On codes with the locality property

Alexander Barg and Itzhak Tamo

***Abstract*—We consider codes that support the local recovery property of each code symbol (LRC codes). Codes of this type were extensively studied in recent years because of their applications in distributed storage systems. We discuss algebraic constructions of LRC codes over small alphabets that attain the best possible distance-locality tradeoff and their extensions to cyclic codes and codes on algebraic curves. We also discuss examples of practical LRC codes used in large-scale storage systems and point out some open questions in this area.**

I. Introduction

Distributed and cloud storage systems have reached such a massive scale that recovery from several node failures is now part of regular operation of the system rather than a rare exception. To support reliable storage, system designers have turned to error correcting codes, introducing redundancy to recover the temporarily or permanently unavailable data. The simplest and to date the most frequently employed solution is to replicate the data several times, writing the copies of each data fragment to distinct physical locations. For example, *Apache Hadoop*, an open source software for distributed storage, uses a default method of 3-way replication. Another common solution, based on Reed-Solomon (RS) codes, provides stronger protection for the same or smaller storage overhead. For instance, the file systems of Facebook and Google use the (14,10) and (9,6) RS codes, respectively. RS codes have been also standardized as a part of the well-known RAID 6 data protection technology.

New challenges in the development of distributed storage systems are to a large extent driven by the exponential growth of the amount of stored data which makes exabyte data volumes today's new reality. One of the new tasks faced by such systems, but not addressed by current solutions, is recovery from a single node failure. Studies show that, although several concurrent failures are possible, and therefore the system should be able to protect against them, the most common scenario is the failure of a single node. Therefore, constructing codes that optimize the repair of a single node becomes an important problem for coding theorists and developers alike.

Recovery of the information stored on a single node, or the *repair problem*, can be carried out successfully because of the redundancy inserted in the information at the time of writing to the memory. The efficiency of the data repair can be measured in several ways. One of them, introduced in the foundational paper [4], proposes to optimize the amount of data transmitted in the system to accomplish the repair. This metric has become known as *repair bandwidth*. The second measure, called *locality*, is related to the total number of nodes accessed during the data recovery [8], [9], [7]. Both metrics have their own merits, and choosing between them is related to the type of the storage system and the underlying scope of applications. In this paper we focus on codes with locality, i.e., codes that in the course of repair of a single node access only a small number of other nodes.

An (n,k,r) locally recoverable (LRC) code encodes k data symbols into n symbols in such a way that the value of any symbol of the encoding can be found by accessing at most r other stored symbols. For example, a code of length $n = 2k$ in which every data symbol is repeated twice, is an LRC code with locality $r = 1$. As another extreme, consider an (n,k) MDS code with locality $r = k$ in which not only one symbol, but the entire encoding can be found by accessing k codeword symbols. Generally the value of locality r satisfies $1 \leq r \leq k$. Yet another simple example is provided by regular LDPC codes with $r+1$ nonzeros in every check equation, meaning that every single symbol of the codeword is a linear combination of some other r symbols. The study of LRC codes forms a new topic in coding theory that gives rise to questions ranging from limits to the maximum size of LRC codes to the constructions and structure of codes and their decoding algorithms. For instance, MDS codes which are optimal for the classical error/erasure correction problem, are far from being optimal in terms of locality because the repair task requires access to a large number of code symbols.

Bounds and constructions of LRC codes have been studied in a number of recent papers. A natural question to ask is as follows: given an (n,k,r) LRC code $\mathcal{C}$, what is the largest possible minimum distance $d(\mathcal{C})$? A useful generalization of the Singleton bound [7], discussed in Section II-A, Eq. (3) below, gave rise to both studies into code bounds and constructions of RS-type codes that form the main topic of this paper. While the LRC Singleton bound, like its classic counterpart, is independent of the code alphabet, another work [3] introduced a bound on the code's distance that accounts for the alphabet size, and more results of this kind appear in the recent paper [17].

Codes whose parameters satisfy the LRC Singleton bound with equality, are called *optimal LRC codes* in the literature. Among the constructions of LRC codes we note the results of [15], [19], [6] that combine some known code families to account for the LRC property. While these constructions are optimal by their parameters, they rely on alphabets of a large size, limiting their usefulness in applications.

Coding for distributed storage is currently an active research area. Codes that optimize the repair bandwidth and codes with locality appear in a large number of publications, too numerous to cite or overview in this article. In [16] we initiated a line of research in this area that begins with a construction of RS-type codes with the locality property and extends to constructions of cyclic codes and codes on algebraic curves, as well as to a study into bounds on the parameters of LRC codes. In this paper we present and discuss this work, apologizing to our many colleagues whose contributions to this area we did not have a chance to mention.

II. Codes with the locality constraint (LRC codes)

We say that a code has locality r if the value of every coordinate of the codeword c is uniquely determined by the values of at most r other coordinates of c. In the context of storage applications, this enables the system to recover the data from a dysfunctional node by accessing at most r other nodes in the storage cluster. At the same time, if a group of more than one nodes become inaccessible, we would still like to be able to restore the data using the remaining storage nodes. In this case, it may not be possible to recover the missing symbols in a local fashion, but we would like to be able to recover them nevertheless by accessing the remaining available symbols of the codeword. Taken together, these conditions call for constructing codes with small locality and large distance d. A formal definition of an LRC code is as follows.

Definition 1: A code $\mathcal{C}$ of length n over a finite alphabet $\mathcal{Q}$ is said to have *locality* r if for every $i \in [n]$ there exists a subset $\mathcal{R}_i \subset [n]\backslash\{i\}, |\mathcal{R}_i| \le r$ and a function ϕ_i such that for every codeword $c \in \mathcal{C}$

$$c_i = \phi_i(\{c_j, j \in \mathcal{R}_i\}). \tag{1}$$

As already remarked, simple examples of LRC codes are obtained by concatenating several copies of some code. For instance, replicating m times a single-parity-check code of length $r+1$, we obtain an $(m(r+1), mr, r)$ LRC code with distance $d = 2$, and repeating twice an $(n/2, k)$ MDS code yields an $(n, k, 1)$ LRC code with distance $d = 2(n/2 - k + 1)$.

Let us give a less trivial example of an LRC code. This example relies on the main LRC code construction discussed in the paper.

Example 1: We will construct an $(n = 9, k = 4, r = 2)$ LRC code $\mathcal{C}$ with distance $d = 5$, choosing $\mathbb{F}_{13}$ to be the code alphabet. Consider the space of polynomials

$$\mathcal{P} = \{f_a(x) = a_0 + a_1x + a_3x^3 + a_4x^4\},$$

where $a = (a_0, a_1, a_3, a_4) \in \mathbb{F}_{13}^4$ denotes the message vector (the omission of x^2 is intended). Consider the linear code

$$\mathcal{C} = \{ev_A(f), f \in \mathcal{P}\},$$

defined by the set of points $A = \{1, 3, 9, 2, 6, 5, 4, 12, 10\}$ and the evaluation map $ev_A : \mathbb{F}_q[x] \to \mathbb{F}_q^n$ given as $ev_A(f) = (f(a), a \in A)$. For instance, taking $a = (1, 1, 1, 1)$, we evaluate the polynomial $f_a(x) = 1 + x + x^3 + x^4$ to find the codeword

$$c := ev_A(f_a) = (4, 8, 7, 1, 11, 2, 0, 0, 0). \tag{2}$$

Since the degree of $f_a(x)$ is at most 4, the distance of the code satisfies $d(\mathcal{C}) \ge 5$. It will be argued later that 5 is the maximum possible distance for any $(9, 4, 2)$ LRC code, so the code $\mathcal{C}$ is optimal. Note that an RS code with $n = 9$ and $k = 4$ has distance 6 which is only one greater than the distance of the code $\mathcal{C}$. Therefore by reducing the distance by one we managed to decrease the locality by a factor of two.

Although the code $\mathcal{C}$ is a subset of a $(n = 9, k = 5)$ RS code[1], we emphasize the special choice of the space $\mathcal{P}$ and the set A which account for the locality property of the code. Indeed, regardless of the exact values of the entries of the information vector a, there is a linear polynomial that passes through the points $f_a(1), f_a(3)$ and $f_a(9)$ of the codeword c. For instance, the polynomial $\delta_1(x) = a_0 + a_3 + (a_1 + a_4)x$ satisfies $\delta_1(i) = f_a(i), i = 1, 3, 9$, and in a similar way, $\delta_2(x) = a_0 + 8a_3 + (a_1 + 8a_4)x$ passes through the coordinates with locations $2, 6$, and 5. It is also possible to construct a linear polynomial $\delta_3(x)$ that passes through the locations $4, 12$, and 10. This property supports local recovery of any one symbol. Indeed, if the value $f_a(1)$ is unavailable, we can compute $\delta_1(x)$ from its values $\delta_1(3), \delta_1(9)$ and find $f_a(1) = \delta_1(1)$. For instance, for the codeword c in (2) we obtain $\delta_1(x) = 2x + 2$ and find the correct value $\delta_1(1) = 4$. This procedure is schematically shown in Fig. 1.

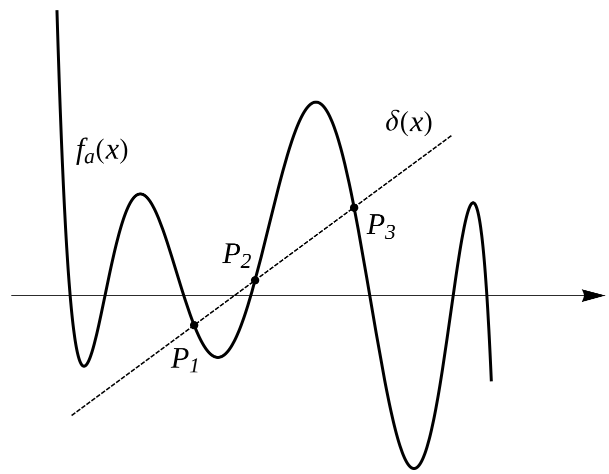

Fig. 1: Local recovery by polynomial interpolation

The local recovery described constitutes a saving compared to the standard decoding of RS codes which calls for computing the polynomial f_a of degree 4 from some of its 5 values. Note also a special property of the construction: the described linear polynomials pass through 3 points of the graph of f_a, which is one point more than is guaranteed by the general interpolation. That this becomes possible is an artifact of the special choice of the polynomial space $\mathcal{P}$ and the set of points A.

A. General Construction of Optimal LRC Codes

There are several classical bounds on the distance of the code in terms of its length and dimension. One of them is the Singleton bound, and a code that meets it is called an MDS code. Moreover, the MDS conjecture (partially proved recently in [1]) claims that, loosely speaking, in order to attain the Singleton bound, the code alphabet has to be of the order

[1]The RS code is obtained by evaluating *all the polynomials* of degree $\deg f \le k - 1 = 4$.

of the length of the code. The Singleton bound was extended to codes with locality in [7] which showed that the distance $d(\mathcal{C})$ of an (n,k,r) LRC code $\mathcal{C}$ is bounded by

$$d(\mathcal{C}) \le n-k-\left\lceil \frac{k}{r} \right\rceil + 2. \qquad (3)$$

An LRC code whose parameters meet this bound with equality is called optimal. Taking $r=k$ in (3), we recover the Singleton bound, so any (n,k) RS code is an optimal (n,k,k) LRC code. Likewise, the subcode of an RS code constructed in Example 1 is also an optimal LRC code. This suggests that to construct optimal LRC codes for a broad range of the parameters n,k,r, it suffices to take an alphabet of size q comparable to n, and RS codes and their subcodes form natural candidates for optimal LRC codes. We will show that this is indeed the case by providing such a construction which we call an RS-type LRC code.

In Example 1 we implicitly defined a partition of the set of locations into subsets $A_1=\{1,3,9\}, A_2=\{2,6,5\}, A_3=\{4,12,10\}$ such that for each of them, there is a linear polynomial that passes through all the codeword coordinates in these locations. Building on this intuition, let us take a subset $A \subset \mathbb{F}_q$ of n points that label the coordinates of the code. Suppose that there is a partition $\mathcal{A}=\{A_1,...,A_{\frac{n}{r+1}}\}$ of the set A into $n/(r+1)$ subsets of size $r+1$ and that there exists a polynomial $g(x)$ of degree $r+1$ such that g is constant on the blocks of the partition, i.e.,

$$g(\alpha)=g(\beta) \text{ for any } \alpha,\beta \in A_i, i=1,\dots,n/(r+1). \qquad (4)$$

We aim at constructing a linear k-dimensional code $\mathcal{C}:\mathbb{F}_q^k \to \mathbb{F}_q^n$. Given a vector $a \in \mathbb{F}_q^k, a=(a_{ij}, i=0,\dots,r-1, j=0,\dots,\frac{k}{r}-1)$, define the polynomial[2]

$$f_a(x)=\sum_{i=0}^{r-1} x^i \sum_{j=0}^{\frac{k}{r}-1} a_{ij} g(x)^j. \qquad (5)$$

and note that $\deg f_a \le k+\frac{k}{r}-2$.

Definition 2: Let $\mathcal{P}$ be the set of polynomials of the form (5) and define the code

$$\mathcal{C}=\{ev_A(f_a), f_a \in \mathcal{P}\}. \qquad (6)$$

The subsets A_i are called *recovery sets*. Once we specify a location α such that $A_i \ni \alpha$, the subset $A_i \backslash \{\alpha\}$ is called the recovery set of α. The main result about this code family is as follows.

Theorem 2.1: The code $\mathcal{C}$ defined in (6) is an optimal (n,k,r) LRC code. The local recovery of the symbol in location α is accomplished by computing a polynomial $\delta(x)$ of degree $r-1$ that passes through all the points of the recovery set of this location.

Sketch of the proof: The distance of $\mathcal{C}$ equals n minus the maximum number of zeros of $f_a(x)$, and is seen to meet the bound (3). The claim that $\dim \mathcal{C}=k$ becomes obvious once we observe that the k polynomials $g(x)^j x^i$ are all of different degrees and therefore span a k-dimensional subspace of $\mathbb{F}_q[x]$. Furthermore, the polynomials f_a are evaluated at $n>k$ distinct points of the field, therefore the evaluation mapping (6) is injective and the code is of dimension k.

The local recovery is accomplished as follows. Given the erased location $\alpha \in A_i$, find the unique polynomial $\delta(x)$ of degree at most $r-1$ that intersects the graph of $f_a(x)$ at all the other r points of the set A_i :

$$\delta(\beta)=f_a(\beta),\ \beta \in A_i \backslash \{\alpha\}.$$

Note that $g(x)$ is constant on A_i, and therefore $\delta(x)$ is the polynomial

$$\delta(x)=\sum_{i=0}^{r-1} x^i \sum_{j=0}^{\frac{k}{r}-1} a_{ij} g(\alpha)^j.$$

Hence, the symbol at location α equals to $\delta(\alpha)=f_a(\alpha)$. ■

Let us make a few observations about the features of the code family.

(i) CONSTRUCTING $g(x)$: The main ingredient of the construction is the polynomial $g(x)$ whose existence is a priori not so obvious. It is not difficult to prove by counting that the required $g(x)$ exists, but we would like to be able to construct it efficiently. This question will be discussed in the next subsection, and it will also enable us to establish relations between the code length n and the size of the alphabet q. The property that $g(x)=\text{const}$ on $A_j, 1 \le j \le n/(r+1)$ also has a natural geometric interpretation which provides a segue to constructing LRC codes on algebraic curves (more on this in Sect. IV-B below).

(ii) DIVISIBILITY ASSUMPTIONS: Both the assumptions $r|k$ and $(r+1)|n$ can be removed. To lift the first one, we simply modify the polynomials $f_a(x)$ by taking the inner sum in (5) to go to $\lfloor \frac{k}{r} \rfloor$ or $\lfloor \frac{k}{r} \rfloor - 1$ depending on whether $i < k \bmod r$ or not. As a result, the properties of the code do not change; in particular, it remains optimal. To construct codes of arbitrary length n, removing the constraint $(r+1)|n$, we take the last recovery set to be of a smaller size as needed. Most properties of the code again do not change, although its distance can be one less than the optimal value given by (3).

(iii) LRC REED-SOLOMON CODES: The codes introduced in this section form a direct extension of the classical *Reed-Solomon* codes; in particular, the code $\mathcal{C}$ is a k-dimensional subcode of an $(n, k+\frac{k}{r}-1)$ RS code. Our construction also reduces to Reed-Solomon codes if r is taken to be k. Indeed, in this case the inner sum in (5) reduces to one term, so $g(x)$ is removed, and we recover the classical definition. Moreover, the set A in this case can be an arbitrary subset of $\mathbb{F}_q$, while the locality condition for $r<k$ imposes a restriction on the choice of the locations.

(iv) SYSTEMATIC ENCODING: Any linear code can be represented in a systematic way, but the described construction can be modified to make this systematic representation explicit and presented in algebraic terms. For $i=1,...,k/r$ let $B_i=\{\beta_{i,1},...,\beta_{i,r}\}$ be some subset of A_i of size r. For each set B_i define r polynomials $\phi_{i,j}, j=1,...,r$ of degree less than r such that $\phi_{i,j}(\beta_{i,l})=\delta_{j,l}$, and similarly define

[2]We assume that both $\frac{n}{r+1}$ and $\frac{k}{r}$ are integer numbers and comment on the other possibilities in the remarks below.

$m = n/(r+1)$ polynomials $f_i(x)$ such that $f_i(A_j) = \delta_{i,j}$. For k information symbols $a = (a_{i,j}, i = 1, \dots, k/r; j = 1, \dots, r)$ construct the polynomial

$$f_a(x) = \sum_{i=1}^{k/r} f_i(x) \Big(\sum_{j=1}^{r} a_{i,j} \phi_{i,j}(x) \Big). \tag{7}$$

Define the evaluation code $\mathcal{C}^{(\text{sys})} := \{ev_A(f_a), f_a \in \mathcal{P}\}$ where $\mathcal{P}$ is the set of all polynomials of the form (7). It is easily seen that this code is systematic, and the message symbols are written in the locations of the sets B_i.

A useful general view of these remarks as well as of the code construction itself is related to the study of properties of the polynomial algebra $\mathbb{F}_{\mathcal{A}}[x]$ spanned by the polynomials constant on the blocks of the partition $\mathcal{A}$. This approach and its connections to the code construction are further developed in [16].

B. Piecewise-constant polynomials

In this section we show how to construct a partition $\mathcal{A}$ of $A \subseteq \mathbb{F}_q$ and a polynomial $g(x)$ of degree $r+1$ that is constant on the blocks of the partition. Let $\mathbb{F}_q^*$ and $\mathbb{F}_q^+$ denote the multiplicative and the additive groups of $\mathbb{F}_q$ respectively. The main idea is expressed in the following simple observation.

Proposition 2.2: Let H be a subgroup of $\mathbb{F}_q^*$ or $\mathbb{F}_q^+$. The annihilator polynomial of the subgroup

$$g(x) = \prod_{h \in H} (x - h) \tag{8}$$

is constant on each coset of H.

Proof: Assume that H is a multiplicative subgroup and let $a, a\bar{h}$ be two elements of the coset aH, where $\bar{h} \in H$, then

$$\begin{aligned} g(a\bar{h}) = \prod_{h \in H} (a\bar{h} - h) &= \bar{h}^{|H|} \prod_{h \in H} (a - h\bar{h}^{-1}) \\ &= \prod_{h \in H} (a - h) \\ &= g(a). \end{aligned}$$

The proof for additive subgroups is completely analogous. ■

If H is a multiplicative subgroup of $\mathbb{F}_q^*$, then $g(x)$ in (8) can be written as $g(x) = x^{|H|} - 1$. Equivalently, we can take $g(x) = x^{|H|}$. Accordingly, the code length n can be any multiple of $r+1$ satisfying $n \le q-1$ (or $n \le q$ in the case of the additive group). In Example 1 we made the following choices: (i) H is the group of cube roots of unity modulo 13, (ii) $A = H \cup 2H \cup 4H$ a union of three cosets (note that we can take any three cosets of the full set of cosets), and (iii) $g(x) = x^3$ (instead of $g(x) = x^3 - 1$).

Example 2: In this example we construct an optimal $(12, 6, 3)$ LRC code with distance $d = 6$ using the additive group of the field. Let α be a primitive element of the field $\mathbb{F}_{2^4}$ and take the additive subgroup $H = \{x + y\alpha : x, y \in \mathbb{F}_2\}$. The polynomial $g(x)$ in (8) equals

$$\begin{aligned} g(x) &= x(x+1)(x+\alpha)(x+\alpha+1) \\ &= x^4 + (\alpha^2 + \alpha + 1)x^2 + (\alpha^2 + \alpha)x. \end{aligned}$$

Let A be the union of any 3 out of the 4 cosets of H. For $a = (a_{i,j}, i = 0, 1, 2;\ j = 0, 1) \in \mathbb{F}_{2^4}^6$ let

$$f_a(x) = \sum_{i=0}^{2} (a_{i,0} + a_{i,1} g(x)) x^i.$$

Constructing a code $\mathcal{C}$ by evaluating the polynomials $f_a(x)$ at the points of A, we obtain an LRC code with locality $r = 3$. Note that any $(12, 6)$ MDS code over $\mathbb{F}_{2^4}$ has minimum distance $d = 7$ and locality $r = 6$. The distance of the code $\mathcal{C}$ is only one less than that, but at the same time the locality is decreased by a factor of two, from 6 to 3.

The method described above gives a way of constructing piecewise-constant polynomials, while at the same time constraining the possible values of the code length due to the natural divisibility constraints. We conclude by noting that the additive and multiplicative structures of the field can be combined into a more general method of constructing the polynomials, increasing the range of options for the code length [16, Section III.B].

III. Extensions: Multiple recovery sets; Correcting more than one erasure

A. Algebraic LRC codes with multiple recovery sets

In distributed storage applications there are fragments of the data that are accessed more often than the remaining contents (they are called "hot data"). In the case that such fragments are accessed simultaneously by many users of the system, it may be desirable to ensure that every symbol has several *disjoint recovery sets*, increasing the instantaneous availability of the data.

Using this as a motivation, let us generalize the definition of LRC codes as follows. A code over the alphabet $\mathcal{Q}$ is said to be *locally recoverable with two recovery sets* (an LRC(2) code) if for every $i \in \{1, \dots, n\}$ there exist disjoint subsets $\mathcal{R}_{i,1}, \mathcal{R}_{i,2} \subset [n] \backslash \{i\}$ and functions $\phi_{i,j}, j = 1, 2$ such that for every codeword $c \in \mathcal{C}$

$$c_i = \phi_{i,j}(c_\ell, \ell \in \mathcal{R}_{i,j}), \quad j = 1, 2. \tag{9}$$

Suppose that $|\mathcal{R}_{i,1}| \le r_1, |\mathcal{R}_{i,2}| \le r_2$ for all i (we do not assume that $r_1 = r_2$). We write the parameters of an LRC(2) code of dimension k as $(n, k, \{r_1, r_2\})$.

Among the obvious ways to construct LRC(2) codes are various two-level constructions such as product codes or codes on bipartite graphs. We focus on algebraic constructions, extending the approach of the previous section to multiple recovery sets.

Suppose that $\mathcal{A}_1$ ($\mathcal{A}_2$) is a partition of a set $A \subset [n]$ into subsets of size $r_1 + 1$ (resp., $r_2 + 1$). Call the partitions $\mathcal{A}_1, \mathcal{A}_2$ *orthogonal* if

$$|A_{1,i} \cap A_{2,j}| \le 1 \quad \text{for all } A_{1,i} \in \mathcal{A}_1, A_{2,j} \in \mathcal{A}_2.$$

If the partitions $\mathcal{A}_1$ and $\mathcal{A}_2$ are orthogonal, then it is possible to construct a code in which every symbol has two *disjoint* recovery sets of size r_1 and r_2. The construction relies on polynomial evaluation and is very similar to the construction of Section II-A. To give an example, consider the field $\mathbb{F}_{16}$. Its additive group $\mathbb{F}_{16}^+$ contains several pairs of subgroups $G \cong H \cong (\mathbb{Z}_2)^2$ such that $G \cap H = 0$. For instance, take $G = \{0, 1, \alpha, \alpha^4\}$ and $H = \{0, \alpha^2, \alpha^3, \alpha^6\}$, where α is a primitive element that satisfies $\alpha^4 = \alpha + 1$. The subgroups G and H define a pair of orthogonal partitions of $\mathbb{F}_{16}^+$ given by

$$\mathcal{A}_G = \{G, \alpha^5 + G, \alpha^6 + G, \alpha^7 + G\}$$
$$\mathcal{A}_H = \{H, 1 + H, \alpha + H, \alpha^4 + H\}.$$

Using each of these partitions, we can construct an LRC code $\mathcal{C}$ with the parameters $(n = 16, k, \{r_1 = 3, r_2 = 3\})$ of dimension $k, 1 \le k \le 8$. Every coordinate of the codeword can be recovered in two independent ways: for instance, the coordinate c_α is found by computing the polynomial $\delta_1(x)$ of degree at most 2 that passes through the points c_0, c_1, c_{α^4} *as well as* the polynomial $\delta_2(x)$ that passes through $c_{\alpha^5}, c_{\alpha^9}, c_{\alpha^{11}}$. Then we have $c_\alpha = \delta_1(\alpha) = \delta_2(\alpha)$.

It is easy to identify a necessary and sufficient condition for two subgroups to generate orthogonal partitions.

Proposition 3.1: Let H and G be two subgroups of a finite group X, then the coset partitions $\mathcal{H}$ and $\mathcal{G}$ defined by H and G respectively are orthogonal iff the subgroups intersect trivially, namely

$$H \cap G = 1.$$

If the group X is cyclic, then it is equivalent to requiring that $\gcd(|H|, |G|) = 1$.

In the context of finite fields we can use both the multiplicative and the additive group of the field to construct LRC(2) codes. It is also easy to find several subgroups that intersect trivially; in particular, this is clearly possible for the additive group $\mathbb{F}_q^+$ in the case of a non-prime q. At the same time, constructing LRC(2) codes from a multiplicative subgroup of $\mathbb{F}_q, q = p^l$ requires one extra condition, namely, that $q - 1$ is not a power of a prime. In this case, we can find two subgroups of $\mathbb{F}_q^*$ of coprime orders that give rise to orthogonal partitions of $\mathbb{F}_q^*$.

Proposition 3.2: Let $\mathbb{F}_q$ be a finite field such that the $q - 1$ is not a power of a prime. Let $r_1, r_2 > 1, gcd(r_1, r_2) = 1$ be two factors of $q - 1$. Then there exists an LRC(2) code $\mathcal{C}$ of length $q - 1$ over $\mathbb{F}_q$ such that every code symbol has two disjoint recovery sets of sizes $r_1 - 1$ and $r_2 - 1$.

The discussed construction gives codes with distance close to the upper bound on LRC(2) codes derived in [17].

The definitions and constructions of this section extend straightforwardly to an arbitrary number $t \ge 2$ of recovery sets, giving rise to easily constructible LRC(t) codes with the parameters $(n, k, \{r_1, \ldots, r_t\})$, where $r_i + 1, i = 1, \ldots, t$ is the size of the blocks in the corresponding partition. At the same time, note that for $t \ge 3$ better parameters are obtained using random expanders; see [17, Theorem C]. Paper [17] also contains results on upper bounds for codes with an arbitrary number of recovery sets.

B. Correcting more than one erasure: $(r + \rho - 1, r)$ Local MDS Codes

A more general version of the local recovery problem calls for correcting more that one erasure within each recovery set. To address this task, we consider LRC codes in which the set of coordinates is partitioned into several subsets of cardinality $r + \rho - 1$ such that every local code is an $(r + \rho - 1, r)$ MDS code, where $\rho \ge 3$. Under this definition, every symbol of the codeword is a function of any r out of the $r + \rho - 2$ symbols, increasing the chances of successful recovery. A compact notation for such codes is (n, k, r, ρ) LRC codes, where n is the block length and k is the code dimension. A generalization of the bound on the distance (3) to the case of (n, k, r, ρ) LRC codes takes the form [11]

$$d \le n - k + 1 - \Big(\Big\lceil \frac{k}{r} \Big\rceil - 1\Big)(\rho - 1). \tag{10}$$

As before, we will say that the LRC code is optimal if its minimum distance attains this bound with equality.

We assume that $(r + \rho - 1)|n$ and $r|k$, although the latter constraint is again unessential. To construct the code using the ideas of Sect. II-A, we begin with a partition $\mathcal{A} = \{A_1, \ldots, A_m\}, m = n/(r + \rho - 1)$ of the set $A \subset \mathbb{F}, |A| = n$, such that $|A_i| = r + \rho - 1, 1 \le i \le m$. Let $g \in \mathbb{F}[x]$ be a polynomial of degree $r + \rho - 1$ that is constant on each of the blocks A_i. Given a message vector $a \in \mathbb{F}^k$, let us write it as $a = (a_0, ..., a_{r-1}) \in \mathbb{F}^k$, where each $a_i = (a_{i,0}, ..., a_{i,\frac{k}{r}-1})$ is a vector of length k/r. In analogy to (5), define the polynomial

$$f_a(x) = \sum_{i=0}^{r-1} \sum_{j=0}^{\frac{k}{r}-1} a_{ij} g(x)^j x^i.$$

The properties of the obtained codes are summarized in the following theorem.

Theorem 3.3: Let $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ be a linear code defined as the image of the evaluation map $a \mapsto ev_A(f_a)$. Then $\mathcal{C}$ is an optimal (n, k, r, ρ) LRC code in which local recovery of an erasure at location α can be performed by polynomial interpolation over any r locations of its recovery set.

IV. More algebraic connections: Cyclic LRC codes and LRC codes on curves

In classical coding theory there are two code families related to RS codes, namely cyclic codes and codes on algebraic curves. Since the codes considered above can be viewed as LRC analogs of RS codes, it is natural to consider these two families in relation to our construction. It turns out that both connections lead to new constructions of LRC codes as well as new problems in algebraic coding theory. Sections IV-A and IV-B below are based on [18] and [2] respectively.

A. Cyclic LRC codes

Cyclic codes form a well-established chapter in coding theory, important both theoretically and in applications. To construct cyclic LRC codes, we will rely on the multiplicative structure of the field $\mathbb{F}_q$. Let us choose the code length n to be a divisor of $q - 1$ and let us assume that the coordinates are labeled by n-th degree roots of unity in $\mathbb{F}_q$, i.e., $A = \{1, \alpha, \ldots, \alpha^{n-1}\}$, where α is a primitive root. Suppose that $(r + 1)|n$ and let $m = n/(r + 1)$ be their quotient. We rely on Proposition 2.2 to construct the polynomial $g(x)$.

Let H be a subgroup of the group $\mathbb{F}_q^*$ of order $|H|$ and let $r = |H| - 1$.

According to the discussion after Proposition 2.2, we can take $g(x) = x^{r+1}$. Examination of the expression (5) shows that the polynomial $f_a(x)$ can be written in the form

$$f_a(x) = \sum_{\substack{i=0 \\ i \neq r \bmod (r+1)}}^{\frac{k}{r}(r+1)-2} a_i x^i, \tag{11}$$

where the a_i's form the message vector. Following the construction (6), we obtain an LRC code, denoted by $\mathcal{C}$.

It is clear that the code $\mathcal{C}$ is cyclic, and it is easy to find its defining set of zeros. From the classical BCH bound it is well known that a set of $d-1$ consecutive zeros guarantees that $d(\mathcal{C}) \geq d$. The following theorem supplements this claim by identifying the set of zeros of $\mathcal{C}$ that supports the locality property.

Theorem 4.1: Consider the following sets of elements of $\mathbb{F}_q$:

$$L = \{\alpha^i, i \bmod (r+1) = l\}$$
$$D = \{\alpha^{j+s}, s = 0, \dots, n - \tfrac{k}{r}(r+1)\},$$

where $0 \leq l \leq r$ and $\alpha^j \in L$. The cyclic code with the defining set of zeros $Z := L \cup D$ is an optimal (n, k, r) q-ary cyclic LRC code[3].

If the set Z contains cosets of two groups of roots of unity of coprime orders $r_1 + 1$ and $r_2 + 1$, then this gives rise to an LRC(2) code $(n, k, \{r_1, r_2\})$ which has two disjoint recovery sets for every coordinate.

The following obvious remark sometimes facilitates the analysis of cyclic LRC codes.

Proposition 4.2: Let $\mathcal{C}$ be a cyclic LRC code with locality r. Suppose that $d^\perp$ is the distance of the dual code $\mathcal{C}^\perp$, then $r = d^\perp - 1$.

So far we were interested in RS-type LRC codes. Subfield subcodes of these codes form a natural analog of the family of BCH codes. Their properties are not so easy to analyze in general, but one possibility has been suggested in [18]. Let $\mathcal{C}$ be an (n, k, r) LRC code over $\mathbb{F}_{q^m}$ and denote by $\mathcal{C}|_{\mathbb{F}_q}$ the subcode of $\mathcal{C}$ formed by the codewords whose coordinates are contained in $\mathbb{F}_q$. Suppose we attempt to construct LRC codes over $\mathbb{F}_q$ as subfield subcodes of RS-type LRC codes over $\mathbb{F}_{q^m}$. Since $\mathcal{C}|_{\mathbb{F}_q} \subset \mathcal{C}$, we have that $d(\mathcal{C}|_{\mathbb{F}_q}) \geq d(\mathcal{C})$. At the same time, the dual distance of a cyclic LRC code $(\mathcal{C}|_{\mathbb{F}_q})^\perp$ may, and often does, decrease from its original value $r+1$. Thus, studying subfield subcodes is an appropriate context for constructing cyclic LRC codes over small alphabets with good distance and small locality.

B. LRC codes on algebraic curves

The RS-type LRC codes constructed above solve the problem of local recovery for codes of length n that is on the order of the size of the alphabet q. Consider again the problem of constructing long LRC codes for a fixed alphabet size. In classical coding theory good codes of this kind are obtained using the Goppa construction of codes on algebraic curves. Here we show how this approach can be utilized for codes with the locality constraint.

We begin with another view of the construction of RS-type LRC codes (4)-(6), focusing on the polynomial $g(x)$. Let $\Bbbk = \mathbb{F}_q$ denote the code alphabet. Recall that $g : \Bbbk \to \Bbbk$ defines a mapping such that there are exactly $r+1$ points that are mapped to every point in the range of g. In other words, we have $|g^{-1}(P)| = r+1$ for all P in the range. Switching to geometric language, let $X = Y = \mathbb{P}^1$ denote the projective line over the field $\Bbbk$, then $g : X \to Y$ is a covering map of lines such that the fiber above any point of Y in its range contains exactly $r+1$ points of X. For instance, in Example 1 the range of $g : x \mapsto x^3$ is the set $\{P_1 = 1, P_2 = 8, P_3 = 12\}$ and $g^{-1}(P_i) = A_i, i = 1, 2, 3$.

This view of our construction suggests the following generalization to codes on curves. Let X and Y be smooth projective absolutely irreducible curves over $\Bbbk$ and let $g : X \to Y$ be a rational separable map of curves of degree $r+1$. For example, let $\Bbbk = \mathbb{F}_9$ and consider the Hermitian curve X of genus 3 given by the equation $x^3 + x = y^4$. The curve X has 27 points in the finite plane, shown in Fig. 2 below, and one point at infinity.

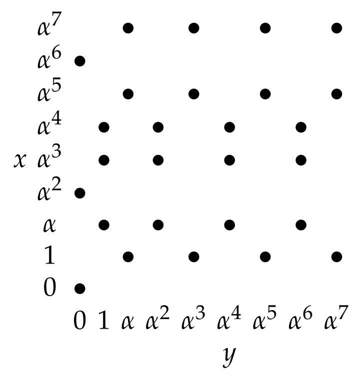

Fig. 2: 27 points of the Hermitian curve over $\mathbb{F}_9$; here $\alpha^2 = \alpha + 1$.

Take $Y = \mathbb{P}^1_\Bbbk$, then we can take g to be the map of degree $r+1 = 3$ given by the natural projection $g : (x, y) \mapsto y$. Another possibility is a degree-4 map $g : (x, y) \mapsto x$ whose range does not include the points $0, \alpha^2$, and α^6.

More generally, let $\Bbbk(X)$ and $\Bbbk(Y)$ denote the fields of rational functions on X and Y. By the primitive element theorem there exists a function $x \in \Bbbk(X)$ such that $\Bbbk(X) = \Bbbk(Y)(x)$ and that satisfies an algebraic equation of degree $r+1$ over $\Bbbk(Y)$. The function x can be considered as a map $x : X \to \mathbb{P}^1_\Bbbk$, and we denote its degree $\deg(x)$ by h.

The codes that we construct again belong to the class of evaluation codes. Let $S = \{P_1, \dots, P_s\} \subset Y(\Bbbk)$ be a subset of $\Bbbk$-rational points of Y in the finite space, and let Q_∞ be a positive divisor of degree $\ell \geq 1$ such that $\operatorname{supp} Q_\infty \subset \pi^{-1}(\infty)$, where $\pi : Y \to \mathbb{P}^1_\Bbbk$ is a projection map. To construct our codes let us assume that

$$A := g^{-1}(S) = \{P_{ij}, i = 0, \dots, r, j = 1, \dots, s\} \subseteq X(\Bbbk); \tag{12}$$

[3]We note that the sets L and D have a nonempty intersection, but their common elements appear in Z only once.

$$g(P_{ij}) = P_j \text{ for all } i, j.$$

Let $\{f_1, \ldots, f_m\}$ be a basis of the Riemann-Roch space $L(Q_\infty)$. Our codes will be constructed as evaluations of functions in the $\mathbb{k}$-subspace V of $\mathbb{k}(X)$ generated by the functions

$$\{f_j x^i, i = 0, \ldots, r-1, j = 1, \ldots, m\} \quad (13)$$

(note an analogy with (5)).

Definition 3: (*LRC codes on curves*). Consider the evaluation map

$$\begin{aligned} ev_A : & V \longrightarrow \mathbb{k}^{(r+1)s} \\ & F \mapsto (F(P_{ij}), i = 0, \ldots, r, j = 1, \ldots, s), \end{aligned} \quad (14)$$

and denote its image by $\mathcal{C}(Q_\infty, g)$. It is a linear code in the space $\mathbb{F}_q^n, n = (r+1)s$, and since $\operatorname{supp} Q_\infty \cap S = \emptyset$, the code is well defined.

The code coordinates are naturally partitioned into s subsets $A_j = \{P_{ij}, i = 0, ..., r\}, j = 1, \ldots, s$ of size $r+1$ each; see (12).

Theorem 4.3: The subspace $\mathcal{C}(Q_\infty, g) \subset \mathbb{F}_q$ forms an (n, k, r) linear LRC code with the parameters

$$\begin{aligned} n &= (r+1)s \\ k &= rm \ge r(\ell - g_Y + 1) \\ d &\ge n - \ell(r+1) - (r-1)h, \end{aligned} \quad (15)$$

provided that the right-hand side of the inequality for d is a positive integer. Local recovery of an erased symbol $F(P_{ij})$ can be performed by polynomial interpolation through the points of the recovery set A_j.

In particular, let us specialize this construction for codes on Hermitian curves. Let $q = q_0^2$, where q_0 is a prime power, let $\mathbb{k} = \mathbb{F}_q$, and let X be the Hermitian curve, i.e., a plane smooth curve of genus $g_0 = q_0(q_0 - 1)/2$ with the affine equation

$$X : x^{q_0} + x = y^{q_0+1}.$$

The curve X has $q_0^3 = q\sqrt{q}$ rational points in the affine plane. By taking g to be the projection on y as discussed above we obtain a family of LRC codes with the parameters

$$\begin{aligned} n = q_0^3,\ k = (\ell+1)(q_0 - 1),\ r = q_0 - 1 \\ d \ge n - \ell q_0 - (q_0 - 2)(q_0 + 1). \end{aligned}$$

It is also possible to take g to be a projection on x, which gives a family of LRC codes with similar parameters and locality $r = q_0$.

Asymptotically good code families. As in classical coding theory, we obtain infinite families of codes with good parameters by taking asymptotically maximal curves such as, for instance, the Garcia-Stichtenoth towers of curves. These curves are constructed by successively extending the function fields, adding algebraic elements that satisfy equations similar to the equation that defines the Hermitian curves. Similarly to the Hermitian case, there are several variants of the code construction. For instance, it is possible to construct a family of q-ary LRC codes whose rate and relative distance satisfy the asymptotic inequality

$$R \ge \frac{r}{r+1}\Big(1 - \delta - \frac{2\sqrt{q}}{q-1}\Big), \quad (16)$$

where $r = \sqrt{q}$ and $q = q_0^2$ for some prime power q_0. For $q_0 \ge 23$ this bound improves upon the Gilbert-Varshamov type bound for LRC codes discussed in the next section (see an example in Fig. 3).

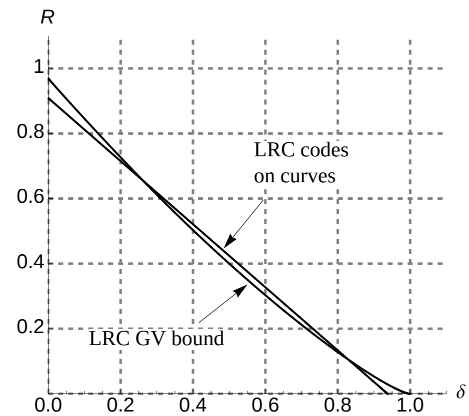

Fig. 3: The bound (16) shown together with the Gilbert-Varshamov type bound ($q_0 = 32$).

While this construction yields sequences of codes with asymptotically good parameters, its locality parameter r is fixed once we choose the code alphabet. In principle one would want to have flexibility in choosing r in a way similar to the construction of RS-type LRC codes. This is indeed possible by studying certain quotients of curves in the Garcia-Stichtenoth tower. As a result, we obtain asymptotically good codes over a fixed field $\mathbb{F}_q$ with a range of values of r with parameters similar to the ones mentioned above.

Concluding this section, note that the proposed approach generalizes to codes with more than one recovery set for every coordinate (the so-called availability problem). Indeed, when discussing the example in Fig. 2 we remarked that there are two natural maps from X to $\mathbb{P}^1$. A closer look confirms that together they define a pair of orthogonal partitions of the set of $n = q_0^3 - q_0 - 1$ affine points of the Hermitian curve, giving rise to an LRC code of length n with two disjoint recovery sets for rdeach codeword symbol.

V. Bounds on the parameters of LRC codes

Here we discuss bounds on the rate and distance of LRC codes introduced in Definition 1. It can be easily seen that the rate of any LRC code $\mathcal{C}$ with locality r is at most $R(\mathcal{C}) \le r/(r+1)$. Intuitively this is justified by the fact that any $r+1$ codeword symbols within a recovery set satisfy a functional relation, so they contain at most r information symbols.

How large can $d(\mathcal{C})$ be? Even in the classical coding problem, this question is addressed in more than one way, depending on whether we account for the value of q or not. The Singleton-type bound (3), discussed above, does not depend on the size of the alphabet. A bound that accounts for the value of q, proved in [3], has the following form:

$$k \leq \min_{s\geq 1} \{sr + k_q(n - s(r+1), d)\}, \quad (17)$$

where $k_q(n,d)$ is the maximum dimension of a code of length n and distance d over $\mathbb{F}_q$ (with no locality assumptions). It is also possible to derive lower Gilbert-Varshamov-type bounds on the parameters of LRC codes using the probabilistic method, bringing the state of bounding the parameters of LRC codes to the same status as bounds on classical error correcting codes. In particular, sequences of codes of asymptotically positive rate exist if and only if the number of correctable errors does not exceed the $(q-1)/2q$ proportion of the code length. The results of [3], [17] imply that the same conclusion is valid once we add the locality constraint (for any constant r). Therefore, adding the locality constraint does not shift the "Plotkin point" for asymptotic relative distance from the value $(q-1)/q$. The best asymptotic lower and upper bounds on LRC codes are shown in Fig. 4.

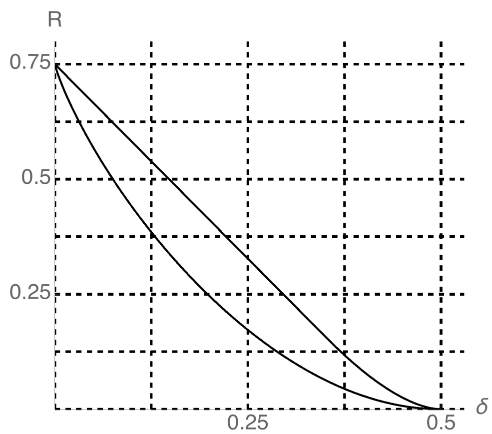

Fig. 4: Asymptotic bounds for the rate R of binary LRC codes as a function of the relative distance δ; $r=3$. The upper curve is obtained from the bound (17), and the lower curve is a GV-type bound.

VI. Outlook

A. LRC codes in industry

Apart from their theoretical merits, LRC codes offer an efficient solution for data protection in large-scale distributed storage systems. Data encoding schemes employed by companies using or providing distributed storage solutions are based primarily on the ease of implementation, update, and maintenance. Driven by these metrics, companies are mostly interested in implementing LRC codes that provide the locality property only for the information part of the codeword. Codes with this property are said to have *information symbol* locality. It turns out that constructing such codes with good minimum distance is relatively simple, which is why these codes are popular in current industry solutions.

To construct an (n,k,r) LRC code with information symbol locality and good minimum distance, begin with an $(n-\frac{k}{r}, k)$ MDS code (typically an RS code). To account for locality, let us partition its k information symbols into k/r disjoint sets of size r and add one parity check symbol for each set.

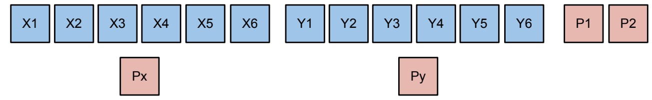

Fig. 5: $(16,12,6)$ LRC code used in Windows Azure storage [10].

This results in a code of length n with information locality r. Examples of LRC codes constructed in this way are already used in practice or have been tested by industry, and here we list a few of them.

The free software storage platform *Ceph* enables the users to protect their information by simple replication, RS code, or an LRC code. In another project [13], the authors constructed a $(16,10,5)$ LRC code based on the $(14,10)$ RS code and tested it on a cluster at Facebook's data warehouse. The construction proposed in [13] has in fact the all-symbol locality property. Finally, Windows Azure Storage (WAS), Microsoft's scalable cloud storage system that has been in use for some years [10], uses a $(16,12,6)$ LRC code shown in Fig. 5. Here P_1 and P_2 are the global parities found from all the 12 information symbols $X_i, Y_i, i = 1,...,6$. They are employed in cases of more than one failure among the nodes. The symbols P_x and P_y are the parities that provide local recovery for the information symbols by accessing 6 other symbols within the recovery set.

Encouraged by the fast embrace of LRC codes by large-scale users of distributed storage, we believe that there is room for implementation and testing of other code families with the locality property. Specific storage applications may benefit from all-symbol locality or large minimum distance. At the same time, the solutions should be tailored to the needs of the application, including update complexity, security and availability of the data, and other features.

B. LRC codes on graphs

An interesting generalization of the LRC coding problem is related to local recovery that is constrained by the topology of the computer network. Consider a graph on n storage nodes whose edges describe the available communication links between the nodes. Similarly to the problem studied above, we require that every node can recover its storage contents by reading the information stored in its neighbor nodes in the graph. A set of vectors over a finite alphabet that can be stored in the nodes to satisfy this constraint, forms an LRC code on the graph, and we seek such codes of the largest possible size. This problem was recently introduced in [12] (in a different form, it was also studied earlier in [5]). It is also shown to be (in some sense) a dual of the well-known index coding problem [12], [14]. Major open questions in this area include finding constructions of good codes for the graph LRC problem, for instance, for families of graphs with some structure, as well as advancing connections between LRC codes and index coding.

C. Maximally recoverable codes: Can a code be LRC and MDS?

MDS codes form a practically appealing family because they provide the best possible error resilience for a given amount of storage overhead. In formal terms, this amounts to saying that any k symbols in a k-dimensional MDS code form an information set. At the same time, the locality constraint requires some dependence among the codeword symbols, so locality and the MDS property cannot be combined in one construction. How close to being both MDS and LRC can a code be? This question brings in the following natural definition: Call an (n,k,r) LRC code *maximally recoverable* [6] if every k coordinates that do not contain a full recovery set form an information set. Note any k-subset that contains a recovery set cannot be an information set.

For large sets of parameters (n,k,r) maximally recoverable codes have been constructed in [15], [19], [6]; however, none of these results yield code families over alphabets q of size comparable to the code length. At the same time, as shown above, it is possible to construct LRC codes over small alphabets. This gives rise to the following *open problem*: is it possible to construct maximally recoverable codes with small q, or does maximality necessarily require a superlinear alphabet size?

Observe that the maximality property is not resolved even for the RS-type LRC code family presented in this paper: we do not know if (apart from the trivial cases of $r=1,k$) among the constructed codes there are maximally recoverable ones.

D. AG codes: Parameters and availability

The construction of LRC codes on curves in Sect. IV-B is rather general in the sense that it applies to any pair of curves equipped with a covering map. At the same time, the estimates of the parameters of the obtained codes derived using this general approach do not take into account specifics of individual families of curves, and for this reason may be somewhat crude. Thus, the initial results reported above could be specialized and improved in examples that rely on properties of specific curves and their maps.

Another problem, mentioned only very briefly in Sect. IV-B and in [2] concerns the availability problem for algebraic geometric codes. While we have explored the most natural approach to this problem, the parameters of the obtained codes are far from optimal. It may be possible to obtain better LRC codes relying on the automorphism groups of curves and their codes, and we envision this as another avenue for further studies.

Yet another topic of possible studies is related to decoding of the constructed codes. While we have focused on local erasure recovery, occasionally we will face the task of global decoding for the purpose of error and erasure correction. Here we tacitly rely on the existing decoding algorithms of algebraic geometric codes, although conceivably the structure of our codes could support decoding algorithms designed specifically for this family. It is also of interest to explore the connection of these codes with generic list decoding algorithms of codes on curves.

ACKNOWLEDGMENTS: We would like to thank our coauthors Robert Calderbank, Alexey Frolov, Sreechakra Goparaju, and Serge Vlăduţ for their collaboration on [18], [17] and [2], as well as the Editor for encouraging us to write this article. We are also grateful to our colleagues Alexandros G. Dimakis and P. Vijay Kumar for insightful discussions of the problems covered in this paper.

A.B. acknowledges support of the NSF grants CCF1422955 and CCF1217245.

REFERENCES

[1] S. Ball, "On sets of vectors of a finite vector space in which every subset of basis size is a basis," *J. Eur. Math. Soc.*, vol. 14, no. 3, pp. 733–748, 2012.

[2] A. Barg, I. Tamo, and S. Vlăduţ, "Locally recoverable codes on algebraic curves," in *Proc. IEEE Int. Sympos. Inform. Theory, Hong Kong*, 2015, pp. 1252–1256.

[3] V. Cadambe and A. Mazumdar, "Upper bounds on the size of locally recoverable codes," *IEEE Trans. Inform. Theory*, to appear. Available online at arXiv:1308.3200v2, 2015.

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

[5] M. Gadouleau and S. Riis, "Graph-theoretical constructions for graph entropy and network coding based communications," *IEEE Trans. Inform. Theory*, vol. 57, no. 10, pp. 6703–6717, 2011.

[6] P. Gopalan, C. Huang, B. Jenkins, and S. Yekhanin, "Explicit maximally recoverable codes with locality," *IEEE Trans. Inform. Theory*, vol. 60, no. 9, pp. 5245–5256, 2014.

[7] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inform. Theory*, vol. 58, no. 11, pp. 6925–6934, 2011.

[8] J. Han and L. A. Lastras-Montano, "Reliable memories with subline accesses," in *Proc. IEEE Internat. Sympos. Inform. Theory*, 2007, pp. 2531–2535.

[9] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in *Sixth IEEE International Symposium on Network Computing and Applications*, 2007, pp. 79–86.

[10] 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. USENIX Annual Technical Conference (USENIX ATC)*, 2012.

[11] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration and erasure correction," *IEEE Trans. Inform. Theory*, vol. 60, no. 8, pp. 4637–4660, 2014.

[12] A. Mazumdar, "Storage capacity of repairable networks," *IEEE Trans. Inform. Theory*, to appear. Available online at arXiv:1408.4862.

[13] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," in *Proceedings of the VLDB Endowment*, 2013.

[14] K. Shanmugam and A. Dimakis, "Bounding multiple unicasts through index coding and locally repairable codes," in *Proc. IEEE Internat. Sympos. Inform. Theory, Honolulu, HI*, 2014, pp. 296–300.

[15] N. Silberstein, A. S. Rawat, O. Koyluoglu, and S. Vishwanath, "Optimal locally repairable codes via rank-metric codes," in *Proc. IEEE Int. Sympos. Inform. Theory, Boston, MA*, 2013, pp. 1819–1823.

[16] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inform. Theory*, vol. 60, no. 8, pp. 4661–4676, 2014.

[17] I. Tamo, A. Barg, and A. Frolov, "Bounds on the parameters of locally recoverable codes," Available online at arXiv:1506.07196, 2015.

[18] I. Tamo, A. Barg, S. Goparaju, and R. Calderbank, "Cyclic LRC codes and their subfield subcodes," in *Proc. IEEE Int. Sympos. Inform. Theory, Hong Kong, PRC*, 2015, pp. 1262–1266.

[19] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis, "Optimal locally repairable codes and connections to matroid theory," in *Proc. 2013 IEEE Internat. Sympos. Inform. Theory*, 2013, pp. 1814–1818.