form of erasure coding employed here. Examples include the $[9, 6]$ RS code employed in the Hadoop Distributed File System, the $[14, 10]$ RS code in Facebook's f4 Storage System and the $[11, 8]$ RS code employed in Yahoo Cloud Object Storage, see [47] for additional examples.

Regenerating Codes

NCCloud: The NCCloud storage system described in [98], is one of the earliest projects that dealt with the performance evaluation of RGCs in practice. The NCCloud system employs an $(n, k = n - 2, d = n - 1)$ MSR code with functional repair. The performance evaluation is carried out for an $(n = 4, k = 2, d = 3)$ case and compared against RAID-6.

Codes with Inherent Double Replication: In [130], the performance of two codes is studied in a Hadoop setting. The first code is the Pentagon

MBR code, discussed in Section 4.1. The second code is a variant of LRGC employing MBR local codes, that is termed the Heptagon-Local code. Both codes possess inherent double replication of code symbols, have storage overhead slightly greater than 2 and in the study, their performance is compared against schemes that employ double and triple replication.

PM-RBT Code: In [180], the authors present an optimal-access version of the PM-MSR code, which they refer to as the PM-RBT code. The results of an experimental evaluation of $(n = 12, k = 6, d = 11)$ PM-RBT code on Amazon EC2 instances are presented.

Beehive Codes: In [139], the authors introduced erasure codes termed as Beehive codes that make use of PM-MSR codes in their construction. These codes repair multiple failures simultaneously and are implemented in C++ using the Intel storage acceleration library. The performance of the $(n = 12, k = 6, d = 10)$ Beehive code for two-node repair is compared against that of an MSR code having the same parameters as well as against that of an $[n = 12, k = 6]$ RS code on Amazon EC2 instances.

Butterfly Code: In [166], the authors present the evaluation of a high-rate MSR code known as the Butterfly code in both Ceph and HDFS. This code is a simplified version of the MSR codes presented in [65] corresponding to the presence of two parity nodes. The code possesses the optimal-access property except in the case of the repair of a specific parity node, and has sub-packetization level $\alpha = 2^{k-1}$. In [166], the authors present the repair performance of Butterfly codes having parameters $(n = 7, k = 5, d = 6)$ and $(n = 9, k = 7, d = 8)$.

Clay Code in Ceph: In [240], the authors present an implementation and evaluation of the CL-MSR code (known in this context by the acronym, Clay code) in the Ceph distributed storage system. Clay codes are the first known implementation of MSR codes for general (n, k). They were also made part of Ceph's release [37] as an erasure code

plugin. As a part of this open source work, vector code support is added to Ceph that enables introducing any other vector erasure code plugin in the future. In [240], results of experimental evaluation of repair performance of Clay code are provided for six different code parameters.

MDS Codes with Reduced Repair Bandwidth

Hitchhiker System: The Hitchhiker erasure-coded system presented in [183] is a practical implementation of the piggybacking framework introduced in [184]. The authors implemented the Hitchhiker in HDFS and evaluated the performance of $(n = 14, k = 10, \alpha = 2)$ code on a data-warehouse cluster at Facebook.

Hashtag Codes: In [128], the HDFS implementation of a class of MDS array codes called HashTag codes is discussed. The theoretical framework of HashTag codes was presented in [127]. These codes allow low sub-packetization levels at the expense of increased repair bandwidth and are designed to efficiently repair systematic nodes. The repair performance of several Hashtag codes with different sub-packetization levels are presented in [128].

LRCs

Windows Azure Code: In [103], the authors compare performance evaluation results of an $(n = 16, k = 12, r = 6, \delta = 2)$ LRC with that of an $[16, 12, 5]$ RS code in the Azure production cluster and demonstrate the repair savings of LRCs. Subsequently the authors [101] implemented an $(n = 18, k = 14, r = 7, \delta = 2)$ LRC in Microsoft's Windows Azure Storage system and showed that this code has repair degree comparable to that of an $[9, 6, 4]$ RS code, but has storage overhead 1.29 versus 1.5 in the case of the RS code. This reduction in storage overhead has reportedly resulted in significant cost savings for Microsoft [161].

HDFS-Xorbas: The authors of [207] implemented HDFS-Xorbas which uses LRCs in place of RS codes in HDFS-RAID. The Xorbas LRC is built on top of an RS code by adding extra local XOR parties. The experimental evaluation of Xorbas was carried out in Amazon EC2 as

well as a cluster in Facebook. In the evaluation, the repair performance of an $(n = 16, k = 10, r = 5, \delta = 2)$ LRC was compared against that of an $[14, 10, 5]$ RS code.

LRCs in Ceph: Ceph is a second distributed storage system that has an LRC plug-in [149]. In [124], a performance comparison of different LRCs is provided, through experimental evaluation over a Ceph cluster.

Acknowledgements

We thank the editors of Science China Information Sciences for allowing reuse of some material from our previously-published article [10]. We would like to thank the editor-in-chief and the publisher for the invitation to write the monograph, as well as for being patient with respect to the submission timeline. Thanks also go out to the editor-in-chief for the helpful initial comments that guided the writing of this monograph.

We would like to thank the anonymous reviewer for the very careful reading and the detailed comments, which helped significantly improve the presentation and coverage of the material. The last author would like to thank Kannan Ramchandran for introducing him to this research topic. He would also like to acknowledge support received under the J C Bose National Fellowship JCB/2017/000017.

References

[1] A. Agarwal, A. Barg, S. Hu, A. Mazumdar, and I. Tamo, "Combinatorial alphabet-dependent bounds for locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, 2018, pp. 3481–3492.

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

[3] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, 2000, pp. 1204–1216.

[4] I. Ahmad and C.-C. Wang, "When can intelligent helper node selection improve the performance of distributed storage networks?" *IEEE Trans. Inf. Theory*, vol. 64, no. 3, 2017, pp. 2142–2171.

[5] N. Alon, "Combinatorial Nullstellensatz," *Combinatorics, Probability and Computing*, vol. 8, no. 1-2, 1999, pp. 7–29.

[6] O. Alrabiah and V. Guruswami, "An exponential lower bound on the sub-packetization of MSR codes," in *Proc. 51st Annual ACM SIGACT Symposium on Theory of Computing, Phoenix, AZ, USA, 2019*, pp. 979–985, 2019.

[7] O. Alrabiah and V. Guruswami, "An exponential lower bound on the sub-packetization of minimum storage regenerating codes," *IEEE Trans. Inf. Theory*, 2021.

[8] B. S. Babu and P. V. Kumar, "Erasure codes for distributed storage: Tight bounds and matching constructions," *CoRR*, vol. abs/1806.04474, 2018.

[9] B. S. Babu, M. Vajha, and P. V. Kumar, "On lower bounds on sub-packetization level of MSR codes and on the structure of optimal-access MSR codes achieving the bound," *CoRR*, vol. abs/1710.05876v3, 2021.

[10] S. B. Balaji, M. N. Krishnan, M. Vajha, V. Ramkumar, B. Sasidharan, and P. V. Kumar, "Erasure coding for distributed storage: An overview," *Science China Information Sciences*, vol. 61, no. 10, 2018, pp. 1–45.

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

[12] S. B. Balaji, K. P. Prasanth, and P. V. Kumar, "Binary codes with locality for multiple erasures having short block length," in *Proc. IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, 2016*, pp. 655–659, 2016.

[13] S. B. Balaji, G. R. Kini, and P. V. Kumar, "A tight rate bound and matching construction for locally recoverable codes with sequential recovery from any number of multiple erasures," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, 2020, pp. 1023–1052.

[14] S. B. Balaji and P. V. Kumar, "Bounds on the rate and minimum distance of codes with availability," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 3155–3159, 2017.

[15] S. B. Balaji and P. V. Kumar, "On partial maximally-recoverable and maximally-recoverable codes," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, China, 2015*, pp. 1881–1885, 2015.

[16] S. Ball, "On sets of vectors of a finite vector space in which every subset of basis size is a basis," *Journal of the European Mathematical Society*, vol. 14, no. 3, 2012, pp. 733–748.

[17] S. Ball and J. De Beule, "On sets of vectors of a finite vector space in which every subset of basis size is a basis II," *Designs, Codes and Cryptography*, vol. 65, no. 1, 2012, pp. 5–14.

[18] S. Ballentine, A. Barg, and S. Vladuţ, "Codes with hierarchical locality from covering maps of curves," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, 2019, pp. 6056–6071.

[19] A. Barg, I. Tamo, and S. Vlăduţ, "Locally recoverable codes on algebraic curves," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, 2017, pp. 4928–4939.

[20] A. Barg, Z. Chen, and I. Tamo, "A construction of maximally recoverable codes," *Designs, Codes and Cryptography*, vol. 90, no. 4, 2022, pp. 939–945.

[21] A. Barg, K. Haymaker, E. W. Howe, G. L. Matthews, and A. Várilly-Alvarado, "Locally recoverable codes from algebraic curves and surfaces," in *Algebraic Geometry for Coding Theory and Cryptography*, Springer, 2017, pp. 95–127.

[22] S. Bhadane and A. Thangaraj, "Unequal locality and recovery for locally recoverable codes with availability," in *Twenty-third National Conference on Communications, Chennai, India, 2017*, pp. 1–6, 2017.

[23] M. Blaum, P. G. Farrell, and H. C. A. van Tilborg, "Array codes," in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, Eds., vol. 2, Amsterdam, The Netherlands: North Holland, 1998, pp. 1855–1909.

[24] M. Blaum, "Construction of PMDS and SD codes extending RAID 5," *CoRR*, vol. abs/1305.0032, 2013.

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

[26] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, 1996, pp. 529–542.

[27] M. Blaum, J. L. Hafner, and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, 2013, pp. 4510–4519.

[28] M. Blaum, J. S. Plank, M. Schwartz, and E. Yaakobi, "Construction of partial MDS and sector-disk codes with two global parity symbols," *IEEE Trans. Inf. Theory*, vol. 62, no. 5, 2016, pp. 2673–2681.

[29] T. Bogart, A. Horlemann-Trautmann, D. A. Karpuk, A. Neri, and M. Velasco, "Constructing partial MDS codes from reducible algebraic curves," *SIAM J. Discret. Math.*, vol. 35, no. 4, 2021, pp. 2946–2970.

[30] V. R. Cadambe and A. Mazumdar, "Bounds on the size of locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, 2015, pp. 5787–5794.

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

[32] 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 *Porc. IEEE International Symposium on Information Theory Proceedings, ISIT 2011, St. Petersburg, Russia, 2011*, pp. 1225–1229, 2011.

[33] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra, "Polynomial length MDS codes with optimal repair in distributed storage," in *Proc. Forty Fifth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA 2011*, pp. 1850–1854, 2011.

[34] H. Cai, Y. Miao, M. Schwartz, and X. Tang, "On optimal locally repairable codes with super-linear length," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, 2020, pp. 4853–4868.

[35] H. Cai, Y. Miao, M. Schwartz, and X. Tang, "A construction of maximally recoverable codes with order-optimal field size," *IEEE Trans. Inf. Theory*, vol. 68, no. 1, 2022, pp. 204–212.

[36] G. Calis and O. O. Koyluoglu, "A general construction for PMDS codes," *IEEE Communications Letters*, vol. 21, no. 3, 2017, pp. 452–455.

[37] *Ceph V14.1.0 Nautilus (Release Candidate 1)*, URL: https://docs.ceph.com/en/nautilus/releases/nautilus/.

[38] B. Chen, S. T. Xia, J. Hao, and F. W. Fu, "Constructions of optimal cyclic (r, δ) locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, 2018, pp. 2499–2511.

[39] J. Chen, K. W. Shum, Q. Yu, and C. W. Sung, "Sector-disk codes and partial MDS codes with up to three global parities," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, China, 2015*, pp. 1876–1880, 2015.

[40] M. Chen, C. Huang, and J. Li, "On the maximally recoverable property for multi-protection group codes," in *Proc. IEEE International Symposium on Information Theory, Nice, France, 2007*, pp. 486–490, 2007.

[41] Z. Chen and A. Barg, "Explicit constructions of MSR codes for clustered distributed storage: The rack-aware storage model," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, 2019, pp. 886–899.

[42] Z. Chen and A. Barg, "Cyclic LRC codes with hierarchy and availability," in *IEEE International Symposium on Information Theory, Los Angeles, CA, USA, 2020*, pp. 616–621, 2020.

[43] Z. Chen, M. Ye, and A. Barg, "Enabling optimal access and error correction for the repair of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, 2020, pp. 7439–7456.

[44] A. Chowdhury and A. Vardy, "Improved schemes for asymptotically optimal repair of MDS codes," *IEEE Trans. Inf. Theory*, vol. 67, no. 8, 2021, pp. 5051–5068.

[45] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," in *Proc. 3rd USENIX Conference on File and Storage Technologies*, San Francisco, CA, pp. 1–14, 2004.

[46] A. Datta and F. E. Oggier, "An overview of codes tailor-made for better repairability in networked distributed storage systems," *SIGACT News*, vol. 44, no. 1, 2013, pp. 89–105.

[47] H. Dau, I. M. Duursma, H. M. Kiah, and O. Milenkovic, "Repairing Reed-Solomon codes with multiple erasures," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, 2018, pp. 6567–6582.

[48] H. Dau and O. Milenkovic, "Optimal repair schemes for some families of full-length Reed-Solomon codes," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 346–350, 2017.

[49] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. Comb. Theory, Ser. A*, vol. 25, no. 3, 1978, pp. 226–241.

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

[51] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, no. 3, 2011, pp. 476–489.

[52] A. Duminuco and E. W. Biersack, "Hierarchical codes: How to make erasure codes attractive for peer-to-peer storage systems," in *Proc. P2P'08, Eighth International Conference on Peer-to-Peer Computing, 2008, Aachen, Germany*, pp. 89–98, IEEE Computer Society, 2008.

[53] A. Duminuco and E. W. Biersack, "Hierarchical codes: A flexible trade-off for erasure codes in peer-to-peer storage systems," *Peer-to-Peer Netw. Appl.*, vol. 3, no. 1, 2010, pp. 52–66.

[54] I. Duursma, X. Li, and H.-P. Wang, "Multilinear algebra for distributed storage," *SIAM Journal on Applied Algebra and Geometry*, vol. 5, no. 3, 2021, pp. 552–587.

[55] I. Duursma and H.-P. Wang, "Multilinear algebra for minimum storage regenerating codes: A generalization of the product-matrix construction," *Applicable Algebra in Engineering, Communication and Computing*, 2021, pp. 1–27.

[56] I. M. Duursma, "Shortened regenerating codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, 2018, pp. 1000–1007.

[57] I. M. Duursma, "Outer bounds for exact repair codes," *CoRR*, vol. abs/1406.4852, 2014.

[58] I. M. Duursma and H. Dau, "Low bandwidth repair of the RS(10, 4) Reed-Solomon code," in *Proc. Information Theory and Applications Workshop, San Diego, CA, USA, 2017*, pp. 1–10, 2017.

[59] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *Proc. 48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1510–1517, 2010.

[60] M. Elyasi, S. Mohajer, and R. Tandon, "Linear exact repair rate region of (k + 1, k, k) distributed storage systems: A new approach," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, China, 2015*, pp. 2061–2065, 2015.

[61] M. Elyasi and S. Mohajer, "Determinant coding: A novel framework for exact-repair regenerating codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, 2016, pp. 6683–6697.

[62] M. Elyasi and S. Mohajer, "A cascade code construction for (n, k, d) distributed storage systems," in *Proc. IEEE International Symposium on Information Theory, Vail, CO, USA, 2018*, pp. 1241–1245, 2018.

[63] M. Elyasi and S. Mohajer, "Determinant codes with helper-independent repair for single and multiple failures," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, 2019, pp. 5469–5483.

[64] M. Elyasi and S. Mohajer, "Cascade codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, 2020, pp. 7490–7527.

[65] E. En Gad, R. Mateescu, F. Blagojevic, C. Guyot, and Z. Bandic, "Repair-optimal MDS array codes over GF(2)," in *Proc. IEEE International Symposium on Information Theory, Istanbul, Turkey, 2013*, pp. 887–891, 2013.

[66] T. Ernvall, T. Westerback, and C. Hollanti, "Constructions of optimal and almost optimal locally repairable codes," in *Proc. 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems, 2014*, pp. 1–5, 2014.

[67] T. Ernvall, "The existence of fractional repetition codes," *CoRR*, vol. abs/1201.3547, 2012.

[68] M. Forbes and S. Yekhanin, "On the locality of codeword symbols in non-linear codes," *Discrete Math.*, vol. 324, 2014, pp. 78–84.

[69] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Problemy Peredachi Informatsii*, vol. 21, no. 1, 1985, pp. 3–16.

[70] R. Gabrys, E. Yaakobi, M. Blaum, and P. H. Siegel, "Constructions of partial MDS codes over small fields," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, 2019, pp. 3692–3701.

[71] P. Gopalan, C. Huang, B. Jenkins, and S. Yekhanin, "Explicit maximally recoverable codes with locality," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, 2014, pp. 5245–5256.

[72] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, 2012, pp. 6925–6934.

[73] P. Gopalan, G. Hu, S. Kopparty, S. Saraf, C. Wang, and S. Yekhanin, "Maximally recoverable codes for grid-like topologies," in *Proc. Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, Barcelona, Spain*, pp. 2092–2108, 2017.

[74] S. Goparaju and A. R. Calderbank, "Binary cyclic codes that are locally repairable," in *Proc. IEEE International Symposium on Information Theory, Honolulu, HI, USA, 2014*, pp. 676–680, 2014.

[75] S. Goparaju, S. El Rouayheb, A. R. Calderbank, and H. V. Poor, "Data secrecy in distributed storage systems under exact repair," in *Proc. International Symposium on Network Coding, Calgary, Canada, 2013*, pp. 1–6, 2013.

[76] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, 2017, pp. 6318–6328.

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

[78] S. Gopi and V. Guruswami, "Improved maximally recoverable LRCs using skew polynomials," *Electron. Colloquium Comput. Complex.*, 2021, p. 25.

[79] S. Gopi, V. Guruswami, and S. Yekhanin, "Maximally recoverable LRCs: A field size lower bound and constructions for few heavy parities," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, 2020, pp. 6066–6083.

[80] M. Grezet, T. Westerbäck, R. Freij-Hollanti, and C. Hollanti, "Uniform minors in maximally recoverable codes," *IEEE Communications Letters*, vol. 23, no. 8, 2019, pp. 1297–1300.

[81] M. K. Gupta, A. Agrawal, and D. Yadav, "On weak dress codes for cloud storage," *CoRR*, vol. abs/1302.3681, 2013.

[82] V. Guruswami and H. Jiang, "Near-optimal repair of Reed-Solomon codes with low sub-packetization," in *Proc. IEEE International Symposium on Information Theory, Paris, France, 2019*, pp. 1077–1081, 2019.

[83] V. Guruswami, L. Jin, and C. Xing, "Constructions of maximally recoverable local reconstruction codes via function fields," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, 2020, pp. 6133–6143.

[84] V. Guruswami and A. S. Rawat, "MDS code constructions with small sub-packetization and near-optimal repair bandwidth," in *Proc. Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, Barcelona, Spain, 2017*, pp. 2109–2122, 2017.

[85] V. Guruswami and M. Wootters, "Repairing Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, 2017, pp. 5684–5698.

[86] V. Guruswami, C. Xing, and C. Yuan, "How long can optimal locally repairable codes be?" *IEEE Trans. Inf. Theory*, vol. 65, no. 6, 2019, pp. 3662–3670.

[87] J. Han and L. A. Lastras-Montano, "Reliable memories with subline accesses," in *Proc. IEEE International Symposium on Information Theory, Nice, France, 2007*, pp. 2531–2535, 2007.

[88] J. Hao, S. T. Xia, and B. Chen, "Some results on optimal locally repairable codes," in *Proc. IEEE International Symposium on Information Theory, Barcelona, Spain, 2016*, pp. 440–444, 2016.

[89] J. Hao, S. T. Xia, and B. Chen, "On optimal ternary locally repairable codes," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 171–175, 2017.

[90] J. Hao, K. Shum, S.-T. Xia, and Y.-X. Yang, "On the maximal code length of optimal linear locally repairable codes," in *Proc. IEEE International Symposium on Information Theory, Vail, CO, USA, 2018*, 2018.

[91] J. Hao, S. Xia, and B. Chen, "On the linear codes with (r, δ)-locality for distributed storage," in *Proc. IEEE International Conference on Communications, ICC 2017, Paris, France, May 21-25, 2017*, pp. 1–6, IEEE.

[92] J. Hao, S. Xia, K. W. Shum, B. Chen, F. Fu, and Y. Yang, "Bounds and constructions of locally repairable codes: Parity-check matrix approach," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, 2020, pp. 7465–7474.

[93] K. Haymaker, B. Malmskog, and G. L. Matthews, "Locally recoverable codes with availability $t \geq 2$ from fiber products of curves," *Adv. Math. Commun.*, vol. 12, no. 2, 2018, pp. 317–336.

[94] T. Helleseth, T. Klove, V. I. Levenshtein, and O. Ytrehus, "Bounds on the minimum support weights," *IEEE Trans. Inf. Theory*, vol. 41, no. 2, 1995, pp. 432–440.

[95] H. Hou, P. P. C. Lee, K. W. Shum, and Y. Hu, "Rack-aware regenerating codes for data centers," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, 2019, pp. 4730–4745.

[96] G. Hu and S. Yekhanin, "New constructions of SD and MR codes over small finite fields," in *Proc. IEEE International Symposium on Information Theory, Barcelona, Spain, 2016*, pp. 1591–1595, 2016.

[97] P. Hu, C. W. Sung, and T. H. Chan, "Broadcast repair for wireless distributed storage systems," in *Proc. 10th International Conference on Information, Communications and Signal Processing, Singapore, 2015*, pp. 1–5, 2015.

[98] Y. Hu, H. C. H. Chen, P. P. C. Lee, and Y. Tang, "NCCloud: Applying network coding for the storage repair in a cloud-of-clouds," in *Proc. 10th USENIX conference on File and Storage Technologies, San Jose, CA, USA, 2012*, p. 21, 2012.

[99] Y. Hu, P. P. C. Lee, and X. Zhang, "Double regenerating codes for hierarchical data centers," in *Proc. IEEE International Symposium on Information Theory, Barcelona, Spain, 2016*, pp. 245–249, 2016.

[100] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, "Cooperative recovery of distributed storage systems from multiple losses with network coding," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, 2010, pp. 268–276.

[101] C. Huang, "Erasure Coding in Windows Azure Storage," talk presented at the SNIA Storage Developer Conference, Santa Clara, Sept 12-15, 2012 (joint work with H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li and S. Yekhanin).

[102] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in *Proc. 6th IEEE Int. Symposium on Network Computing and Applications, Cambridge, Massachusetts, USA, 2007*, pp. 79–86, 2007.

[103] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows azure storage," in *Proc. 2012 USENIX Annual Technical Conference*, pp. 15–26, Boston, MA, 2012.

[104] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," *ACM Trans. Storage*, vol. 9, no. 1, 2013, 3:1–3:28.

[105] K. Huang, U. Parampalli, and M. Xian, "On secrecy capacity of minimum storage regenerating codes," *IEEE Trans. on Inf. Theory*, vol. 63, no. 3, 2017, pp. 1510–1524.

[106] K. Huang, U. Parampalli, and M. Xian, "Security concerns in minimum storage cooperative regenerating codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, 2016, pp. 6218–6232.

[107] K. Huang, U. Parampalli, and M. Xian, "Improved upper bounds on systematic-length for linear minimum storage regenerating codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, 2018, pp. 975–984.

[108] P. Huang, E. Yaakobi, H. Uchikawa, and P. H. Siegel, "Cyclic linear binary locally repairable codes," in *Proc. IEEE Information Theory Workshop, Jerusalem, Israel, 2015*, pp. 1–5, 2015.

[109] P. Huang, E. Yaakobi, H. Uchikawa, and P. H. Siegel, "Binary linear locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, 2016, pp. 6268–6283.

[110] *Information theory inequality prover*, URL: http://user-www.ie.cuhk.edu.hk/~ITIP/.

[111] L. Jin, H. Kan, and Y. Zhang, "Constructions of locally repairable codes with multiple recovering sets via rational function fields," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, 2020, pp. 202–209.

[112] L. Jin, L. Ma, and C. Xing, "Construction of optimal locally repairable codes via automorphism groups of rational function fields," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, 2019, pp. 210–221.

[113] S. Kadhe and A. Sprintson, "Security for minimum storage regenerating codes and locally repairable codes," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 1028–1032, 2017.

[114] S. Kadhe and A. R. Calderbank, "Rate optimal binary linear locally repairable codes with small availability," *CoRR*, vol. abs/1701.02456, 2017.

[115] S. Kadhe and A. R. Calderbank, "Rate optimal binary linear locally repairable codes with small availability," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 166–170, 2017.

[116] G. M. Kamath, N. Silberstein, N. Prakash, A. S. Rawat, V. Lalitha, O. O. Koyluoglu, P. V. Kumar, and S. Vishwanath, "Explicit MBR all-symbol locality codes," in *Proc. IEEE International Symposium on Information Theory, Istanbul, Turkey, 2013*, IEEE, pp. 504–508, 2013.

[117] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration and erasure correction," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, 2014, pp. 4637–4660.

[118] A. M. Kermarrec, N. L. Scouarnec, and G. Straub, "Repairing multiple failures with coordinated and adaptive regenerating codes," in *Proc. International Symposium on Networking Coding, Beijing, China, 2011*, pp. 1–6, 2011.

[119] C. Kim and J. S. No, "New constructions of binary and ternary locally repairable codes using cyclic codes," *IEEE Communications Letters*, vol. 22, no. 2, 2018, pp. 228–231.

[120] M. Kleckler and S. Mohajer, "Secure determinant codes: A class of secure exact-repair regenerating codes," in *Proc. IEEE International Symposium on Information Theory, Paris, France, 2019*, pp. 211–215, 2019.

[121] M. Kleckler and S. Mohajer, "Secure determinant codes: Type-II security," in *Proc. IEEE International Symposium on Information Theory, Los Angeles, CA, USA, 2020*, pp. 652–657, 2020.

[122] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, 2003, pp. 782–795.

[123] O. Kolosov, A. Barg, I. Tamo, and G. Yadgar, "Optimal LRC codes for all lenghts $n <= q$," *CoRR*, vol. abs/1802.00157, 2018.

[124] O. Kolosov, G. Yadgar, M. Liram, I. Tamo, and A. Barg, "On fault tolerance, locality, and optimality in locally repairable codes," *ACM Transactions on Storage*, vol. 16, no. 2, 2020, pp. 1–32.

[125] J. C. Koo and J. T. G. III, "Scalable constructions of fractional repetition codes in distributed storage systems," in *Proc. 49th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 2011*, pp. 1366–1373, 2011.

[126] O. O. Koyluoglu, A. S. Rawat, and S. Vishwanath, "Secure cooperative regenerating codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, 2014, pp. 5228–5244.

[127] K. Kralevska, D. Gligoroski, and H. Øverby, "General sub-packetized access-optimal regenerating codes," *IEEE Communications Letters*, vol. 20, no. 7, 2016, pp. 1281–1284.

[128] K. Kralevska, D. Gligoroski, R. E. Jensen, and H. Øverby, "Hash-Tag erasure codes: From theory to practice," *IEEE Transactions on Big Data*, 2017.

[129] M. N. Krishnan, A. Narayanan R, and P. V. Kumar, "Codes with combined locality and regeneration having optimal rate, $d_{\min}$ and linear field size," in *Proc. IEEE International Symposium on Information Theory, Vail, CO, USA, 2018*, pp. 1196–1200, 2018.

[130] M. N. Krishnan, N. Prakash, V. Lalitha, B. Sasidharan, P. V. Kumar, S. Narayanamurthy, R. Kumar, and S. Nandi, "Evaluation of codes with inherent double replication for Hadoop," in *Proc. 6th USENIX Workshop on Hot Topics in Storage and File Systems, Philadelphia, PA, USA*, 2014.

[131] M. N. Krishnan, B. Puranik, P. V. Kumar, I. Tamo, and A. Barg, "Exploiting locality for improved decoding of binary cyclic codes," *IEEE Trans. Commun.*, vol. 66, no. 6, 2018, pp. 2346–2358.

[132] M. N. Krishnan and P. V. Kumar, "On MBR codes with replication," in *Proc. IEEE International Symposium on Information Theory, Barcelona, Spain, 2016*, pp. 71–75, 2016.

[133] S. Kruglik, K. Nazirkhanova, and A. Frolov, "New bounds and generalizations of locally recoverable codes with availability," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, 2019, pp. 4156–4166.

[134] P. V. Kumar, "Codes with local regeneration," talk presented at the conference on *Trends in Coding Theory*, Ascona, Switzerland, Oct. 28 to Nov. 2, 2012 (joint work with G. M. Kamath, N. Prakash, V. Lalitha).

[135] V. Lalitha and S. V. Lokam, "Weight enumerators and higher support weights of maximally recoverable codes," in *Proc. 53rd Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 2015*, pp. 835–842, 2015.

[136] J. Li and B. Li, "Erasure coding for cloud storage systems: A survey," *Tsinghua Science and Technology*, vol. 18, no. 3, 2013, pp. 259–272.

[137] J. Li, X. Tang, and C. Tian, "A generic transformation for optimal repair bandwidth and rebuilding access in MDS codes," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 1623–1627, 2017.

[138] J. Li, Y. Liu, and X. Tang, "A systematic construction of MDS codes with small sub-packetization level and near-optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 67, no. 4, 2020, pp. 2162–2180.

[139] J. Li and B. Li, "Beehive: Erasure codes for fixing multiple failures in distributed storage systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, 2017, pp. 1257–1270.

[140] W. Li, Z. Wang, and H. Jafarkhani, "A tradeoff between the sub-packetization size and the repair bandwidth for Reed-Solomon code," in *Proc. 55th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 2017*, pp. 942–949, 2017.

[141] X. Li, L. Ma, and C. Xing, "Construction of asymptotically good locally repairable codes via automorphism groups of function fields," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, 2019, pp. 7087–7094.

[142] X. Li, L. Ma, and C. Xing, "Optimal locally repairable codes via elliptic curves," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, 2019, pp. 108–117.

[143] S. J. Lin, W. H. Chung, Y. S. Han, and T. Y. Al-Naffouri, "A unified form of exact-MSR codes via product-matrix frameworks," *IEEE Trans. Inf. Theory*, vol. 61, no. 2, 2015, pp. 873–886.

[144] S. Lin and W. Chung, "Novel repair-by-transfer codes and systematic exact-MBR codes with lower complexities and smaller field sizes," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, 2014, pp. 3232–3241.

[145] S. Liu and F. Oggier, "An overview of coding for distributed storage systems," *Network Coding and Subspace Designs*, 2018, pp. 363–383.

[146] S. Liu and F. E. Oggier, "On storage codes allowing partially collaborative repairs," in *Proc. IEEE International Symposium on Information Theory, Honolulu, HI, USA, 2014*, pp. 2440–2444, 2014.

[147] S. Liu and C. Xing, "Maximally recoverable local reconstruction codes from subspace direct sum systems," *CoRR*, vol. abs/2111.03244, 2021.

[148] Y. Liu, J. Li, and X. Tang, "Explicit constructions of high-rate MSR codes with optimal access property over small finite fields," *IEEE Trans. Communications*, vol. 66, no. 10, 2018, pp. 4405–4413.

[149] *Locally repairable erasure code plugin*, URL: http://docs.ceph.com/docs/master/rados/operations/erasure-code-lrc/.

[150] M. Luby, "Repair rate lower bounds for distributed storage," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, 2021, pp. 5711–5730.

[151] M. Luby, R. Padovani, T. J. Richardson, L. Minder, and P. Aggarwal, "Liquid cloud storage," *ACM Trans. Storage*, vol. 15, no. 1, 2019, 2:1–2:49.

[152] M. Luby and T. Richardson, "Distributed storage algorithms with optimal tradeoffs," *CoRR*, vol. abs/2101.05223, 2021.

[153] G. Luo and X. Cao, "Constructions of optimal binary locally recoverable codes via a general construction of linear codes," *IEEE Transactions on Communications*, vol. 69, no. 8, 2021, pp. 4987–4997.

[154] Y. Luo, C. Xing, and C. Yuan, "Optimal locally repairable codes of distance 3 and 4 via cyclic codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, 2019, pp. 1048–1053.

[155] J. Ma and G. Ge, "Optimal binary linear locally repairable codes with disjoint repair groups," *SIAM J. Discret. Math.*, vol. 33, no. 4, 2019, pp. 2509–2529.

[156] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977.

[157] J. Mardia, B. Bartan, and M. Wootters, "Repairing multiple failures for scalar MDS codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, 2019, pp. 2661–2672.

[158] U. Martínez-Peñas, "A general family of MSRD codes and PMDS codes with smaller field sizes from extended Moore matrices," *CoRR*, vol. abs/2011.14109, 2020.

[159] U. Martínez-Peñas and F. R. Kschischang, "Universal and dynamic locally repairable codes with maximal recoverability via sum-rank codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, 2019, pp. 7790–7805.

[160] M. Mehrabi and M. Ardakani, "On minimum distance of locally repairable codes," in *Proc. 15th Canadian Workshop on Information Theory, Quebec, Canada, 2017*, pp. 1–5, 2017.

[161] *Microsoft research blog: A better way to store data*, URL: https://www.microsoft.com/en-us/research/blog/better-way-store-data/.

[162] S. Mohajer and R. Tandon, "New bounds on the (n, k, d) storage systems with exact repair," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, China, 2015*, pp. 2056–2060, 2015.

[163] M. Y. Nam and H. Y. Song, "Binary locally repairable codes with minimum distance at least six based on partial t -spreads," *IEEE Communications Letters*, vol. 21, no. 8, 2017, pp. 1683–1686.

[164] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *Proc. IEEE INFOCOM, Shanghai, China, 2011*, pp. 1215–1223, 2011.

[165] O. Olmez and A. Ramamoorthy, "Fractional repetition codes with flexible repair from combinatorial designs," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, 2016, pp. 1565–1591.

[166] L. Pamies-Juarez, F. Blagojevic, R. Mateescu, C. Guyot, E. En Gad, and Z. Bandic, "Opening the chrysalis: On the real repair performance of MSR codes," in *Proc. 14th USENIX Conference on File and Storage Technologies, Santa Clara, CA, USA, 2016*, pp. 81–94, 2016.

[167] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through Hadamard designs," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, 2013, pp. 3021–3037.

[168] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, 2014, pp. 5843–5855.

[169] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Trans. on Inf. Theory*, vol. 57, no. 10, 2011, pp. 6734–6753.

[170] S. Pawar, N. Noorshams, S. El Rouayheb, and K. Ramchandran, "DRESS Codes for the storage cloud: Simple randomized constructions," in *Proc. IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 2011*, pp. 2338–2342, 2011.

[171] N. Prakash, V. Abdrashitov, and M. Médard, "The storage versus repair-bandwidth trade-off for clustered storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, 2018, pp. 5783–5805.

[172] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, "Optimal linear codes with a local-error-correction property," in *Proc. IEEE International Symposium on Information Theory Proceedings, Cambridge, MA, USA, 2012*, pp. 2776–2780, 2012.

[173] N. Prakash and M. N. Krishnan, "The storage-repair-bandwidth trade-off of exact repair linear regenerating codes for the case d=k=n-1," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, 2015*, pp. 859–863, 2015.

[174] N. Prakash, V. Lalitha, S. B. Balaji, and P. V. Kumar, "Codes with locality for two erasures," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, 2019, pp. 7771–7789.

[175] N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with locality for two erasures," in *Proc. IEEE International Symposium on Information Theory, Honolulu, HI, USA, 2014*, pp. 1962–1966, 2014.

[176] C. Rajput and M. Bhaintwal, "Optimal RS-like LRC codes of arbitrary length," *Applicable Algebra in Engineering, Communication and Computing*, vol. 31, no. 3, 2020, pp. 271–289.

[177] V. A. Rameshwar and N. Kashyap, "Achieving secrecy capacity of minimum storage regenerating codes for all feasible (n, k, d) parameter values," in *National Conference on Communications, Bangalore, India, 2019*, pp. 1–6, 2019.

[178] V. Ramkumar, M. Vajha, S. B. Balaji, M. N. Krishnan, B. Sasidharan, and P. V. Kumar, "Codes for distributed storage," in *Concise Encyclopedia of Coding Theory*, W. C. Huffman, J.-L. Kim, and P. Solé, Eds., Chapman and Hall/CRC, 2021, pp. 735–761.

[179] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. 47th Annu. Allerton Conf. Communication, Control, and Computing*, pp. 1243–1249, Urbana-Champaign, IL, 2009.

[180] K. V. Rashmi, P. Nakkiran, J. Wang, N. B. Shah, and K. Ramchandran, "Having your cake and eating it too: Jointly optimal erasure codes for I/O, storage, and network-bandwidth," in *Proc. 13th USENIX Conference on File and Storage Technologies, Santa Clara, CA, USA, 2015*, pp. 81–94, 2015.

[181] K. V. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Regenerating codes for errors and erasures in distributed storage," in *Proc. IEEE International Symposium on Information Theory, Cambridge, MA, USA, 2012*, pp. 1202–1206, 2012.

[182] K. V. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Information-theoretically secure erasure codes for distributed storage," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, 2018, pp. 1621–1646.

[183] K. V. Rashmi, N. B. Shah, D. Gu, H. Kuang, D. Borthakur, and K. Ramchandran, "A "Hitchhiker's" guide to fast and efficient data reconstruction in erasure-coded data centers," in *Proc. ACM SIGCOMM Conference, Chicago, IL, USA, 2014*, pp. 331–342, 2014.

[184] K. V. Rashmi, N. B. Shah, and K. Ramchandran, "A piggybacking design framework for read-and download-efficient distributed storage codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, 2017, pp. 5802–5820.

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

[186] N. Raviv, N. Silberstein, and T. Etzion, "Constructions of high-rate minimum storage regenerating codes over small fields," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, 2017, pp. 2015–2038.

[187] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, 2014, pp. 212–236.

[188] A. S. Rawat, "Secrecy capacity of minimum storage regenerating codes," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 1406–1410, 2017.

[189] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, 2013, pp. 212–236.

[190] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Progress on high-rate MSR codes: Enabling arbitrary number of helper nodes," in *Proc. Information Theory and Applications Workshop, La Jolla, CA, USA, 2016*, pp. 1–6, 2016.

[191] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Centralized repair of multiple node failures with applications to communication efficient secret sharing," *IEEE Trans. Inf. Theory*, vol. 64, no. 12, 2018, pp. 7529–7550.

[192] A. S. Rawat, A. Mazumdar, and S. Vishwanath, "Cooperative local repair in distributed storage," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, 2015, pp. 1–17.

[193] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Trans. Inf. Theory*, vol. 62, no. 8, 2016, pp. 4481–4493.

[194] A. S. Rawat, I. Tamo, V. Guruswami, and K. Efremenko, "ϵ-MSR codes with small sub-packetization," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 2043–2047, 2017.

[195] A. S. Rawat, I. Tamo, V. Guruswami, and K. Efremenko, "MDS code constructions with small sub-packetization and near-optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, 2018, pp. 6506–6525.

[196] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the SIAM*, vol. 8, no. 2, 1960, pp. 300–304.

[197] C. Salgado, A. Várilly-Alvarado, and J. F. Voloch, "Locally recoverable codes on surfaces," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, 2021, pp. 5765–5777.

[198] B. Sasidharan, K. Senthoor, and P. V. Kumar, "An improved outer bound on the storage-repair-bandwidth tradeoff of exact-repair regenerating codes," in *Proc. IEEE International Symposium on Information Theory, Honolulu, HI, USA, 2014*, pp. 2430–2434, 2014.

[199] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "Codes with hierarchical locality," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, China, 2015*, pp. 1257–1261, 2015.

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

[201] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "Codes with hierarchical locality," *CoRR*, vol. abs/1501.06683, 2015.

[202] B. Sasidharan and P. V. Kumar, "High-rate regenerating codes through layering," in *Proc. IEEE International Symposium on Information Theory, Istanbul, Turkey, 2013*, pp. 1611–1615, 2013.

[203] B. Sasidharan, N. Prakash, M. N. Krishnan, M. Vajha, K. Senthoor, and P. V. Kumar, "Outer bounds on the storage-repair bandwidth trade-off of exact-repair regenerating codes," *Int. Journal Inf. Coding Theory*, vol. 3, no. 4, 2016, pp. 255–298.

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

[205] B. Sasidharan, M. Vajha, and P. V. Kumar, "An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair," *CoRR*, vol. abs/1607.07335, 2016.

[206] B. Sasidharan, M. Vajha, and P. V. Kumar, "An explicit, coupled-layer construction of a high-rate regenerating code with low sub-packetization level, small field size and d<(n-1)," *CoRR*, vol. abs/1701.07447, 2022.

[207] M. Sathiamoorthy, M. Asteris, D. S. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," *PVLDB*, vol. 6, no. 5, 2013, pp. 325–336.

[208] S. Schechter, "On the inversion of certain matrices," *Mathematical Tables and Other Aids to Computation*, vol. 13, no. 66, 1959, pp. 73–77.

[209] B. Segre, "Curve razionali normali ek-archi negli spazi finiti," *Annali di Matematica Pura ed Applicata*, vol. 39, no. 1, 1955, pp. 357–379.

[210] K. Senthoor, B. Sasidharan, and P. V. Kumar, "Improved layered regenerating codes characterizing the exact-repair storage-repair bandwidth tradeoff for certain parameter sets," in *Proc. IEEE Information Theory Workshop, Jerusalem*, pp. 1–5, 2015.

[211] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, 2012, pp. 1837–1852.

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

[213] N. B. Shah, "On minimizing data-read and download for storage-node recovery," *IEEE Communications Letters*, vol. 17, no. 5, 2013, pp. 964–967.

[214] M. Shahabinejad, M. Khabbazian, and M. Ardakani, "A class of binary locally repairable codes," *IEEE Transactions on Communications*, vol. 64, no. 8, 2016, pp. 3182–3193.

[215] K. Shanmugam, D. S. Papailiopoulos, A. G. Dimakis, and G. Caire, "A Repair framework for scalar MDS codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, 2014, pp. 998–1007.

[216] S. Shao, T. Liu, C. Tian, and C. Shen, "On the tradeoff region of secure exact-repair regenerating codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, 2017, pp. 7253–7266.

[217] D. Shivakrishna, V. A. Rameshwar, V. Lalitha, and B. Sasidharan, "On maximally recoverable codes for product topologies," in *Proc. Twenty Fourth National Conference on Communications*, IEEE, pp. 1–6, 2018.

[218] K. W. Shum and Y. Hu, "Cooperative regenerating codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, 2013, pp. 7229–7258.

[219] N. Silberstein and A. Zeh, "Optimal binary locally repairable codes via anticodes," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, 2015*, pp. 1247–1251, 2015.

[220] N. Silberstein, "Optimal locally repairable codes via rank-metric codes," talk presented at the conference on *Trends in Coding Theory*, Ascona, Switzerland, Oct. 28 to Nov. 2, 2012 (joint work with A. S. Rawat and S. Vishwanath).

[221] N. Silberstein and T. Etzion, "Optimal fractional repetition codes based on graphs and designs," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, 2015, pp. 4164–4180.

[222] N. Silberstein and A. Zeh, "Anticode-based locally repairable codes with high availability," *Designs, Codes and Cryptography*, vol. 86, Feb. 2018.

[223] R. Singleton, "Maximum distance q-nary codes," *IEEE Trans. Inf. Theory*, vol. 10, no. 2, 1964, pp. 116–118.

[224] J.-Y. Sohn, B. Choi, S. W. Yoon, and J. Moon, "Capacity of clustered distributed storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, 2019, pp. 81–107.

[225] W. Song, S. H. Dau, C. Yuen, and T. J. Li, "Optimal locally repairable linear codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, 2014, pp. 1019–1036.

[226] W. Song, K. Cai, C. Yuen, K. Cai, and G. Han, "On sequential locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, 2018, pp. 3513–3527.

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

[228] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, 2014, pp. 4661–4676.

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

[230] I. Tamo, A. Barg, and A. Frolov, "Bounds on the parameters of locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, 2016, pp. 3070–3083.

[231] I. Tamo, A. Barg, S. Goparaju, and A. R. Calderbank, "Cyclic LRC codes, binary LRC codes, and upper bounds on the distance of cyclic codes," *Int. J. Inf. Coding Theory*, vol. 3, no. 4, 2016, pp. 345–364.

[232] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis, "Optimal locally repairable codes and connections to matroid theory," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, 2016, pp. 6661–6671.

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

[234] I. Tamo, M. Ye, and A. Barg, "The repair problem for Reed–Solomon codes: Optimal repair of single and multiple erasures with almost optimal node size," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, 2018, pp. 2673–2695.

[235] R. Tandon, S. Amuru, T. C. Clancy, and R. M. Buehrer, "Toward optimal secure distributed storage systems with exact repair," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, 2016, pp. 3477–3492.

[236] C. Tian, B. Sasidharan, V. Aggarwal, V. Vaishampayan, and P. V. Kumar, "Layered exact-repair regenerating codes via embedded error correction and block designs," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, 2015, pp. 1933–1947.

[237] C. Tian, "Characterizing the rate region of the (4, 3, 3) exact-repair regenerating codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, 2014, pp. 967–975.

[238] C. Tian, "A note on the rate region of exact-repair regenerating codes," *CoRR*, vol. abs/1503.00011, 2015.

[239] M. Vajha, B. S. Babu, and P. V. Kumar, "Explicit MSR codes with optimal access, optimal sub-packetization and small field size for $d = k+1, k+2, k+3$," in *Proc. IEEE International Symposium on Information Theory, Vail, CO, USA, 2018*, pp. 2376–2380, 2018.

[240] M. Vajha, V. Ramkumar, B. Puranik, G. R. Kini, E. Lobo, B. Sasidharan, P. V. Kumar, A. Barg, M. Ye, S. Narayanamurthy, S. Hussain, and S. Nandi, "Clay codes: Moulding MDS codes to yield an MSR Code," in *Proc. 16th USENIX Conference on File and Storage Technologies, Oakland, CA, USA, 2018*, pp. 139–154, 2018.

[241] A. Wang and Z. Zhang, "Repair locality with multiple erasure tolerance," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, 2014, pp. 6979–6987.

[242] A. Wang and Z. Zhang, "An integer programming-based bound for locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, 2015, pp. 5280–5294.

[243] A. Wang, Z. Zhang, and D. Lin, "Bounds and constructions for linear locally repairable codes over binary fields," in *Proc. IEEE International Symposium on Information Theory, Aachen, Germany, 2017*, pp. 2033–2037, 2017.

[244] A. Wang and Z. Zhang, "Exact cooperative regenerating codes with minimum-repair-bandwidth for distributed storage," in *Proc. IEEE INFOCOM, Turin, Italy, 2013*, pp. 400–404, 2013.

[245] A. Wang, Z. Zhang, and D. Lin, "Two classes of (r, t)-locally repairable codes," in *Proc. IEEE International Symposium on Information Theory, Barcelona, Spain, 2016*, pp. 445–449, 2016.

[246] A. Wang, Z. Zhang, and M. Liu, "Achieving arbitrary locality and availability in binary codes," in *Proc. IEEE International Symposium on Information Theory, Hong Kong, 2015*, pp. 1866–1870, 2015.

[247] G. Wang, M.-Y. Niu, and F.-W. Fu, "Constructions of (r,t)-LRC based on totally isotropic subspaces in symplectic space over finite fields," *International Journal of Foundations of Computer Science*, vol. 31, Apr. 2020, pp. 327–339.

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

[249] Z. Wang, I. Tamo, and J. Bruck, "Long MDS codes for optimal repair bandwidth," in *Proc. IEEE International Symposium on Information Theory, Cambridge, MA, USA, 2012*, pp. 1182–1186, 2012.

[250] V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 5, 1991, pp. 1412–1418.

[251] Y. Wu, "Existence and construction of capacity-achieving network codes for distributed storage," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, 2010, pp. 277–288.

[252] E. Yavari and M. Esmaeili, "Locally repairable codes: Joint sequential–parallel repair for multiple node failures," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, 2020, pp. 222–232.

[253] F. Ye, K. W. Shum, and R. W. Yeung, "The rate region for secure distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, 2017, pp. 7038–7051.

[254] M. Ye and A. Barg, "Explicit constructions of MDS array codes and RS codes with optimal repair bandwidth," in *Proc. IEEE International Symposium on Information Theory, Barcelona, Spain, 2016*, pp. 1202–1206, 2016.

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

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

[257] M. Ye and A. Barg, "Cooperative repair: Constructions of optimal MDS codes for all admissible parameters," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, 2018, pp. 1639–1656.

[258] R. W. Yeung, "A framework for linear information inequalities," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, 1997, pp. 1924–1934.

[259] A. Zeh and E. Yaakobi, "Optimal linear and cyclic locally repairable codes over small fields," in *Proc. IEEE Information Theory Workshop, Jerusalem, Israel, 2015*, pp. 1–5, 2015.

[260] G. Zhang and H. Liu, "Constructions of optimal codes with hierarchical locality," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, 2020, pp. 7333–7340.

[261] J. Zhang, X. Wang, and G. Ge, "Some improvements on locally repairable codes," *CoRR*, vol. abs/1506.04822, 2015.

[262] M. Zhang and R. Li, "Two families of LRCs with availability based on iterative matrix," in *Proc. 13th International Symposium on Computational Intelligence and Design, Hangzhou, China, 2020*, pp. 334–337, 2020.

[263] L. Zhou and Z. Zhang, "Explicit construction of minimum bandwidth rack-aware regenerating codes," *CoRR*, vol. abs/2103.01533, 2021.

[264] B. Zhu, K. W. Shum, H. Li, and H. Hou, "General fractional repetition codes for distributed storage systems," *IEEE Commun. Lett.*, vol. 18, no. 4, 2014, pp. 660–663.

[265] M. Zorgui and Z. Wang, "Centralized multi-node repair regenerating codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, 2019, pp. 4180–4206.

Index