On the Parameters of Codes for Data Access

Altan B. Kılıç[*], **Alberto Ravagnani**[*], **Emina Soljanin**[†]
[*]Department of Mathematics and Computer Science, Eindhoven University of Technology, the Netherlands
[†]Department of Electrical & Computer Engineering, Rutgers University, U.S.A.
{a.b.kilic, a.ravagnani}@tue.nl, emina.soljanin@rutgers.edu

2024 IEEE International Symposium on Information Theory (ISIT) | 979-8-3503-8284-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISIT57864.2024.10619288

***Abstract*—This paper studies two crucial problems in the context of coded distributed storage systems directly related to their performance: 1) for a fixed alphabet size, determine the minimum number of servers the system must have for its service rate region to contain a prescribed set of points; 2) for a given number of servers, determine the minimum alphabet size for which the service rate region of the system contains a prescribed set of points. The paper establishes rigorous upper and lower bounds, as well as code constructions based on techniques from coding theory, optimization, and projective geometry.**

I. Introduction

Storage layers provide data access services for executing applications. Thus, a computing system's overall performance depends on its underlying storage system's data access performance. Modern storage systems replicate data objects across multiple servers. An object's replication degree corresponds to its expected demand. However, the access request volume and data object popularity fluctuate in practice. Redundancy schemes that combine erasure coding with replication can support such scenarios better than replication alone.

Storage servers can handle access requests up to a specific maximal service rate. The service rate region of a redundant storage scheme is a recently introduced performance metric notion [1]; see also [2] and references therein. It is defined as a set of all data access request rates that the system implementing the scheme can support.

Two main problem directions are associated with distributed storage access: 1) For a given (implemented) redundancy scheme, we ask what its service rate region is. 2) For a given (desired) service rate region, we ask which redundancy scheme has the service rate region that includes the desired one with some optimal cost or some required properties. Many other related problems are briefly outlined in [2, Sec. VIII], most notably performance analysis of storage schemes.

The service rate region is a new concept and the subject of several recent papers [1]–[5]. This work has answered some questions in the above directions, primarily for selected binary codes or systems storing two data objects. The main difference between this collection of papers and nearly all recent work on coded distributed storage is that it primarily addresses the external uncertainty in the storage systems (download request fluctuations) rather than the internal uncertainty (e.g., straggling) in operations of the system itself (see, e.g., [6]–[12]). A related line of work concerning external uncertainty considers systems with uncertainty in the mode and level of access to the system [13], [14].

This paper addresses a problem within the second main direction mentioned above: designing codes for a given service rate region. It focuses on scenarios when a set of the points in the region is provided and asks the following two questions:

1) How many servers do we need if the field size is fixed?
2) What is the minimum field size for a code over a fixed number of servers (i.e., code length)?

Both questions are essential in practice. The first question concerns systems with computational complexity (field size) limits, and we want to minimize the number of servers. The second question is relevant when the number of servers is limited, and we want to minimize the field size to reduce the computational complexity. The first problem was studied in [15]. The second problem has not been studied before.

The rest of the paper is organized as follows: Sec. II formulates the problem. Sec. III, presents two general approaches to attack the problems of the paper. Sec. IV is devoted to bounds and existence results. Sec. V shows some applications of the paper's results. Because of space constraints, most proofs are omitted and will appear in an extended version of this work.

II. Service Rate Region and Problem Statement

In this section, we establish the notation, define the service rate region of distributed data storage systems, and present the parameters that are the subject of the two main problems addressed in this paper. For a standard reference on coding theory, we refer to [16].

Notation 1. Throughout the paper, $\mathscr{P}$ denotes the set of prime powers, $\mathbb{F}_q$ denotes a finite field of size q with $q \in \mathscr{P}$. Unless otherwise stated, all vectors in the paper are column vectors, and $e_i \in \mathbb{F}_q^k$ is the i-th standard basis vector for $i \in \{1, \ldots, k\}$.

We consider a distributed storage system with $n \in \mathbb{Z}_{\geq k}$ identical servers where $k \in \mathbb{Z}_{\geq 2}$ distinct data objects (elements of $\mathbb{F}_q$) are stored on the servers. Each server stores an $\mathbb{F}_q$-linear combination of the k objects. Therefore, the system can be specified by a matrix $G \in \mathbb{F}_q^{k \times n}$, called the **generator matrix** of the system. Each column of G corresponds to a server. More precisely, if $u = (u_1, \ldots, u_k)$ is the tuple of objects to be stored, then the ν-th column of G stores the ν-th coordinate of $u \cdot G$ for $\nu \in \{1, \ldots, n\}$. If a column of G is a non-zero

A.B.K. is supported by the Dutch Research Council through grant VI.Vidi.203.045. A. R. is supported by OCENW.KLEIN.539, VI.Vidi.203.045, and by the Academy of Arts and Sciences of the Netherlands. E. Soljanin's work was in part supported by NSF Award CIF-2122400.

scalar multiple of one of the standard basis vectors, it is called a **systematic server**. Otherwise, it is called a **coded server**.

Notation 2. In the sequel, G denotes a $k \times n$ matrix over $\mathbb{F}_q$ of rank k. We assume that G has no zero column. The list of columns of G is denoted by $\mathrm{col}(G)$, and the ν-th column of G is denoted by G^ν.

In our model, each server can simultaneously process the data access requests with a cumulative serving rate of at most one. A **demand vector** is a k-tuple of nonnegative real numbers $(\lambda_1, \ldots, \lambda_k)$. The assumption that G has no zero column makes sense since a server storing the zero linear combination is useless. The restriction on the capacity of the servers (each server has capacity 1) implies that the system cannot "support" any demand vector. When a server reaches its capacity, other servers need to be contacted to serve users. The next definition formalizes these concepts.

Definition 1. For each $i \in \{1, \ldots, k\}$ we construct sets $\mathscr{R}_i(G)$ as follows. A set $R \subseteq \{1, \ldots, n\}$ is in $\mathscr{R}_i(G)$ if:

- $e_i \in \langle G^\nu \mid \nu \in R \rangle_{\mathbb{F}_q}$, and
- there is no $R' \subsetneq R$ with $R' \in \mathscr{R}_i(G)$.

The elements of $\mathscr{R}_i(G)$ are called the (**minimal**) **recovery sets** for the i-th object. A demand vector $(\lambda_1, \ldots, \lambda_k)$ is **supported** by the system (or supported by G) if there exists real numbers

$$(\lambda_{i,R} \in \mathbb{R}_{\geq 0} \mid i \in \{1, \ldots, k\},\ R \in \mathscr{R}_i(G))$$

with the following properties:

$$\begin{cases} \displaystyle\sum_{R \in \mathscr{R}_i} \lambda_{i,R} = \lambda_i \text{ for } i \in \{1, \ldots, k\}, & (1) \\ \displaystyle\sum_{i=1}^{k} \sum_{\substack{R \in \mathscr{R}_i \\ \nu \in R}} \lambda_{i,R} \leq 1 \quad \text{for } \nu \in \{1, \ldots, n\}. & (2) \end{cases}$$

Lastly, the **service rate region** of the system generated by G is defined as

$$\Lambda(G) := \{\lambda \in \mathbb{R}^k \mid \lambda \text{ is supported by } G\} \subseteq \mathbb{R}^k.$$

The constraints in (1) ensure that the demand vector is served, and those in (2) guarantee that the servers are not overloaded.

Example 1. Let

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{3\times 4}.$$

We have $\mathscr{R}_i(G) = \{\{i\}, \{1,2,3,4\} \setminus \{i\}\}$ for $i \in \{1,2,3\}$. The service rate region $\Lambda(G)$ is depicted in Figure 1. As an example, we have $(1.4, 0.6, 0.6) \in \Lambda(G)$ by letting

$$\lambda_{1,R} = \begin{cases} 1 & \text{if } R = \{1\}, \\ 0.4 & \text{if } R = \{2,3,4\}, \end{cases} \quad \lambda_{i,R} = \begin{cases} 0.6 & \text{if } R = \{i\}, \\ 0 & \text{otherwise,} \end{cases}$$

for $i \in \{2,3\}$ in Definition 1.

Note that the service rate region in Example 1 is a polytope. Recall that a polytope $P \subseteq \mathbb{R}^k$ is called **down-monotone** if $0 \leq x \leq y$ and $y \in P$ imply $x \in P$. The service rate region is always a convex, down-monotone polytope. Therefore, we will call it the **service rate region polytope** of the matrix G, see [17]. Recall that any convex polytope is the convex hull of its vertices (see [18] for more on polytopes). Therefore, we have the following list of observations.

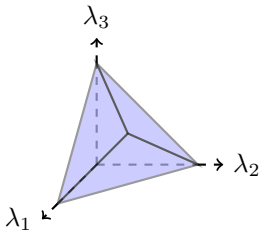


Fig. 1: The service rate region for Example 1.

Proposition 1. *Let $S \subseteq \mathbb{R}^k$ and $G \in \mathbb{F}_q^{k\times n}$ be a full-rank matrix. We have*

1) *$S \subseteq \Lambda(G)$ implies $\mathrm{conv}(S) \subseteq \Lambda(G)$, where $\mathrm{conv}(S)$ is the convex hull of the points of S,*
2) *$\mathrm{conv}(S) = \mathrm{conv}(S')$ where*

$$S' = \{s' \in S \mid s' \notin \mathrm{conv}(S \setminus \{s'\})\}.$$

Notation 3. In the sequel, $S \subseteq \mathbb{R}^k$ will always denote a nonempty set in reduced form. Following the notation of Proposition 1, we call S' the **reduced form** of S.

In practice, we expect to know the demand vectors, and one needs to design systems (i.e., G) whose service rate region polytopes contain these demand vectors. Many systems can be chosen (see, for instance, Figure 2), and the most efficient one will be selected (here "efficiency" depends on the needs of the user). To this end, we study the following parameters, whose importance has been explained in Sec. I.

Definition 2. When q is fixed, we define

$$n_q(S) = \min\left\{n \in \mathbb{Z}_{\geq k} \mid \exists G \in \mathbb{F}_q^{k\times n} \text{ with } S \subseteq \Lambda(G)\right\}$$

and when n is fixed, we define

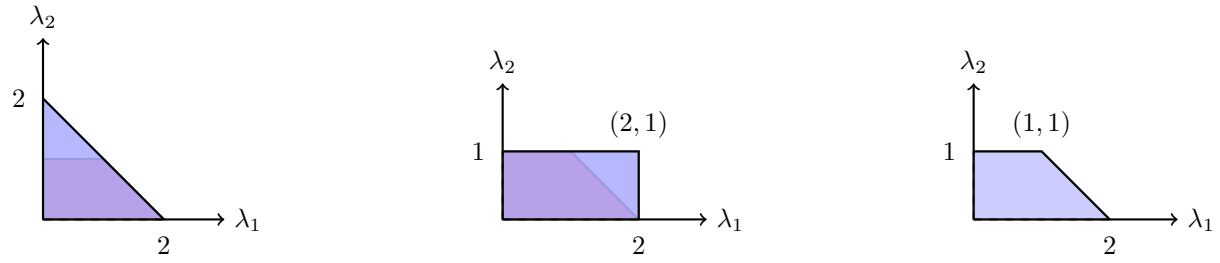
$$q_n(S) = \min\left\{q \in \mathscr{P} \mid \exists G \in \mathbb{F}_q^{k\times n} \text{ with } S \subseteq \Lambda(G)\right\}.$$

The quantity $n_q(S)$ is a performance metric, and the minimum in its definition always exists since we can always use replication, resulting in many servers (n) compared to the optimal value $n_q(S)$.

Example 2. Let $S = \{(1,1), (2,0)\}$. We have $n_2(S) = 3$. There are different polytopes $\Lambda(G)$ that contain $\mathrm{conv}(S)$ with $G \in \mathbb{F}_2^{2\times 3}$, see Figure 2 for three of them.

While the number $n_q(S)$ always exists, the existence of $q_n(S)$ is not guaranteed.

Example 3. Let $S = \{(2,1), (1,2)\}$. We have $n_2(S) = 4$. The following argument shows that $q_3(S)$ does not exist. Regardless of the field size, one needs to fully use at least



(a) $G_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ (b) $G_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ (c) $G_3 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

Fig. 2: Set $S = \{(1,1),(2,0)\}$ is in $\Lambda(G_i)$ for $i \in \{1,2,3\}$ and $\Lambda(G_3) = \text{conv}(S)$. Set $\text{conv}(S)$ is shown in (a) and (b) with a different color. Observe that 1) any code whose service rate region contains the points in S must contain the minimal region shown in (c), and 2) regions that are proper supersets of the minimal are achievable without enlarging the field size.

three servers for only one of the points in S. However, the servers used for the first point cannot support the remaining point. Therefore, there must be at least 4 servers.

III. Methods and Tools

We now propose a characterization of $n_q(S)$ that will allow us to apply mixed integer optimization to compute it.

Remark 1. Let $q \in \mathscr{P}$. Thus, $q = p^r$ for some prime p and positive integer r. Let z denote a root of an irreducible polynomial in $\mathbb{F}_p[x]$ of degree r in $\mathbb{F}_q$. We represent $\mathbb{F}_q$ as $\mathbb{F}_p^r$ using the function $g : \mathbb{F}_q \to \mathbb{F}_p^r$, where $g(z^{j-1}) = e_{r+1-j}$ for all $j \in \{1,\dots,r\}$. For $h \in \mathbb{F}_q$, let $\overline{h} \in \{0,\dots,q-1\}$ be the integer whose p-ary expansion is equal to $g(h)$. We then have $\mathbb{Z}_q = \{0,1,\dots,q-1\} = \{\overline{h} \mid h \in \mathbb{F}_q\}$.

Let $\mathbf{0}_k$ be the all-zeros vector in $\mathbb{Z}_q^k$. We denote by $\overline{v_j} \in \mathbb{Z}_q^k \setminus \mathbf{0}_k$ the vector corresponding to the q-ary expansion of the integer $j \in \{1,\dots,q^k-1\}$. Lastly, for all $\overline{v_j} \in \mathbb{Z}_q^k \setminus \mathbf{0}_k$, we construct $v_j \in \mathbb{F}_q^k \setminus \mathbf{0}_k$ such that the i-th coordinate of v_j is equal to $h \in \mathbb{F}_q$ when the i-th coordinate of $\overline{v_j}$ is $\overline{h} \in \mathbb{Z}_q$ for each $i \in \{1,\dots,k\}$.

We also define following two sets:

$$D(H) = \left\{ d \in \left\{1,\dots,\frac{q^k-1}{q-1}\right\} \ : \ u_d \in PG(k-1,q) \setminus H \right\},$$
$$I(H) = \{i \in \{1,\dots,k\} \ : \ e_i \in PG(k-1,q) \setminus H\}.$$

for a hyperplane H of $PG(k-1,q)$. In words, $D(H)$ and $I(H)$ are the set of projective points and standard basis vectors that do not lie on H, respectively.

Using the representation of Remark 1 we get the following:

Proposition 2. *We have*

$$n_q(S) = \min\left\{ \sum_{j=1}^{q^k-1} n_j \mid \text{ there exists a matrix } G \text{ over } \mathbb{F}_q \text{ with } k \text{ rows, } |\{\nu \mid G^\nu = v_j\}| = n_j \text{ for all } j, \text{ and } S \subseteq \Lambda(G) \right\}.$$

We continue by translating the concept of recovery sets under the representation of Remark 1.

Definition 3. Let $i \in \{1,\dots,k\}$. A set $T \subseteq \{1,\dots,q^k-1\}$ is called a **recovery set** for the i-th object if $e_i \in \langle v_j \mid j \in T\rangle_{\mathbb{F}_q}$ and there is no $T' \subsetneq T$ with $e_i \in \langle v_j \mid j \in T'\rangle_{\mathbb{F}_q}$. We denote the set of recovery sets of the i-th object by $\mathscr{T}_i$. Note that $|\mathscr{T}_i| = |\mathscr{T}_j|$ for all $i,j \in \{1,\dots,k\}$.

We illustrate the concepts introduced in Remark 1 and Definition 3 with an example.

Example 4. Let $k = 2$ and $q = 3$. Then $v_1 = (0,1)$, $v_2 = (0,2)$, $v_3 = (1,0)$, $v_4 = (1,1)$, $v_5 = (1,2)$, $v_6 = (2,0)$, $v_7 = (2,1)$, $v_8 = (2,2)$. Moreover, we have

$$\mathscr{T}_1 = \{\{1,8\},\{2,8\},\{3\},\{1,4\},\{2,4\},\{5,8\},\{1,5\},\{2,5\},\{4,5\},\{6\},\{7,8\},\{1,7\},\{2,7\},\{4,7\}\},$$

$$\mathscr{T}_2 = \{\{1\},\{2\},\{3,8\},\{3,4\},\{5,8\},\{3,5\},\{4,5\},\{6,8\},\{4,6\},\{5,6\},\{7,8\},\{3,7\},\{4,7\},\{6,7\}\}.$$

The following result gives a sufficient condition for a demand vector to be supported under the terminology of Remark 1. It can be applied to compute $n_q(S)$ for any set S of finite cardinality using, for instance, mixed integer optimization; see [19]. Any feasible solution of the program gives an upper bound on $n_q(S)$.

Theorem 1. *Assume that there exists a collection of non-negative integers n_j for $j \in \{1,\dots,q^k-1\}$ and a collection of non-negative real numbers $\{\theta_{i,T} \mid i \in \{1,\dots,k\}, T \in \mathscr{T}_i\}$ with the following properties:*

$$\begin{cases} \displaystyle\sum_{T \in \mathscr{T}_i} \theta_{i,T} = \lambda_i \quad \text{for } 1 \le i \le k, & (3) \\ \displaystyle\sum_{i=1}^{k} \sum_{\substack{T \in \mathscr{T}_i \\ j \in T}} \theta_{i,T} \le n_j \quad \text{for } 1 \le j \le q^k - 1. & (4) \end{cases}$$

There exists a system with $n = \sum_{j=1}^{q^k-1} n_j$ servers that supports (see Definition 1) the point $(\lambda_1,\dots,\lambda_k) \in \mathbb{R}^k$.

The conditions in Theorem 1 can be regarded as constraints of an optimization problem that minimizes the number of servers of a system such that the system supports the given list of demand vectors. We illustrate how to treat two demand vectors with an example focusing on a particularly tractable input set for convenience of exposition. This illustrates how Theorem 1 is applied.

Example 5. Let $a, b \in \mathbb{Z}$ with $a \geq b > 0$, and $S = \{(a,0),(0,b)\}$. We want to compute $n_2(S)$ using Theorem 1 combined with optimization. We have $v_1 = (0,1)$, $v_2 = (1,0)$, $v_3 = (1,1)$, $\mathscr{T}_1 = \{\{1,3\},\{2\}\}$, and $\mathscr{T}_2 = \{\{1\},\{2,3\}\}$. By applying Theorem 1 to both demand vectors in S we obtain that $n_2(S)$ equals the optimal value of the following integer linear optimization program (ILP). Here we slightly abuse the notation (instead of following the notation of Theorem 1) to emphasize which variable of ILP contributes to which point of S:

$$\begin{array}{ll} \text{minimize} & n_1 + n_2 + n_3 \\ \text{subject to} & s^1_{1,\{1,3\}} + s^1_{1,\{2\}} = a, \\ & s^2_{2,\{1\}} + s^2_{2,\{2,3\}} = b, \\ & s^1_{1,\{1,3\}} \leq n_1,\ s^1_{1,\{2\}} \leq n_2,\ s^1_{1,\{1,3\}} \leq n_3, \\ & s^2_{2,\{1\}} \leq n_1,\ s^2_{2,\{2,3\}} \leq n_2,\ s^2_{2,\{2,3\}} \leq n_3, \\ & s^1_{1,\{1,3\}},\ s^1_{1,\{2\}},\ s^2_{2,\{1\}},\ s^2_{2,\{2,3\}} \in \mathbb{R}_{\geq 0}, \\ & n_1,\ n_2,\ n_3 \in \mathbb{Z}_{\geq 0}. \end{array}$$

We now point out a key observation on the role of the number of coded servers, namely n_3. We must have $n_3 \leq n_1$ and $n_3 \leq n_2$. This follows from the fact $\{3\} \notin \mathscr{T}_1 \cup \mathscr{T}_2$, but $\{1,3\} \in \mathscr{T}_1$ and $\{2,3\} \in \mathscr{T}_2$. Since $a \geq b$ by assumption, one can further assume that $n_2 \geq n_1 \geq n_3$. Thus, the problem reduces to:

$$\begin{array}{ll} \text{minimize} & n_1 + n_2 + n_3 \\ \text{subject to} & n_2 + n_3 \geq a, \\ & n_1 + n_3 \geq b, \\ & n_2 - n_1 \geq 0, \\ & n_1 - n_3 \geq 0, \\ & n_1,\ n_2,\ n_3 \in \mathbb{Z}_{\geq 0}. \end{array}$$

One feasible solution is

$$(n_1, n_2, n_3) = (\lceil b/2 \rceil, a - \lfloor b/2 \rfloor, \lfloor b/2 \rfloor).$$

Therefore, we have $n_2(S) \leq \lceil a + b/2 \rceil$. We will later show that $n_2(S) \geq \lceil a + b/2 \rceil$ by Theorem 4, which in turn proves that $n_2(S) = a + \lceil b/2 \rceil$.

The following result uses projective points and has been proven in [15] for $q = 2$ with slightly different notations. We state it here for arbitrary q. The proof is essentially the same. We will apply later in Theorem 4 to derive a lower bound on $n_q(S)$ for some sets S with a particular structure.

Theorem 2. *Let $S = \{s^1, \ldots, s^m\} \subseteq \mathbb{R}^k$. Let $G \in \mathbb{F}_q^{k \times n_q(S)}$ and let H be a hyperplane of $\mathrm{PG}(k-1,q)$. Following the notation of Remark 1, if $|\{\nu \mid G^\nu = v_d\}| = n_d$, then*

$$\sum_{d \in D(H)} n_d \geq \left\lceil \max\Big\{ \sum_{i \in I(H)} s_i^t \ :\ 1 \leq t \leq m \Big\} \right\rceil.$$

IV. Upper and Lower Bounds

This section focuses on bounds, existence results, and the interplay of the two parameters introduced in Definition 2.

We start with the simplest possible case, $|S| = 1$. Here, we have to cover a single demand vector and determine the minimum number of servers and field size for which this is possible.

Proposition 3. *Let $s \in \mathbb{R}^k_{\geq 0}$ and $S = \{s\}$. For any q, we have*

$$n_q(S) = \sum_{i=1}^{k} \lceil s_i \rceil .$$

Furthermore,

$$q_n(S) = \begin{cases} \text{does not exist} & \text{if } n < \sum_{i=1}^{k} \lceil s_i \rceil , \\ 2 & \text{otherwise.} \end{cases}$$

The case where $|S| = 2$ is significantly more involved then $|S| = 1$ and it will appear in the extended version of this work. We continue with general upper and lower bounds on $n_q(S)$ for a set S of any cardinality.

Theorem 3. *Let $S = \{s^1, \ldots, s^m\} \subseteq \mathbb{R}^k$. We have*

$$\alpha(S) \leq n_q(S) \leq \beta(S),$$

where

$$\alpha(S) = \max\left\{ \sum_{i=1}^{k} \lceil s_i^t \rceil \mid 1 \leq t \leq m \right\},$$

$$\beta(S) = \sum_{i=1}^{k} \max\left\{ \lceil s_i^t \rceil \mid 1 \leq t \leq m \right\}.$$

Obtaining bounds on $q_n(S)$ appears to be more difficult, in general, than obtaining bounds on $n_q(S)$. The quantities introduced in Theorem 3 can, however, be used to prove non-existence results on $q_n(S)$.

Proposition 4. *The quantity $q_{\alpha(S)}(S)$ never exists if $|S| > 1$ (and S is in reduced form). In particular, following the notation of Theorem 3, if $|S| > 1$ then $n_q(S) \geq \alpha(S) + 1$ for any q.*

In the remainder of this section, we focus on sets of the form $S = \{X_i e_i \mid 1 \leq i \leq k\}$, for a given vector $X \in \mathbb{R}^k_{\geq 0}$. Note that the convex hull of such a set S is a simplex polytope in the positive orthant, one of the best-known polytopes.

The next result establishes a lower bound for the quantity $n_q(S)$, which explicitly depends on the underlying field size q. Its proof uses Theorem 2, which we stated as a preliminary result in Sec. III.

Theorem 4. *Let $(X_i)_{i=1}^k$ be a non-increasing sequence of non-negative integers and let $S = \{X_i e_i \mid 1 \leq i \leq k\}$. For all q, we have*

$$n_q(S) \geq \left\lceil \sum_{i=1}^{k} q^{1-i} X_i \right\rceil .$$

Proof. Let $\begin{bmatrix} a, b \end{bmatrix}_q$ denote the number of b-dimensional subspaces of an a-dimensional vector space over $\mathbb{F}_q$ for non-negative integers $a \geq b$; see [20]. We apply Theorem 2 to all hyperplanes of $\mathrm{PG}(k-1, q)$. This gives $\begin{bmatrix} k, k-1 \end{bmatrix}_q = \begin{bmatrix} k, 1 \end{bmatrix}_q$ inequalities, since $\mathrm{PG}(k-1, q)$ has that many hyperplanes. We can represent these inequalities in a system

$$A\mathbf{x} \geq \mathbf{b} \tag{5}$$

for a matrix A of suitable size and vectors $\mathbf{x}$ and $\mathbf{b}$ of length $\begin{bmatrix} k, 1 \end{bmatrix}_q$. Since $(X_i)_{i=1}^k$ is a non-increasing sequence of integers, we have

$$|\{d \mid b_d = X_i\}| = \begin{bmatrix} k-i+1, 1 \end{bmatrix}_q - \begin{bmatrix} k-i, 1 \end{bmatrix}_q$$

for all $1 \leq i \leq k$. Moreover, following the notation of Theorem 2, each n_d occurs in exactly $|D(H)|$ inequalities.

Let $Z = \begin{bmatrix} k-i+1, 1 \end{bmatrix}_q - \begin{bmatrix} k-i, 1 \end{bmatrix}_q$ for the rest of the proof. Summing all the inequalities in (5) and dividing by $|D(H)|$ gives

$$\begin{aligned} n_q(S) &\geq \frac{1}{|D(H)|} \sum_{i=1}^{k} Z\lceil X_i \rceil = \frac{1}{|D(H)|} \sum_{i=1}^{k} ZX_i \\ &= \frac{1}{\begin{bmatrix} k, 1 \end{bmatrix}_q - \begin{bmatrix} k-1, 1 \end{bmatrix}_q} \sum_{i=1}^{k} ZX_i \\ &= X_1 + \frac{q-1}{q^k - q^{k-1}} \sum_{i=2}^{k} ZX_i \\ &= X_1 + \frac{X_2}{q} + \ldots + \frac{X_k}{q^{k-1}}. \end{aligned}$$

Taking the ceiling, we bound on the integer $n_q(S)$. □

When $q = 2$, we can establish an upper bound for $n_q(S)$, where S has the same properties stated in Theorem 4. The proof of the following result is constructive and will appear in the extended version of this work because of space constraints.

Theorem 5. *Let $(X_i)_{i=1}^k$ be a non-increasing sequence of non-negative integers and $S = \{X_i e_i \mid 1 \leq i \leq k\}$. We have*

$$n_2(S) \leq \sum_{i=1}^{k} \left\lceil \frac{X_i}{2^{i-1}} \right\rceil.$$

By combining Theorems 4 and 5 we obtain the following result for the case $q = 2$.

Corollary 1. *Let $(X_i)_{i=1}^k$ be a non-increasing sequence of non-negative integers and let $S = \{X_i e_i \mid 1 \leq i \leq k\}$. Then*

$$\left\lceil \sum_{i=1}^{k} \frac{X_i}{2^{i-1}} \right\rceil \leq n_2(S) \leq \sum_{i=1}^{k} \left\lceil \frac{X_i}{2^{i-1}} \right\rceil.$$

Note that the lower and upper bounds of Corollary 1 coincide when 2^{i-1} divides X_i for all $i \in \{2, \ldots, k\}$. Moreover, if $X_i = 2^{k-1}$ for all $i \in \{1, \ldots, k\}$, then the well-known binary simplex code (of length $2^k - 1$) attains $n_2(S)$.

V. Applications and Examples

We provide evidence of how the established results can be applied in concrete scenarios. The proofs of the results in this paper use different techniques, which capture different mathematical properties of the quantities $n_q(S)$ and $q_n(S)$. As a natural consequence, the results perform best when combined. For instance, Theorem 3 together with Proposition 4 can be used to compute the value of $n_q(S)$ in some cases.

Example 6. Let $S = \{(2.5, 1), (1, 2)\}$. Following the notation of Theorem 3, we have $\alpha(S) = 4$ and $\beta(S) = 5$. Therefore, $4 \leq n_q(S) \leq 5$. The number $q_4(S)$ does not exist for any $q \in \mathscr{P}$ by Proposition 4. Thus $n_q(S)$ has to be equal to 5 for every $q \in \mathscr{P}$.

A general observation is that bounds on $n_q(S)$ that depend on q automatically give information on $q_n(S)$. That is the case of Theorem 4, which implies the following result.

Corollary 2. *Let $(X_i)_{i=1}^k$ be a non-increasing sequence of non-negative integers and let $S = \{X_i e_i \mid 1 \leq i \leq k\}$. For any $n \in \mathbb{Z}_{\geq k}$ s.t. $q_n(S)$ exists, we have*

$$q_n(S) \geq \frac{X_2}{n - X_1}.$$

Proof. Let $n \in \mathbb{Z}_{\geq k}$ s.t. $q_n(S)$ exists. By Theorem 4, we have

$$n \geq \sum_{i=1}^{k} \lceil q^{1-i} X_i \rceil \geq X_1 + \frac{X_2}{q},$$

yielding the result. □

Even though Corollary 2 is a quite coarse approximation of Theorem 4, it provides very good bounds in some instances.

Example 7. Let $S = \{(100, 0), (0, 99)\}$. Corollary 2 gives $q_{100}(S) \geq 99$ which implies that $q_{100}(S) \geq 101$ since it must be a prime power. It can also be checked that $n_2(S) = 150$ in this example.

We conclude the paper by showing that Corollary 2 can actually be met with equality for some parameter sets.

Example 8. Let $S = \{(2, 0), (0, 2)\}$ and $n = 3$. Then Corollary 2 implies that $q_3(S) \geq 2$ and it is met with equality, since $S \subseteq \Lambda(G)$ for

$$G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{2 \times 3}.$$

Conclusions and Future Work

We addressed the problem of computing the minimum number of servers and minimum alphabet size to support a demand vector for the service rate region problem. We proposed a method to obtain the exact values of the first parameter based on optimization. We then established upper and lower bounds for both parameters, studied their interplay, and discussed their sharpness. Future work includes sharpening the bounds and investigating more general sets of demand vectors than those considered in this paper.

REFERENCES

[1] M. Aktaş, S. E. Anderson, A. Johnston, G. Joshi, S. Kadhe, G. L. Matthews, C. Mayer, and E. Soljanin, "On the service capacity region of accessing erasure coded content," in *2017 55th Annual Allerton Conf. on Communication, Control, and Computing (Allerton'17)*, 2017.

[2] M. Aktaş, G. Joshi, S. Kadhe, F. Kazemi, and E. Soljanin, "Service rate region: A new aspect of coded distributed system design," *IEEE Trans. on Information Theory*, vol. 67, no. 12, pp. 7940–7963, 2021.

[3] F. Kazemi, E. Karimi, E. Soljanin, and A. Sprintson, "A combinatorial view of the service rates of codes problem, its equivalence to fractional matching and its connection with batch codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020.

[4] F. Kazemi, S. Kurz, and E. Soljanin, "A geometric view of the service rates of codes problem and its application to the service rate of the first order reed-muller codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 66–71, 2020.

[5] G. N. Alfarano, A. Ravagnani, and E. Soljanin, "Dual-code bounds on multiple concurrent (local) data recovery," in *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2613–2618, 2022.

[6] G. Joshi, Y. Liu, and E. Soljanin, "Coding for fast content download," in *Proceedings of the Allerton Conference on Comm., Control and Computing*, pp. 326–333, Oct. 2012.

[7] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyytiä, and A. Scheller-Wolf, "Reducing latency via redundant requests: Exact analysis," in *Proceedings of the ACM SIGMETRICS*, Jun. 2015.

[8] N. B. Shah, K. Lee, and K. Ramchandran, "When do redundant requests reduce latency?," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 715–722, 2016.

[9] K. V. Rashmi, M. Chowdhury, J. Kosaian, I. Stoica, and K. Ramchandran, "Ec-cache: Load-balanced, low-latency cluster caching with online erasure coding," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, (Savannah, GA), pp. 401–417, 2016.

[10] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2017.

[11] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," in *Advances In Neural Information Processing Systems*, pp. 2100–2108, 2016.

[12] A. Mallick, U. Sheth, G. Palanikumar, M. Chaudhari, and G. Joshi, "Rateless Codes for Near-Perfect Load Balancing in Distributed Matrix-Vector Multiplication," in *ACM Sigmetrics 2020*, May 2020.

[13] P. Peng, M. Noori, and E. Soljanin, "Distributed storage allocations for optimal service rates," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6647–6660, 2021.

[14] P. Peng and E. Soljanin, "On distributed storage allocations of large files for maximum service rate," in *56th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2018, Monticello, IL, USA, October 2-5, 2018*, pp. 784–791, IEEE, 2018.

[15] F. Kazemi, S. Kurz, E. Soljanin, and A. Sprintson, "Efficient storage schemes for desired service rate regions," in *2020 IEEE Information Theory Workshop (ITW)*, 2021.

[16] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, vol. 16. Elsevier, 1977.

[17] G. N. Alfarano, A. B. Kılıç, A. Ravagnani, and E. Soljanin, "The service rate region polytope," *to appear in SIAGA*, 2024.

[18] G. M. Ziegler, *Lectures on polytopes*, vol. 152. Springer Science & Business Media, 2012.

[19] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[20] G. E. Andrews, *The theory of partitions*. No. 2, Cambridge university press, 1998.