

# Efficient Storage Schemes for Desired Service Rate Regions

\*Fatemeh Kazemi, <sup>†</sup>Sascha Kurz, <sup>‡</sup>Emina Soljanin, and \*Alex Sprintson

\*Dept. of ECE, Texas A&M University, USA (E-mail: {fatemeh.kazemi, spalex}@tamu.edu)

<sup>†</sup> Dept. of Mathematics, University of Bayreuth, Germany (E-mail: sascha.kurz@uni-bayreuth.de)

<sup>‡</sup>Dept. of ECE, Rutgers University, USA (E-mail: emina.soljanin@rutgers.edu)

**Abstract**—A major concern in cloud/edge storage systems is serving a large number of users simultaneously. The service rate region is introduced recently as an important performance metric for coded distributed systems, which is defined as the set of all data access requests that can be simultaneously handled by the system. This paper studies the problem of designing a coded distributed storage system storing  $k$  files where a desired service rate region  $\mathcal{R}$  of the system is given and the goal is 1) to determine the minimum number of storage nodes  $n(\mathcal{R})$  for serving all demand vectors inside the set  $\mathcal{R}$  and 2) to design the most storage-efficient redundancy scheme with the service rate region covering the set  $\mathcal{R}$ . Towards this goal, we propose three general lower bounds for  $n(\mathcal{R})$ . Also, for  $k = 2$ , we characterize  $n(\mathcal{R})$ , i.e., we show that the proposed lower bounds are tight, via designing a novel storage-efficient redundancy scheme with  $n(\mathcal{R})$  storage nodes and the service rate region covering  $\mathcal{R}$ .

## I. INTRODUCTION

**Motivation:** The past two decades have seen an explosive growth in the amount of data stored in the cloud data centers which was accompanied by a rapid increase in the volume of users accessing it. To handle these ever-increasing demands in a fast and reliable manner, chunks of a data object are stored redundantly over multiple storage nodes through either replication or erasure coding. Although replication has been typically preferred due its simplicity, it can be expensive in terms of storage. Erasure codes have been shown to be effective in achieving various goals such as providing reliability against node failures (e.g., [1]), ensuring availability of stored content during high demand (e.g., [2]), enabling the recovery of a data object from multiple disjoint groups of nodes (e.g., [3]), and providing fast content download (e.g., [4]–[6]).

Serving a large number of users simultaneously is a major concern in cloud storage systems and is considered as one of the most significant considerations in the design of coded distributed systems. The service rate region has been recently recognized as an important performance metric for coded distributed systems which is the set of all data access requests that can be simultaneously served by the system [7]–[13]. It has been shown that erasure coding of data objects can increase the overall volume of the service rate region through handling skews in the request rates more flexibly [7]–[9].

Part of this research is based upon work supported by the National Science Foundation under Grants No. CIF-1717314, as well as work while some authors were in residence at the Schloss Dagstuhl Research Institute during the Algebraic Coding Theory for Networks, Storage, and Security Seminar in 2018. The authors thank Dr. Y. Zhang for helpful discussion.

The service rate problem considers a distributed storage system in which  $k$  files  $f_1, \dots, f_k$  are stored across  $n$  servers using a linear  $[n, k]_q$  code. The requests to download file  $f_i$  arrive at rate  $\lambda_i$ , and the service rate of each server is  $\mu$ . A goal of the service rate problem is to determine the service rate region of this system which is the set of all request rates  $\lambda = (\lambda_1, \dots, \lambda_k)$  that can be served by this system.

**Previous Work:** All the existing studies on the service rate problem focus on characterizing the service rate region of a given coded storage scheme and finding the optimal request allocation, that is, the optimal policies for splitting incoming requests across the nodes to maximize the service rate region (see [7]). In [8], the service rate region was characterized for MDS codes when  $n \geq 2k$ , binary simplex codes and systems with arbitrary  $n$  when  $k = 2$ . The service rate region of the systems with arbitrary  $n$  when  $k = 3$  was determined in [9]. A connection between the service rate problem and the fractional matching problem is established in [10]. Also, it has been shown that the service rate problem can be viewed as a generalization of the batch codes problem. In [11], we characterized the service rate regions of the binary first order Reed-Muller codes and binary simplex codes using a novel geometric technique. Also, we showed that given the service rate region of a code, a lower bound on the minimum distance of the code can be derived.

**Main Contributions:** In this paper, we consider a practical setting of designing a coded distributed storage system where we are asked to store  $k$  files redundantly across some number of storage nodes in the system. Also, we are given a bounded subset  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$  as a desired service rate region for this distributed storage system. Our goal is: 1) to find the minimum number of storage nodes  $n(\mathcal{R})$  (or a lower bound on  $n(\mathcal{R})$ ) required for serving all demand vectors  $\lambda$  inside the desired service rate region  $\mathcal{R}$ , and 2) to design the most storage-efficient redundancy scheme with service rate region covering the set  $\mathcal{R}$ . In fact, in this paper, unlike the existing work, we focus on designing the underlying erasure code for covering a given service rate region with minimum storage. Towards this goal, we present three different lower bounds for  $n(\mathcal{R})$ . Also, we show that for  $k = 2$ , these bounds are tight and we design an efficient storage scheme that achieves the desired service rate region while minimizing the storage. *Due to the space constraints all the proofs are omitted and can be found in [14].*

## II. PROBLEM SETUP AND FORMULATION

### A. Basic Notation

Throughout this paper, we denote vectors by bold-face lower-case letters and matrices by bold-face capital letters. Let  $\mathbb{Z}_{\geq 0}$  and  $\mathbb{N}$ , respectively, denote the set of non-negative integers, and the set of positive integers. For  $k \in \mathbb{N}$ , let  $\mathbf{0}_k$  and  $\mathbf{1}_k$ , respectively, denote the all-zero and all-one column vectors of length  $k$ . Let  $\mathbf{e}_i$  be a unit vector of length  $k$ , having a one at position  $i$  and zeros elsewhere. For any  $i \in \mathbb{N}$ , we define  $[i] \triangleq \{1, \dots, i\}$ . Let  $\mathbb{F}_q$  be the finite field of order  $q$ , and  $\mathbb{F}_q^n$  be the  $n$ -dimensional vector space over  $\mathbb{F}_q$ . Let  $[n, k]_q$  denote a  $q$ -ary linear code  $\mathcal{C}$  of length  $n$  and dimension  $k$ . We denote the cardinality of a set or multiset  $\mathcal{S}$  by  $\#\mathcal{S}$ . Let  $\langle \mathcal{S} \rangle$  and  $\text{conv}(\mathcal{S})$ , respectively, denote the span and the convex hull of the set  $\mathcal{S}$  of vectors. For two vectors  $\mathbf{x} = (x_1, \dots, x_k)$  and  $\mathbf{y} = (y_1, \dots, y_k)$ , let  $\mathbf{x} \leq \mathbf{y}$  define  $x_i \leq y_i$  for all  $i \in [k]$ .

### B. Coded Storage System

Consider a coded storage system wherein  $k$  files  $f_1, \dots, f_k$  are stored redundantly across  $n$  servers using a linear code of length  $n$  and dimension  $k$  over  $\mathbb{F}_q$  with generator matrix  $\mathbf{G}$ . Suppose all files are of the same size, and all servers have a storage capacity of one file. A set  $Y$  is a recovery set for file  $f_i$  if the unit vector  $\mathbf{e}_i$  can be recovered through a linear combination of the columns of  $\mathbf{G}$  indexed by the set  $Y$ , i.e., if there exist coefficients  $\alpha_j$ 's  $\in \mathbb{F}_q$  such that  $\sum_{j \in Y} \alpha_j \mathbf{g}_j = \mathbf{e}_i$  where  $\mathbf{g}_j$  denotes the  $j$ th column of  $\mathbf{G}$ . For each file, w.l.o.g. we consider reduced recovery sets defined as the recovery sets that are not a proper superset of any other recovery sets for that file. In other words, the reduced recovery sets are obtained by considering non-zero coefficients  $\alpha_j$ 's and linearly independent columns  $\mathbf{g}_j$ 's. Let  $\mathcal{Y}_i = \{Y_{i,1}, \dots, Y_{i,t_i}\}$  denote the  $t_i$  recovery sets for file  $f_i$ .

We assume that the service rate of each server is  $\mu$ , i.e., each server can resolve the received requests at the average rate  $\mu$ . We further assume that the requests to download file  $f_i$  arrive at rate  $\lambda_i$ ,  $i \in [k]$ . The request arrival rates for the  $k$  files are denoted by the demand vector  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_k)$ . We consider the class of scheduling strategies that assign a fraction of requests for a file to each of its recovery sets. Let  $\lambda_{i,j}$  be the portion of requests for file  $f_i$  that is assigned to the recovery set  $Y_{i,j}$ ,  $j \in [t_i]$ .

### C. Service Rate Region

The demand vector  $\boldsymbol{\lambda}$  can be served by a coded distributed storage system with generator matrix  $\mathbf{G} \in \mathbb{F}_q^{k \times n}$  and service rate  $\mu$  iff there exists a set  $\{\lambda_{i,j} : i \in [k], j \in [t_i]\}$ , referred to as a valid allocation, that satisfies the constraints below:

$$\sum_{j=1}^{t_i} \lambda_{i,j} = \lambda_i, \quad \text{for all } i \in [k], \quad (1a)$$

$$\sum_{i=1}^k \sum_{\substack{j \in [t_i] \\ \ell \in Y_{i,j}}} \lambda_{i,j} \leq \mu, \quad \text{for all } \ell \in [n], \quad (1b)$$

$$\lambda_{i,j} \in \mathbb{R}_{\geq 0}, \quad \text{for all } i \in [k], j \in [t_i]. \quad (1c)$$

The constraints (1a) guarantee that the demands for all files are served, and the constraints (1b) ensure that the total rates assigned to each server is not more than its service rate.

The *service rate region* of an erasure coded storage system with the generator matrix  $\mathbf{G}$  and service rate  $\mu$ , denoted by  $\mathcal{S}(\mathbf{G}, \mu) \subseteq \mathbb{R}_{\geq 0}^k$ , is defined as the set of all demand vectors  $\boldsymbol{\lambda}$  that can be served by the system. In what follows, w.l.o.g. we assume that  $\mu = 1$  and abbreviate  $\mathcal{S}(\mathbf{G}, 1)$  as  $\mathcal{S}(\mathbf{G})$ .

Note that there are several generator matrices that span the same linear code, i.e., whenever the row span of two matrices  $\mathbf{G}$  and  $\mathbf{G}'$  coincides, they span the same code. However, the service rate regions of generator matrices  $\mathbf{G}$  and  $\mathbf{G}'$  of the same linear code might not be the same, i.e.,  $\mathcal{S}(\mathbf{G}) \neq \mathcal{S}(\mathbf{G}')$ .

### D. Geometric description of Linear Codes

Here, we briefly review some preliminaries regarding the notions of projective space, multiset, and projective multisets induced by linear codes that we will use in Sec.II-E. For more details, see e.g., [15], [16].

**Definition 1.** For a vector space  $\mathcal{V}$  of dimension  $v$  over  $\mathbb{F}_q$ , the *projective space* of  $\mathcal{V}$ , denoted as  $\text{PG}(\mathcal{V})$ , is the set of equivalence classes of  $\mathcal{V} \setminus \{\mathbf{0}_v\}$  under the equivalence relation  $\sim$  defined as  $x \sim y$  if there is a non-zero element  $\alpha \in \mathbb{F}_q$  such that  $x = \alpha y$ .

Note that the 1-dimensional subspaces of  $\mathcal{V}$  are the points of the projective space  $\text{PG}(\mathcal{V})$ . The 2-dimensional subspaces of  $\mathcal{V}$  are the lines of  $\text{PG}(\mathcal{V})$  and the  $v-1$  dimensional subspaces of  $\mathcal{V}$  are called the hyperplanes of  $\text{PG}(\mathcal{V})$ .

For a vector space  $\mathcal{V}$  of (geometric) dimension  $v$  over  $\mathbb{F}_q$ , the projective space  $\text{PG}(\mathcal{V})$  is also denoted by  $\text{PG}(v-1, q)$ , referred to as projective space of (algebraic) dimension  $v-1$  over  $\mathbb{F}_q$ . This notion makes sense since up to isomorphism, the  $\text{PG}(\mathcal{V})$  only depends on the order  $q$  of the base field and the dimension  $v$  of the vector space  $\mathcal{V}$ . Thus,  $\text{PG}(v-1, q)$  can be defined as the set of  $v$ -tuples of elements of  $\mathbb{F}_q$ , not all zero, under the equivalence relation given by  $(x_1, \dots, x_v) \sim (\alpha x_1, \dots, \alpha x_v)$ ,  $\alpha \neq 0$ ,  $\alpha \in \mathbb{F}_q$ . The definition implies that if  $(x_1, \dots, x_v)$  is a point in  $\text{PG}(v-1, q)$ , its scalar multiple (by any non-zero scalar  $\alpha \in \mathbb{F}_q$ )  $(\alpha x_1, \dots, \alpha x_v)$  is the same point in  $\text{PG}(v-1, q)$ .

A *multiset*, unlike a set, allows for multiple instances for each of its elements. A multiset  $\mathcal{S}$  on a base set  $\mathcal{X}$  is defined with its characteristic function, denoted as  $\chi_{\mathcal{S}} : \mathcal{X} \rightarrow \mathbb{N}$ , mapping  $x \in \mathcal{X}$  to the multiplicity of  $x$  in  $\mathcal{S}$ . The cardinality of  $\mathcal{S}$  is computed as  $\#\mathcal{S} = \sum_{x \in \mathcal{X}} \chi_{\mathcal{S}}(x)$ . The multiset  $\mathcal{S}$  is also called *# $\mathcal{S}$ -multiset*. As a simple example, consider the multiset  $\mathcal{S} = \{a, a, b, b, b, c\}$  on the base set  $\mathcal{X} = \{a, b, c\}$  that is identified with  $\chi_{\mathcal{S}}(a) = 2$ ,  $\chi_{\mathcal{S}}(b) = 3$  and  $\chi_{\mathcal{S}}(c) = 1$ .

Let  $\mathbf{G}$  be the generator matrix of an  $[n, k]_q$  code  $\mathcal{C}$  that is a  $k$ -dimensional subspace of the  $n$ -dimensional vector space  $\mathbb{F}_q^n$ . Let  $\mathbf{g}_i$ ,  $i \in [n]$  be the  $i$ th column of  $\mathbf{G}$ . Then, each  $\mathbf{g}_i$  is a point in the projective space  $\text{PG}(k-1, q)$ , and  $\mathcal{G} := \{g_1, g_2, \dots, g_n\}$  is an  $n$ -multiset of points in  $\text{PG}(k-1, q)$  where each point is counted with the appropriate multiplicity. In general,  $\mathcal{G}$  is called the  $n$ -multiset induced by  $\mathcal{C}$ .

**Proposition 1.** *There exists a one-to-one correspondence between the equivalence classes of full-length  $q$ -ary linear codes and the projective equivalence classes of multisets in finite projective spaces.*

An  $[n, k]_q$  code can be described by a generator matrix  $\mathbf{G}$  or as discussed by an  $n$ -multiset  $\mathcal{G}$  of points in  $\text{PG}(k-1, q)$ . In what follows, for the ease of notation, we restrict ourselves to the binary field. We associate the points of  $\text{PG}(k-1, 2)$  with the non-zero vectors in  $\mathbb{F}_2^k \setminus \{\mathbf{0}_k\}$ , then we interpret each such vector as the binary expansion of the corresponding integer  $i \in [\ell]$  where  $\ell := 2^k - 1$ . We denote by  $v_i$  the vector corresponding to the integer  $i \in [\ell]$ . As two examples, in  $\mathbb{F}_2^3 \setminus \{\mathbf{0}_3\}$ , the vectors  $v_3 = (0, 1, 1)$  and  $v_4 = (1, 0, 0)$  are corresponding to the integers 3 and 4, respectively. In order to uniquely characterize a multiset of points  $\mathcal{G}$  in  $\text{PG}(k-1, 2)$ , we use multiplicities  $n_i \in \mathbb{Z}_{\geq 0}$ ,  $i \in [\ell]$ , counting the number of occurrences of the vector  $v_i \in \mathbb{F}_2^k \setminus \{\mathbf{0}_k\}$ ,  $i \in [\ell]$ , in the generator matrix  $\mathbf{G}$ . Thus, we have  $\sum_{i \in [\ell]} n_i = n$ . Also, due to the correspondence between a generator matrix  $\mathbf{G}$  and a multiset of points  $\mathcal{G}$ , we write  $\mathcal{S}(\mathcal{G})$  instead of  $\mathcal{S}(\mathbf{G})$  for the service rate region and we will directly define  $\mathcal{S}(\mathcal{G})$  later on.

### E. Geometric Interpretation of the Service Rate Region

A recovery set for file  $f_i$ ,  $i \in [k]$ , is a subset  $Y \subseteq [\ell]$  such that the span of the set  $\{v_j \mid j \in Y\}$  contains the unit vector  $\mathbf{e}_i$ . A recovery set  $Y$  for  $f_i$  is called reduced if there does not exist a proper subset  $Y' \subsetneq Y$  with  $\mathbf{e}_i \in \langle \{v_j \mid j \in Y'\} \rangle$ . For  $q = 2$  and a reduced recovery set  $Y$ , there is no need to specify the index  $i$  of the file that is recovered since  $\sum_{j \in Y} v_j = \mathbf{e}_i$ . However, this is not necessarily true for  $q > 2$ . As an example, in  $\mathbb{F}_3$  the set  $\{\mathbf{e}_1 + \mathbf{e}_2, \mathbf{e}_1 + 2\mathbf{e}_2\}$  spans a 2-dimensional subspace containing both  $\mathbf{e}_1$  and  $\mathbf{e}_2$ , while none of these two unit vectors are contained in the span of a proper subset. Since we assume  $q = 2$ , we will mostly speak just of a recovery set without specifying the index  $i$  of the file that it recovers. By  $\mathcal{Y}_i$  we denote the set of all reduced recovery sets for file  $f_i$ , where  $i \in [k]$ . As an example, for  $k = 3$  we have  $\mathcal{Y}_2 = \{\{2\}, \{4, 6\}, \{1, 3\}, \{5, 7\}, \{1, 4, 7\}\}$ . Note that the maximum cardinality of a reduced recovery set is  $k$ , which can indeed be attained.

Let  $\alpha_{i,Y}$  be the portion of request rates for file  $f_i$  assigned to the recovery set  $Y \in \mathcal{Y}_i$ . Given a multiset of points  $\mathcal{G}$  in  $\text{PG}(k-1, 2)$ , described by the multiplicities  $n_j$ ,  $j \in [\ell]$ , the service rate region  $\mathcal{S}(\mathcal{G})$  is the set of all vectors  $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^k$  for which there exist  $\alpha_{i,Y}$ 's, satisfying the following:

$$\sum_{Y \in \mathcal{Y}_i} \alpha_{i,Y} = \lambda_i, \quad \text{for all } i \in [k], \quad (2a)$$

$$\sum_{i=1}^k \sum_{\substack{Y \in \mathcal{Y}_i \\ j \in Y}} \alpha_{i,Y} \leq n_j, \quad \text{for all } j \in [\ell], \quad (2b)$$

$$\alpha_{i,Y} \in \mathbb{R}_{\geq 0}, \quad \text{for all } i \in [k], Y \in \mathcal{Y}_i. \quad (2c)$$

Recall that the constraints (2a) guarantee that the demands for all files are served, and constraints (2b) certify that no node receives requests at a rate in excess of its service rate.

As noted, for  $q = 2$ , each reduced recovery set uniquely characterizes the file it recovers, that is,  $\mathcal{Y}_i$ 's where  $i \in [k]$  are pairwise disjoint and form a partition of  $\mathcal{Y} := \cup_{i \in [k]} \mathcal{Y}_i$ . With this we can simplify the above characterization, i.e., the service rate region  $\mathcal{S}(\mathcal{G})$  is the set of all vectors  $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^k$  for which there exists  $\alpha_Y$ , satisfying the following constraints:

$$\sum_{Y \in \mathcal{Y}_i} \alpha_Y \geq \lambda_i, \quad \text{for all } i \in [k], \quad (3a)$$

$$\sum_{\substack{Y \in \mathcal{Y} \\ j \in Y}} \alpha_Y \leq n_j, \quad \text{for all } j \in [\ell], \quad (3b)$$

$$\alpha_Y \in \mathbb{R}_{\geq 0}, \quad \text{for all } i \in [k], Y \in \mathcal{Y}_i. \quad (3c)$$

### F. Problem Statement

After these preparations, we can state the problems that we explore to address in this paper. Consider a practical scenario where we are asked to store  $k$  files redundantly across some number of nodes in a coded distributed storage system. Also, we are given a bounded subset  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$  as a desired service rate region for this distributed storage system. Two natural questions arising in the design of this storage system are the following: 1) What is the minimum number  $n(\mathcal{R})$  of storage nodes (or servers) required for serving all demand vectors  $\boldsymbol{\lambda}$  inside the desired service rate region  $\mathcal{R}$ ? 2) What is the most storage-efficient redundancy scheme with service rate region covering the set  $\mathcal{R}$  (i.e., how should the files be stored redundantly in  $n(\mathcal{R})$  storage nodes)?

For each desired service rate region  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$ , the goal is to characterize (or derive a lower bound on) the minimum number of nodes  $n(\mathcal{R})$  such that there exists a generator matrix  $\mathbf{G}$  with  $\mathcal{R} \subseteq \mathcal{S}(\mathbf{G})$  (or alternatively, a multiset of points  $\mathcal{G}$  in  $\text{PG}(k-1, q)$  with  $\mathcal{R} \subseteq \mathcal{S}(\mathcal{G})$ ). Thus, deriving lower bounds and constructive upper bounds for  $n(\mathcal{R})$  is of great significance in the context of designing distributed storage systems, which we aim to address in this paper.

## III. MAIN RESULTS

In this section, first we investigate a few structural properties and formulate the problem of determining  $n(\mathcal{R})$ . Then, using a geometric approach, we derive multiple lower bounds on  $n(\mathcal{R})$  and finally we show that for  $k = 2$  the derived lower bounds are tight by proposing an storage-efficient scheme.

### A. Structural Properties of the Service Rate Region

Here, before we present integer linear programming (ILP) formulations for the determination of  $n(\mathcal{R})$  we first study a few structural properties.

**Lemma 1.** *For  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$ , we have  $n(\mathcal{R}) = n(\text{conv}(\mathcal{R}))$ .*

**Definition 2.** *For a set  $S \subseteq \mathbb{R}_{\geq 0}^k$ , the lower set  $S\downarrow$  is defined as  $S\downarrow := \{\mathbf{x} \in \mathbb{R}_{\geq 0}^k \mid \exists \mathbf{y} \in S : \mathbf{x} \leq \mathbf{y}\}$ .*

**Lemma 2.** *For a subset  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$ , we have  $n(\mathcal{R}) = n(\mathcal{R}\downarrow)$ .*

Taken the above two observations into account, we want to parameterize a large class of reasonable subsets  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$  through a function  $T: 2^{[k]} \rightarrow \mathbb{N}$  that maps the subsets of  $[k]$  to integers in  $\mathbb{N}$ , where  $T(\emptyset) = 0$ .

**Definition 3.** Let  $T: 2^{[k]} \rightarrow \mathbb{N}$  with  $T(\emptyset) = 0$ . We define

$$\mathcal{R}(T) := \left\{ \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^k \mid \sum_{i \in S} \lambda_i \leq T(S) \forall \emptyset \neq S \subseteq [k] \right\}$$

By construction  $\mathcal{R}(T)$  is a convex polytope and  $\mathcal{R}(T) \downarrow = \mathcal{R}(T)$ , i.e.,  $\mathcal{R}(T)$  is its own lower set. (See e.g., [11].) Note that in some cases, the values of the function  $T: 2^{[k]} \rightarrow \mathbb{N}$  can be modified without changing  $\mathcal{R}(T)$ .

**Lemma 3.** For each function  $T: 2^{[k]} \rightarrow \mathbb{N}$ , with  $T(\emptyset) = 0$ , there exists a monotone and subadditive function  $T': 2^{[k]} \rightarrow \mathbb{N}$ , with  $T'(\emptyset) = 0$ , such that  $\mathcal{R}(T) = \mathcal{R}(T')$ .<sup>1</sup>

**Definition 4.** For a subset  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$  with property  $\mathcal{R} \downarrow = \mathcal{R}$ , we say that a finite set  $S \subset \mathbb{R}_{\geq 0}^k$  is a generating set of  $\mathcal{R}$  if  $\text{conv}(S) \downarrow = \mathcal{R}$ . Moreover, we call  $S$  minimal if no proper subset of  $S$  is a generating set of  $\mathcal{R}$ .

Note that in what follows, without explicitly mentioning, we consider the minimal generating sets for each  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$ . As an example, consider the function  $T: 2^{[2]} \rightarrow \mathbb{N}$  given by  $T(\emptyset) = 0$ ,  $T(\{1\}) = T(\{2\}) = 2$ , and  $T(\{1, 2\}) = 3$ . Here, a generating set of  $\mathcal{R}(T)$  is given by  $\{(1, 2), (2, 1)\}$ . Note that the generating set of  $\mathcal{R}(T)$  is always unique, since  $\mathcal{R}(T)$  is a polytope that can be written as  $\mathcal{R}(T) = \text{conv}(V)$ , where  $V$  is the set of vertices of the polytope. The generating set of  $\mathcal{R}(T)$  is obtained from  $V$  by removing all vectors  $\mathbf{v} \in V$  such that there exists a vector  $\mathbf{v}' \in V$  with  $\mathbf{v} \leq \mathbf{v}'$ . Next, we present an ILP formulation for the determination of  $n(\mathcal{R})$ .

**Proposition 2.** For a desired service rate region  $\mathcal{R} \subset \mathbb{R}_{\geq 0}^k$ , assume  $\mathcal{R} \downarrow = \mathcal{R}$ . Let  $\{\boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(m)}\}$  be the generating set of  $\mathcal{R}$ . Then,  $n(\mathcal{R})$  coincides with the optimal target value of

$$\begin{aligned} \min \quad & \sum_{j \in [\ell]} n_j & (4) \\ \text{s.t.} \quad & \sum_{Y \in \mathcal{Y}^j} \alpha_Y^i \geq \lambda_j^{(i)} & \forall i \in [m], j \in [k] \\ & \sum_{\substack{Y \in \mathcal{Y} \\ j \in Y}} \alpha_Y^i \leq n_j, & \forall j \in [\ell], \forall i \in [m] \\ & \alpha_Y^i \in \mathbb{R}_{\geq 0}, & \forall i \in [m], \forall Y \in \mathcal{Y} \\ & n_j \in \mathbb{N}, & \forall j \in [\ell] \end{aligned}$$

where  $\lambda_j^{(i)}$  is the  $j$ th element of the  $\boldsymbol{\lambda}^{(i)}$  and  $\alpha_Y^i$  is the portion of requests coming from  $\boldsymbol{\lambda}^{(i)}$  assigned to the recovery set  $Y$ .

The ILP formulation (4) underlies a massive combinatorial explosion. Indeed, when the number of files  $k$  increases, the number of recovery sets  $\#\mathcal{Y}$  grows doubly exponential, i.e.,  $\#\mathcal{Y}$  gets quite large even for moderate values of  $k$ . To obtain a lower bound on  $n(\mathcal{R})$ , one simple way is to consider the ceiling of the optimal target value for the LP relaxation of the ILP (4). However, this approach again suffers from the same drawback and runs into a similar problem since to list all the constraints of the LP relaxation of the ILP (4), one needs to

<sup>1</sup>A function  $T: 2^{[k]} \rightarrow \mathbb{N}$  is monotone iff  $T(U) \leq T(V)$  holds for all  $\emptyset \subseteq U \subseteq V \subseteq [k]$ , and is subadditive iff  $T(U \cup V) \leq T(U) + T(V)$ .

explicitly know all possible recovery sets which becomes increasingly complex when the number of files  $k$  increases. Thus, introducing a technique which is not depending on the enumeration of recovery sets is of great significance. Towards this goal, we introduce a novel geometric approach.

**B. Using Geometric Approach to derive Bounds on  $n(\mathcal{R})$**

Here, we present three lower bounds for  $n(\mathcal{R}(T))$  that are obtained using a geometric technique.

**Lemma 4.** Let  $\mathbf{G}$  be the generator matrix of an  $[n, k]_q$  code and  $\mathcal{G}$  be the corresponding multiset of points of cardinality  $n$  which is described by point multiplicities  $n_j$  where  $j \in [\ell]$ . If  $\{\boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(m)}\}$  be the generating set of  $\mathcal{R}$ , then we have

$$\sum_{j: \mathbf{v}_j \in \text{PG}(k-1, 2) \setminus \mathcal{H}} n_j \geq \max \left\{ \sum_{s \in \mathcal{E}(\mathcal{H})} \lambda_s^{(i)} \mid i \in [m] \right\}, \quad (5)$$

where  $\mathcal{H}$  is a hyperplane of  $\text{PG}(k-1, 2)$  and

$$\mathcal{E}(\mathcal{H}) = \{h \in [k] \mid \mathbf{e}_h \notin \langle \{\mathbf{v} \mid \mathbf{v} \in \mathcal{H}\} \rangle\}$$

is the set of indices  $h$  such that the hyperplane  $\mathcal{H}$  does not contain the unit vector  $\mathbf{e}_h$ , i.e.,  $\mathbf{e}_h$  lies in  $\text{PG}(k-1, 2) \setminus \mathcal{H}$ .

**Corollary 1.** If  $\{\boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(m)}\}$  is the generating set of  $\mathcal{R}$ , then  $n(\mathcal{R})$  is lower bounded by the optimal target value of

$$\begin{aligned} \min \quad & \sum_{j \in [\ell]} n_j & (6) \\ \text{s.t.} \quad & (5) \text{ holds } \forall \text{ hyperplane } \mathcal{H} \text{ of } \text{PG}(k-1, 2) \\ & n_j \in \mathbb{N} \quad \forall j \in [\ell]. \end{aligned}$$

Note that the ILP of Corollary 1 contains  $2^k - 1$  constraints and (integer) variables. So, with respect to the LP relaxation of the ILP (4), we have obtained a smaller formulation for the determination of a lower bound on  $n(\mathcal{R})$ .

**Definition 5.** Consider the  $P = \{\mathbf{x} \in \mathbb{R}^k \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$  as a polytope in  $\mathbb{R}^k$  with description  $(\mathbf{A}, \mathbf{b})$ . We say that a constraint  $\mathbf{a}^{(i)}\mathbf{x} \leq b_i$  is redundant, where  $\mathbf{a}^{(i)}$  denotes the  $i$ th row of  $\mathbf{A}$ , if  $P = \{\mathbf{x} \in \mathbb{R}^k \mid \mathbf{A}'\mathbf{x} \leq \mathbf{b}', \mathbf{x} \geq 0\}$ , where  $\mathbf{A}'$  and  $\mathbf{b}'$  obtained from  $\mathbf{A}$  and  $\mathbf{b}$  by removing the  $i$ th row, respectively. We say that a constraint  $\mathbf{a}^{(i)}\mathbf{x} \leq b_i$  is strictly redundant if there does not exist  $\bar{\mathbf{x}} \in P$  with  $\mathbf{a}^{(i)}\bar{\mathbf{x}} = b_i$ .

For example, consider  $T: 2^{[2]} \rightarrow \mathbb{N}$  defined as  $T(\emptyset) = 0$ ,  $T(\{1\}) = T(\{2\}) = T(\{1, 2\}) = 1$ . Consider the polytope  $P = \{\boldsymbol{\lambda} \in \mathbb{R}^2 \mid \sum_{i \in U} \lambda_i \leq T(U), \emptyset \neq U \subseteq \{1, 2\}, \boldsymbol{\lambda} \geq 0\}$ . The inequalities  $\lambda_1 \leq T(\{1\})$ ,  $\lambda_2 \leq T(\{2\})$  are redundant, while the inequality  $\lambda_1 + \lambda_2 \leq T(\{1, 2\})$  is not redundant since e.g. the vector  $(1, 1)$  is not contained in the polytope. Here, none of the inequalities are strictly redundant since the vectors  $(1, 0)$ ,  $(0, 1)$  are contained in the polytope.

**Theorem 1.** Given  $T: 2^{[k]} \rightarrow \mathbb{N}$  for some  $k \in \mathbb{N}$ , we have

$$n(\mathcal{R}(T)) \geq \left\lceil \frac{\sum_{\emptyset \neq U \subseteq [k]} T(U)}{2^{k-1}} \right\rceil,$$

where none of the constraints  $\sum_{i \in U} \lambda_i \leq T(U)$  are strictly redundant in  $\mathbb{R}_{\geq 0}^k$ .

As we will show shortly the lower bound of Theorem 1 is indeed tight if  $k = 2$  and  $T$  is monotone and subadditive. However, this bound is not tight in general for  $K \geq 3$ . In [14] we provide an example that shows for  $K = 3$  this bound is not tight, while none of the constraints are strictly redundant.

**Corollary 2.** For some  $k \in \mathbb{N}$  and  $X \in \mathbb{N}$ , given the function  $T: 2^{[k]} \rightarrow \mathbb{N}$  defined as  $T(\emptyset) = 0$ ,  $T(U) = X$  for all subsets  $\emptyset \neq U \subseteq [k]$ , we have

$$n(\mathcal{R}(T)) \geq \left\lceil \frac{X \cdot (2^k - 1)}{2^{k-1}} \right\rceil.$$

Moreover, if  $X = t \cdot 2^{k-1}$  for some integer  $t$ , then the lower bound is tight.

Next, two more general lower bounds for  $n(\mathcal{R}(T))$ , similar to that of Theorem 1, are provided that are obtained in the search of finding a tighter lower bound for  $k \geq 3$ .

**Theorem 2.** For some integer  $k \geq 2$ , let  $T: 2^{[k]} \rightarrow \mathbb{N}$  be a function such that none of the constraints  $\sum_{i \in U} \lambda_i \leq T(U)$  are strictly redundant in  $\mathbb{R}_{\geq 0}^k$ . Then, for each  $i \in [k]$  we have

$$n(\mathcal{R}(T)) \geq \left\lceil \frac{\alpha_i + \beta_i}{2} \right\rceil \text{ where}$$

$$\alpha_i = \left\lceil \frac{\sum_{\emptyset \neq U \subseteq [k] \setminus \{i\}} T(U)}{2^{k-2}} \right\rceil, \beta_i = \left\lceil \frac{\sum_{\{i\} \subseteq U \subseteq [k]} T(U)}{2^{k-2}} \right\rceil.$$

**Theorem 3.** For some integer  $k \geq 2$ , let  $T: 2^{[k]} \rightarrow \mathbb{N}$  be a function such that none of the constraints  $\sum_{i \in U} \lambda_i \leq T(U)$  are strictly redundant in  $\mathbb{R}_{\geq 0}^k$ . Then, for each  $j \in [\ell]$  we have

$$n(\mathcal{R}(T)) \geq \left\lceil \frac{\sum_{\emptyset \neq U \subseteq [k]: \#(U \cap J) \equiv 0 \pmod{2}} T(U)}{2^{k-2}} \right\rceil,$$

where  $J \subseteq [k]$  such that  $\mathbf{v}_j = \sum_{h \in J} \mathbf{e}_h$ .

**Example 1.** For some  $x \in \mathbb{N}$ , let  $T: 2^{[3]} \rightarrow \mathbb{N}$  be defined via  $T(\{1\}) = T(\{2\}) = T(\{3\}) = T(\{1, 2\}) = T(\{1, 3\}) = x$  and  $T(\{2, 3\}) = T(\{1, 2, 3\}) = 2x$ . Based on Theorem 1, we have  $n(\mathcal{R}(T)) \geq \left\lceil \frac{9x}{4} \right\rceil$ , and according to Theorem 3, considering  $j = 3$  we have  $n(\mathcal{R}(T)) \geq \left\lceil \frac{5x}{2} \right\rceil$ . So, for  $x \geq 3$ , the lower bound obtained from Proposition 3 is tighter the one obtained from Theorem 1.

### C. Storage-Efficient Schemes for $k = 2$

Let w.l.o.g. (based on Lemma 3) the function  $T: 2^{[2]} \rightarrow \mathbb{N}$  be monotone, subadditive, and satisfy  $T(\emptyset) = 0$ . Note that for  $k = 1$  each  $T: 2^{\{1\}} \rightarrow \mathbb{N}$  is monotone and subadditive, while for  $k = 2$  the conditions can be summarized to

$$\max\{T(\{1\}), T(\{2\})\} \leq T(\{1, 2\}) \leq T(\{1\}) + T(\{2\}).$$

The following Lemma describes the generating set of  $\mathcal{R}(T)$ .

**Lemma 5.** If  $T: 2^{[2]} \rightarrow \mathbb{N}$  is monotone, subadditive, and satisfies  $T(\emptyset) = 0$ , the generating set of  $\mathcal{R}(T)$  is given by

$$S = \left\{ (T(\{1\}), T(\{1, 2\}) - T(\{1\})), (T(\{1, 2\}) - T(\{2\}), T(\{2\})) \right\}.$$

We remark that  $\#S$  in Lemma 5 is 2 or 1, where the latter happens iff  $T(\{1, 2\}) = T(\{1\}) + T(\{2\})$ .

**Lemma 6.** Let  $\{\lambda\}$  be the generating set of  $\mathcal{R}$  and  $\mathbf{n}$  be an integral solution of the ILP of Corollary 1. If  $\lambda \in \mathbb{R}_{\geq 0}^2$  and  $\mathcal{G}$  is the multiset corresponding to the  $\mathbf{n}$ , then  $\lambda \in \mathcal{S}(\mathcal{G})$ , i.e., there exists a feasible choice of  $\alpha_Y$  satisfying (3a)-(3c).

**Definition 6.** For a set  $\emptyset \neq S \subseteq \mathbb{N}$ , we denote by  $\text{Simpl}(S)$  the set of non-zero vectors in  $\{\mathbf{e}_i \mid i \in S\}$  over  $\mathbb{F}_2$ .

**Proposition 3.** For each  $\emptyset \neq S \subseteq [k]$ ,  $\#\text{Simpl}(S) = 2^s - 1$  and  $\mathcal{S}(\text{Simpl}(S)) = \mathcal{R}(T)$ , where  $s = \#S$  and  $T: 2^{[k]} \rightarrow \mathbb{N}$  is given by  $T(U) = 2^{s-1}$  for all  $U \subseteq [k]$  satisfying  $U \cap S \neq \emptyset$  and  $T(U) = 0$  otherwise (for all  $U \subseteq [k]$  with  $U \cap S = \emptyset$ ).

**Theorem 4.** For the desired service rate region  $\mathcal{R}$  given by

$$\mathcal{R} = \left\{ \lambda \in \mathbb{R}_{\geq 0}^2 : \lambda_1 \leq X, \lambda_2 \leq Y, \lambda_1 + \lambda_2 \leq \Sigma \right\},$$

where  $X, Y, \Sigma$  are non-negative integers with  $\max\{X, Y\} \leq \Sigma \leq X + Y$ , we have  $n(\mathcal{R}) = \left\lceil \frac{X+Y+\Sigma}{2} \right\rceil$ .

### REFERENCES

- [1] 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, pp. 476–489, 2011.
- [2] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.
- [3] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Transactions on Information Theory*, vol. 62, no. 8, pp. 4481–4493, 2016.
- [4] G. Joshi, E. Soljanin, and G. W. Wornell, "Efficient replication of queued tasks for latency reduction in cloud systems," in *2015 53rd Annual Allerton Conf. on Commun., Control, and Comput.*, 2015.
- [5] —, "Efficient redundancy techniques for latency reduction in cloud systems," *TOMPECS*, vol. 2, no. 2, pp. 12:1–12:30, 2017.
- [6] M. F. Aktas, S. Kadhe, E. Soljanin, and A. Sprintson, "Download time analysis for distributed storage codes with locality and availability," *arXiv:1912.09765*, Dec 2019.
- [7] M. Aktas, G. Joshi, S. Kadhe, F. Kazemi, and E. Soljanin, "Service rate region: A new aspect of coded distributed system design," *arXiv:2009.01598*, Sep 2020.
- [8] M. Aktas, 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 Commun., Control, and Comput.*, 2017.
- [9] S. E. Anderson, A. Johnston, G. Joshi, G. L. Matthews, C. Mayer, and E. Soljanin, "Service rate region of content access from erasure coded storage," in *2018 IEEE Information Theory Workshop (ITW)*, 2018.
- [10] 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)*, June 2020.
- [11] 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)*, June 2020.
- [12] P. Peng and E. Soljanin, "On distributed storage allocations of large files for maximum service rate," in *2018 56th Annual Allerton Conf. on Commun., Control, and Comput.* IEEE, 2018.
- [13] M. Noori, E. Soljanin, and M. Ardakani, "On storage allocation for maximum service rate in distributed storage systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016.
- [14] F. Kazemi, S. Kurz, E. Soljanin, and A. Sprintson, "Efficient storage schemes for desired service rate regions," *arXiv*, 2020.
- [15] S. Dodunekov and J. Simonis, "Codes and projective multisets," *The Electronic Journal of Combinatorics*, vol. 5, no. 1, p. 37, 1998.
- [16] A. Beutelspacher, B. Albrecht, and U. Rosenbaum, *Projective geometry: from foundations to applications*. Cambridge Uni. Press, 1998.