Linear locally recoverable codes with locality $r=1$

Teo, Samuel Tien Ho

2017

Teo, S. T. H. (2017). Linear locally recoverable codes with locality $r=1$. Master's thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/69622>

<https://doi.org/10.32657/10356/69622>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**LINEAR LOCALLY RECOVERABLE CODES
WITH LOCALITY $r = 1$**

SAMUEL TEO TIEN HO

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

2017

NANYANG
TECHNOLOGICAL
UNIVERSITY

**LINEAR LOCALLY RECOVERABLE CODES
WITH LOCALITY $r = 1$**

SAMUEL TEO TIEN HO

School of Physical and Mathematical Sciences

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Master of Science

2017

Acknowledgments

My post-graduate experience has been fruitful and exciting, thanks to the people that have accompanied me through this journey. They have played a significant role in molding me into who I am today.

Firstly, I would like to express my heartfelt gratitude to my supervisor Prof. Chee Yeow Meng for his guidance and support throughout my graduate studies. I was fortunate to have the opportunity to work with him in the field of coding theory. His vast knowledge of this field gave me great insights in my own research. I am also grateful for his sound advice and encouragement whenever I was facing difficulties.

I am greatly appreciative of my friend and mentor, Kiah Han Mao. He had on many occasions shared his academic experience with me and always offered precious advice in my academic path. It was a pleasure discussing and collaborating with him and I am grateful for his help on my research.

I am deeply grateful to my friend and colleague, Punarbasu. He has always been generous in sharing his knowledge and offering a helping hand in obstacles that I faced. I learnt many invaluable lessons from him both academically and in life.

I would also like to thank my friends, peers and professors from NTU. I have enjoyed the companionship of friends which made learning more interesting, peers who contributed to a positive work environment and professors who are always friendly and approachable in sharing their wisdom.

Last but not least, I am truly blessed to have the love, support and encouragement from my beloved family and my adorable girlfriend, Khai Ting. They have been my pillar of strength in this journey and continuously motivates me to strive for the better.

Contents

1	Introduction	11
1.1	Background	11
1.2	Structure and Contributions of Thesis	12
2	Fundamentals of Coding Theory	13
2.1	Classical Coding Theory	13
2.2	Bounds for Optimal Size of Code	16
2.3	Constructions of Linear Codes over $\mathbb{F}_q$	17
3	Fundamentals of Locally Recoverable Codes	19
3.1	Locally Recoverable Codes (LRCs)	19
3.2	Bounds on Locally Recoverable Codes	22
4	Linear Locally Recoverable Codes with Locality $r = 1$	24
4.1	Properties of Linear $\text{LRC}_q^{(r=1)}(n, k, d)$	24
4.2	Bounds on Linear $\text{LRC}_q^{(r=1)}(n, k, d)$	26

4.3	Optimal Linear LRC $_q^{(r=1)}(n, k, d)$	29
5	Code Construction using Partitions for Linear LRC$_q^{(r=1,t)}(n, d)$	34
5.1	Preliminaries	34
5.2	Code Construction using Partitions for Linear LRC $_q^{(r=1)}(n, d)$	35
5.3	Tightness of Bound on Binary Linear LRC $^{(r=1)}(n, d)$	36
5.4	Properties and Bounds of Linear LRC $_q^{(r=1,t)}(n, k, d)$	49
6	Conclusion	51
	Bibliography	52

List of Symbols

$[n, k, d]$	Common expression used to denote the length, dimension and minimum distance of a linear code.
$[n]$	The set: $\{1, 2, \dots, n\}$.
$ \mathcal{C} $	The size of a code $\mathcal{C}$, i.e. the number of codewords in $\mathcal{C}$.
$\mathbf{x}$	A vector $\mathbf{x}$ that is written as a tuple of its n coordinates: $\mathbf{x} = (x_1, x_2, \dots, x_n)$.
$\mathbf{x} _R$	Vector $\mathbf{x}$ restricted to a set of coordinates R . If $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ and $R = \{1, 4, 7\}$, then $\mathbf{x} _R = (x_1, x_4, x_7)$.
$\mathbb{F}_2$	A binary finite field.
$\mathbb{F}_q$	A finite field with q elements.
$\lceil x \rceil$	Ceiling of x , the smallest integer greater than or equal to x .
$\lfloor x \rfloor$	Floor of x , the greatest integer less than or equal to x .
$\mathbb{N}$	A set of natural numbers: $\{1, 2, \dots\}$.
$d(\mathbf{x}, \mathbf{y})$	The Hamming distance between vectors $\mathbf{x}$ and $\mathbf{y}$.
$d_{\min}(\mathcal{C})$	The minimum distance of a code $\mathcal{C}$.

Abstract

A code is called a locally recoverable code (LRC) with locality r if any symbol of a codeword can be recovered by accessing r other symbols that forms the recovering set. A LRC has availability t if each symbol has at least t disjoint recovering sets. In this thesis, we summarise the known properties and bounds of linear LRCs and will focus primarily on linear LRCs with locality $r = 1$ and availability $t = 1$. We will derive a few propagation rules for linear LRCs with locality $r = 1$ and present a code construction method using partitions of length n of a LRC. We will prove the optimality of linear LRCs with locality $r = 1$ for certain values of length n and distance d , and compare upper bounds and lower bounds of binary linear LRCs with locality $r = 1$ with respect to dimension k . The investigation into the optimal dimensions of linear LRCs is important to improve efficiency in their applications in distributed and cloud storage systems.

Chapter 1

Introduction

1.1 Background

A common problem with large-scale distributed and cloud storage systems is the recovery from failure of its nodes. Traditionally, distributed storage systems rely on replication of data packets across different nodes to ensure reliability against node failure. However, the main disadvantage of this method is that it creates a large storage overhead and is therefore inefficient. Alternative solutions involve the use of erasure codes, which achieves high data reliability with considerably smaller storage overhead [15].

Today's storage systems are less susceptible to multiple concurrent node failures and the more common case is the failure of a single node. Therefore, it is important to design storage systems that allow the node to quickly reconstruct its lost packets from others. One particular metric for the repair efficiency of a single node failure is the repair locality, i.e. the number of nodes required to reconstruct the lost node.

A (n, k, d) locally recoverable code (LRC) with length n , dimension k and minimum distance d , is said to have locality r , if any single erasure can be recovered by accessing r other symbols of the codeword which forms a recovering set. The concept of locality was introduced by Gopalan et al. in [4] where the paper performed an in-depth study of relationships between redundancy, locality and minimum distance of LRCs. This leads to a large number of publications on locally recoverable codes [1], [3], [4], [9].

Realising that local recovery of the symbols would fail if one of the r symbols were also erased, Wang and Zhang introduced in [14] the notion of availability of a LRC. A LRC with locality r is said to have availability t if each symbol of the codeword has at least t disjoint

recovering sets. The properties of LRC with locality r and availability t were further studied in [2], [8] and [11].

The main study of LRC is to ideally find optimal LRCs that have large values of d and a small values of r . Hence, in this thesis, we shall look through known properties and bounds of linear LRCs and will focus primarily on linear LRCs with locality $r = 1$ and availability $t = 1$. The study of the optimality of linear LRC with locality $r = 1$ has been mentioned by Tamo et al. in [10] and [13]. It states that by replicating each symbol once in an $[\frac{n}{2}, k, \frac{d}{2}]_q$ MDS code, we can obtain an optimal linear LRC with locality $r = 1$. Nonetheless, this code construction method only accounts for parameters of linear LRC with locality $r = 1$ where the values of length n and minimum distance d are even. Hence, this gives us the opportunity to look for optimal linear LRCs beyond the values where n and d are even.

1.2 Structure and Contributions of Thesis

Chapter 2 gives a brief overview of fundamentals of coding theory. It shows all the properties and bounds found in coding theory that are relevant to this thesis. Chapter 3 introduces the concept of locality and availability of a LRC. It also contains all the known upper bounds and properties of LRCs.

Chapter 4 narrows down the focus of our thesis to linear LRCs with locality $r = 1$. We will discuss the properties and bounds specific to linear LRCs with locality $r = 1$ as well as introduce some propagation techniques for linear LRCs with locality $r = 1$ that were derived from propagation rules of linear codes. We will also prove the optimality of the dimension k of linear LRCs with locality $r = 1$ for certain values of length n and minimum distance d .

Our main contribution of this thesis comes in Chapter 5, where we present a code construction method for linear LRCs with locality $r = 1$ using the concept of partitions for the length n . We show that this construction is a good lower bound for the optimal dimension of binary linear LRCs with locality $r = 1$ in Section 5.3 where we compare the upper bounds of linear LRCs with locality $r = 1$ with the values that we derive from the lower bound of our code construction method. We also extend some ideas of LRCs with locality $r = 1$ discussed in Chapter 4 and the code construction method to LRCs with locality $r = 1$ and availability t in Section 5.4.

We suggest the readers unfamiliar with coding theory to read in the following sequence: Section 2.1, Section 2.2, Chapter 1, and then sequentially from Section 2.3. For readers who are familiar with locally recoverable codes, we suggest to skip to Chapter 4. For the remaining readers, you are encouraged to read from Chapter 3.

Chapter 2

Fundamentals of Coding Theory

In this chapter, we will introduce readers to the fundamental concepts of coding theory. We will begin with Section 2.1, which is a brief introduction to classical coding theory by using a few examples to explain the technical properties of a code. In Section 2.2, we introduce some classical bounds for codes that are useful to us in the later chapters. We end off the chapter with Section 2.3, where we present some code construction techniques that are well known for linear codes.

2.1 Classical Coding Theory

Let us consider the classical example where we have a communication model, where Alice wants to send a message through a channel to Bob and let us represent the message using a length n q -ary vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $\mathbf{x} \in \mathbb{F}_q^n$. In the ideal situation, the vector $\mathbf{x}$ will be sent through the channel without errors and Bob will receive the intended message. But often in practice, channels are noisy and it is very likely that some of the symbols will be changed or lost during transmission. The difference between the intended message and the received message that occur due to noise are called errors.

Coding theory allows us to solve these problems and preserve the integrity of the messages through the use of error correcting codes. Error correcting codes (or simply codes) introduce redundancy into the data transmitted to account for errors. They do this in the form of error detection and error correction, where we are able to tell when an error has occurred in the communication and to correct the received message to the intended message.

A q -ary *code*, denoted by $\mathcal{C}$, is a set of vectors over a finite field $\mathbb{F}_q$ with cardinality q . The

elements of $\mathcal{C}$ are called *codewords*. If $\mathcal{C} \subset \mathbb{F}_q^n$ is also a subspace of $\mathbb{F}_q^n$ then $\mathcal{C}$ is called a *linear code*. A *generator matrix* for a linear code $\mathcal{C}$ is a $k \times n$ matrix whose rows are linearly independent codewords of the code. Hence, we can also say that if each codeword of $\mathcal{C}$ belongs to the row space of the generator matrix of $\mathcal{C}$, then $\mathcal{C}$ is a linear code.

Example 2.1.1. Suppose that Alice wants to send the following message to Bob: $\mathbf{x} = (1, 0, 1, 1)$. By going through the following process called *encoding*, we can introduce some redundancy into the message to account for errors.

Alice's message could be encoded to a codeword of a *Hamming code* in the following way:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ \mathbf{x} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ \mathbf{y} \end{bmatrix},$$

where G is the generator matrix of a Hamming code. By multiplying the message $\mathbf{x}$ with G , we create a codeword $\mathbf{y}$ of the Hamming code.

The Hamming code mentioned in Example 2.1.1 is a linear code. Linear codes are commonly described using the following three parameters: $[n, k, d]$. We will now define the parameters and some common properties of a code (see [5],[6],[7] for more details):

The *length* of each codeword of a code is represented by n .

The *Hamming distance* between two vectors $\mathbf{x} = x_1 \dots x_n$, and $\mathbf{y} = y_1 \dots y_n$ is the number of places where they differ component-wise, and is denoted by $d(\mathbf{x}, \mathbf{y})$. The *Hamming weight* of a vector $\mathbf{x} = x_1 \dots x_n$, is the number of nonzero x_i and is denoted by $wt(\mathbf{x})$. Further on in this thesis, all distance and weight referred to between codewords will be known as Hamming distance and Hamming weight.

Another parameter of a code $\mathcal{C}$ is the *minimum distance* of a code, denoted by d , where

$$d = \min d(\mathbf{x}, \mathbf{y}), \text{ for all } \mathbf{x}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}.$$

This is the minimum of the distances of all codewords in $\mathcal{C}$. If a code has minimum distance d , then any two codewords in $\mathcal{C}$ differ in at least d places. For a linear code, minimum distance d is also the minimum weight of any nonzero codeword.

The *size* of a code $\mathcal{C}$ is the number of codewords in $\mathcal{C}$. For q -ary linear codes, the size is equals to q^k where k is the rank of the generator matrix of the code which is also referred as the *dimension* of the code. The largest size of a q -ary code of length n and minimum distance d is denoted by $A_q(n, d)$ and the largest size of a linear code of length n and minimum distance d is denoted by $B_q(n, d)$. When the codes are binary, i.e. $q = 2$, it is denoted by $A(n, d)$ and $B(n, d)$ respectively.

A q -ary linear code with length n , dimension k and minimum distance d is usually called a $[n, k, d]_q$ -linear code. If the code is binary, i.e. $q = 2$, then we omit the q and call it a $[n, k, d]$ -linear code.

Example 2.1.2. Let us find the parameters of the binary linear code $\mathcal{C}$ with its generator matrix G given below,

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

The length of each codeword is 5 and the rank of the generator matrix is 2. We can find the codewords of $\mathcal{C}$ by taking the linear combination of the rows of the generator matrix G . Hence, the code $\mathcal{C} = \{00000, 11100, 00111, 11011\}$ (The codewords of a code are commonly written as a sequence of digits e.g. 101 instead of its vector form e.g. $(1, 0, 1)$). We can find the minimum distance of $\mathcal{C}$ by finding the minimum of the Hamming distance of all distinct pairs of codewords in $\mathcal{C}$,

$$d(00000, 11100) = 3,$$

$$d(00000, 00111) = 3,$$

$$d(00000, 11011) = 4,$$

$$d(11100, 00111) = 4,$$

$$d(11100, 11011) = 3,$$

$$d(00111, 11011) = 3,$$

or we can find the minimum weight of any nonzero codeword in $\mathcal{C}$,

$$wt(11100) = 3,$$

$$wt(00111) = 3,$$

$$wt(11011) = 4.$$

From the above result, we find that the minimum distance of $\mathcal{C}$ is 3. Hence, $\mathcal{C}$ is a $[5, 2, 3]$ -linear code.

The following is an important property for linear codes which is required in the later chapters.

Definition 2.1.3. Two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are *equivalent* if one code can be obtained from the other by an arbitrary permutation of components. Equivalent linear codes have the same dimension and minimum distance.

2.2 Bounds for Optimal Size of Code

In classical coding theory, we often set out to find the optimal size of a code to improve the cost and efficiency in the transmission of information for its applications. The various classical bounds for $A_q(n, d)$ mentioned in [7] are stated below. They show the relationship between the length, size and minimum distance of a code.

Theorem 2.2.1. (*Singleton Bound*). *Let $1 \leq d \leq n$. Then*

$$A_q(n, d) \leq q^{n-d+1}. \quad (2.1)$$

Theorem 2.2.2. (*Singleton Bound for linear codes*). *Let $1 \leq d \leq n$ and the maximum dimension of a (n, d) -code be k . Then*

$$A_q(n, d) = q^k \leq q^{n-d+1} \quad \Rightarrow \quad k \leq n - d + 1. \quad (2.2)$$

A linear $[n, k, d]_q$ -code which meets the Singleton bound i.e. $k = n - d + 1$, is called a *Maximum Distance Separable (MDS)* code.

Example 2.2.3. (*Trivial MDS codes*)

- Linear codes like the q -ary repetition code, contains only q codewords and have distance $d = n$, and hence satisfies the Singleton bound with $k = 1$.
- Linear codes that uses the whole of $\mathbb{F}_n^q$ have minimum distance $d = 1$ satisfies the Singleton bound with $k = n$.
- Linear codes with a single parity symbol that have minimum distance $d = 2$ can achieve the Singleton bound with $k = n - 1$.

Theorem 2.2.4. (*Gilbert-Vashamov (GV) and Hamming Bound*). *Let $1 \leq d \leq n$. Then*

$$\frac{q^n}{V_q(n, d-1)} \leq A_q(n, d) \leq \frac{q^n}{V_q(n, \lfloor (d-1)/2 \rfloor)}, \quad (2.3)$$

where

$$V_q(n, d) = \sum_{t=0}^d \binom{n}{t} (q-1)^t. \quad (2.4)$$

Theorem 2.2.5. (*Plotkin Bound*). *Suppose $qd - (q-1)n > 0$. Then*

$$A_q(n, d) \leq \frac{qd}{qd - (q-1)n}. \quad (2.5)$$

Theorem 2.2.6. (*Griesmer Bound*). *For a $[n, k, d]_q$ -linear code, the Griesmer bound states that:*

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil. \quad (2.6)$$

2.3 Constructions of Linear Codes over $\mathbb{F}_q$

In this section, we state a few propagation rules taken from [6] and [7] that allow for construction of new linear codes from old linear codes. These propagation techniques are helpful for finding the lower bounds for linear codes with larger parameters.

Theorem 2.3.1. *Suppose there is a $[n, k, d]_q$ -linear code. Then*

- (i) *(lengthening) there exists a $[n + r, k, d]_q$ -linear code for any $r \geq 1$;*
- (ii) *(subcodes) there exists a $[n, k - r, d]_q$ -linear code for any $1 \leq r \leq k - 1$;*
- (iii) *(puncturing) there exists a $[n - r, k, d - r]_q$ -linear code for any $1 \leq r \leq d - 1$;*
- (iv) *there exists a $[n, k, d - r]_q$ -linear code for any $1 \leq r \leq d - 1$;*
- (v) *there exists a $[n - r, k - r, d]_q$ -linear code for any $1 \leq r \leq k - 1$.*

Theorem 2.3.2. *(Augmenting) Let $\mathcal{C}$ be a binary $[n, k, d]$ -linear code which does not contain the all-ones vector $\mathbf{1}$. Then the augmented code $\mathcal{C}^{(a)}$ defined by*

$$\mathcal{C}^{(a)} = \mathcal{C} \cup \{\mathbf{1} + \mathcal{C}\}$$

is a $[n, k + 1, \min\{d, n - d'\}]$ -linear code, where d' is the largest weight of any codeword of $\mathcal{C}$.

Remark 2.3.3. Let G be a generator matrix of $\mathcal{C}$. It is then easy to see that the matrix

$$\begin{bmatrix} G \\ 1 \cdots 1 \end{bmatrix}$$

is a generator matrix of $\mathcal{C}^{(a)}$.

Theorem 2.3.4. *(Direct Sum) Let $\mathcal{C}_i$ be a $[n_i, k_i, d_i]_q$ -linear code for $i = 1, 2$. Then the direct sum of $\mathcal{C}_1$ and $\mathcal{C}_2$ defined by*

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(\mathbf{u}, \mathbf{v}) : \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$$

is a $[n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\}]_q$ -linear code.

Remark 2.3.5. Let G_i be a generator matrix of $\mathcal{C}_i$, for $i = 1, 2$. It is then easy to see that the matrix

$$\begin{bmatrix} G_1 & O \\ O & G_2 \end{bmatrix}$$

is a generator matrix of $\mathcal{C}_1 \oplus \mathcal{C}_2$, where O stands for the zero matrix (note that the two zero matrices might have different sizes).

Theorem 2.3.6. *(($\mathbf{u}, \mathbf{u} + \mathbf{v}$)-Construction) Let $\mathcal{C}_i$ be a $[n, k_i, d_i]_q$ -linear code for $i = 1, 2$. Then the code $\mathcal{C}$ defined by*

$$\mathcal{C} = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) : \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$$

is a $[2n, k_1 + k_2, \min\{2d_1, d_2\}]_q$ -linear code.

Remark 2.3.7. Let G_i be a generator matrix of $\mathcal{C}_i$, for $i = 1, 2$. It is then easy to see that the matrix

$$\begin{bmatrix} G_1 & G_1 \\ O & G_2 \end{bmatrix}$$

is a generator matrix of $\mathcal{C}$ from the $((\mathbf{u}, \mathbf{u}+\mathbf{v})$ -Construction), where O stands for the zero matrix.

Corollary 2.3.8. *Let C be a binary $[n, k, d]$ -linear code. Then the code $\mathcal{C}$ defined by*

$$\mathcal{C} = \{(\mathbf{u}, \mathbf{u}) : \mathbf{u} \in C\} \cup \{(\mathbf{u}, \mathbf{1} + \mathbf{u}) : \mathbf{u} \in C\}$$

is a binary $[2n, k + 1, \min\{n, 2d\}]$ -linear code.

Remark 2.3.9. Let G be a generator matrix of C . It is then easy to see that the matrix

$$\begin{bmatrix} G & G \\ 0 \cdots 0 & 1 \cdots 1 \end{bmatrix}$$

is a generator matrix of $\mathcal{C}$.

Chapter 3

Fundamentals of Locally Recoverable Codes

In this chapter, we will formally define locally recoverable codes and introduce the concept of locality and availability of a locally recoverable code. We will also state the upper bounds on the minimum distance or the maximum size of a locally recoverable codes that were proven in [1], [2], [4], [9], [11], [12] and [14].

3.1 Locally Recoverable Codes (LRCs)

The concept of locality of a code can be fairly abstract, hence we introduce it through the help of some examples of locally recoverable codes.

Given the generator matrix G of the binary *Simplex code*, consider the following encoding of the message $\mathbf{x} = (x_1, x_2, x_3)$ into the codeword $\mathbf{y}$, where $x_1, x_2, x_3 \in \{0, 1\}$:

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ \mathbf{x} \end{bmatrix} \begin{matrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \\ G \end{matrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_1 + x_2 & x_1 + x_3 & x_2 + x_3 & x_1 + x_2 + x_3 \\ \mathbf{y} \end{bmatrix}$$

Since x_1, x_2, x_3 are arbitrary, $\mathbf{y}$ can represent any codeword in the binary Simplex code. Let us assume that x_1 , the first bit of $\mathbf{y}$ is lost during transmission. Of the remaining bits of $\mathbf{y}$, we can use a combination of two bits to recover the lost symbol x_1 . By denoting the t -th set of coordinates whose symbols recover x_i as R_i^t for $1 \leq i \leq 7$, we have the following:

$$R_1^1 = \{2, 4\}, \quad R_1^2 = \{3, 5\}, \quad R_1^3 = \{6, 7\}.$$

We observe that over the binary finite field $\mathbb{F}_2$, we can have the following equations:

$$x_1 = x_2 + (x_1 + x_2), \quad x_1 = x_3 + (x_1 + x_3), \quad x_1 = (x_2 + x_3) + (x_1 + x_2 + x_3).$$

R_1^1 , R_1^2 and R_1^3 are referred to as the *recovering sets* for the symbol x_1 .

The maximum cardinality that any recovering set can have is defined as the *locality* of the symbol; in other words, the maximum number of symbols required to recover a lost symbol is the locality of that lost symbol. Thus, x_1 has locality equal to 2.

The total number of disjoint recovering sets for a symbol is defined as the *availability* of the symbol. In our example, the availability of x_1 is equal to 3 since there are three disjoint recovering sets that recovers the first symbol.

We shall use r and t to denote locality and availability of a code respectively. In the above example of the binary Simplex code, we can find $t = 3$ disjoint recovering sets with cardinality $r = 2$ for each of the remaining symbols of $\mathbf{y}$ as well. The recovering sets for the symbols of the coordinates 2-7 are given explicitly below:

$$\begin{aligned} R_2^1 &= \{1, 4\}, & R_2^2 &= \{3, 6\}, & R_2^3 &= \{5, 7\}; \\ R_3^1 &= \{1, 5\}, & R_3^2 &= \{2, 6\}, & R_3^3 &= \{4, 7\}; \\ R_4^1 &= \{1, 2\}, & R_4^2 &= \{3, 7\}, & R_4^3 &= \{5, 6\}; \\ R_5^1 &= \{1, 3\}, & R_5^2 &= \{2, 7\}, & R_5^3 &= \{4, 6\}; \\ R_6^1 &= \{2, 3\}, & R_6^2 &= \{1, 7\}, & R_6^3 &= \{4, 5\}; \\ R_7^1 &= \{1, 6\}, & R_7^2 &= \{2, 5\}, & R_7^3 &= \{3, 4\}. \end{aligned}$$

Hence, we can say that the Simplex $[7, 3, 4]$ -code has locality $r = 2$ with availability $t = 3$.

Example 3.1.1. We present a $[9, 2, 6]$ -linear code $\mathcal{C}$ with its codewords arranged vertically.

$$\mathcal{C} = \left\{ \begin{array}{l} 000000000, \\ 111111000, \\ 111000111, \\ 000111111 \end{array} \right\}$$

We observe that every symbol of each coordinate can be recovered from at least 1 other symbol.

Considering the locality of $\mathcal{C}$ to be $r = 1$, we have for the first coordinate,

$$R_1^1 = \{2\}, \quad R_1^2 = \{3\}.$$

We also observe that the symbol of every coordinate can be recovered from $t = 2$ other symbols.

But if we now consider the locality of $\mathcal{C}$ to be $r = 2$, then the symbol of every coordinate can be recovered from 3 other disjoint recovering sets. For the first coordinate, we can have

$$R_1^1 = \{2\}, \quad R_1^2 = \{3\}, \quad R_1^3 = \{4, 7\}, \quad R_1^4 = \{5, 8\}, \quad R_1^5 = \{6, 9\}.$$

Hence, $\mathcal{C}$ is a $[9, 2, 6]$ locally recoverable code with locality $r = 1$ and availability $t = 2$, or a $[9, 2, 6]$ locally recoverable code with locality $r = 2$ and availability $t = 5$.

After seeing a few examples of locally recoverable codes, we will now present the definitions for locally recoverable codes and its availability. These can be found from [12].

Definition 3.1.2. (Locally recoverable codes (LRCs)) A code $\mathcal{C} \subset \mathbb{F}_q^n$ has locality r if every symbol of the codeword $\mathbf{x} \in \mathcal{C}$ can be recovered from a subset of r other symbols of $\mathbf{x}$. In other words, given $\mathbf{x} \in \mathcal{C}$, $i \in [n]$, there exists a subset of coordinates $R_i \subset [n] \setminus i$, where $|R_i| \leq r$ such that the value of $x_i = \sum_{j=1}^r x_j \pmod q$ where $x_j \in \mathbf{x}|_{R_i}$. The subset R_i is called a *recovering set* for the symbol x_i .

To formally state the following definition, we denote by $\mathcal{C}_I$ as the restriction of the code $\mathcal{C}$ to a subset of coordinates $I \subset [n]$. Given $a \in \mathbb{F}_q$ define the set of codewords $\mathcal{C}(i, a) = \{\mathbf{x} \in \mathcal{C} : x_i = a\}$, $i \in [n]$.

Definition 3.1.3. (Availability of LRCs) A code $\mathcal{C} \subset \mathbb{F}_q^n$ with locality r is said to have availability t if for every $i \in [n]$ there are t pairwise disjoint recovering sets $R_i^1, \dots, R_i^t \subset [n] \setminus i$ such that for all $j = 1, \dots, t$,

$$\mathcal{C}(i, a)_{R_i^j} \cap \mathcal{C}(i, a')_{R_i^j} = \emptyset, \quad a \neq a'. \quad (3.1)$$

In the context of this thesis, we will denote a q -ary (n, k, d) locally recoverable code with locality r and availability t as $\text{LRC}_q^{(r,t)}(n, k, d)$ and q -ary (n, d) locally recoverable code with locality r and availability t as $\text{LRC}_q^{(r,t)}(n, d)$. In the context where the availability of the LRC is $t = 1$, we will use $\text{LRC}_q^{(r)}(n, k, d)$ and $\text{LRC}_q^{(r)}(n, d)$ instead. We will also omit q if we are describing binary LRCs using $\text{LRC}^{(r)}(n, k, d)$ and $\text{LRC}^{(r)}(n, d)$.

3.2 Bounds on Locally Recoverable Codes

We understand from the definition of LRCs that the concept of locality and availability are additional properties that a code has to satisfy, hence any upper bound on the size of a length n code with a given minimum distance d which were previously known also applies to LRCs.

Remark 3.2.1. Let the maximum size of a $\text{LRC}_q^{(r,t)}(n, d)$ be $A_q^{(r,t)}(n, d)$. Then for any $\text{LRC}_q^{(r,t)}(n, d)$,

$$A_q^{(r,t)}(n, d) \leq A_q(n, d). \quad (3.2)$$

But the above bound does not give us a tight upper bound for LRCs. Hence we need to investigate for better and tighter upper bounds on LRCs in hopes of finding optimal sizes and minimum distances for LRCs. We will present some known bounds for LRCs below.

3.2.1 Bounds on $\text{LRC}_q^{(r)}(n, k, d)$

The following upper bounds on the minimum distance and rate of LRC codes that were proved by Gopalan et al. in [4], and Papailiopoulos, Dimakis in [9] is one of the first main results for bounds on the parameters of LRCs:

Theorem 3.2.2. *Let $\mathcal{C}$ be a linear $\text{LRC}_q^{(r)}(n, k, d)$. The rate of $\mathcal{C}$ satisfies*

$$\frac{k}{n} \leq \frac{r}{r+1}, \quad (3.3)$$

and the minimum distance of $\mathcal{C}$ satisfies

$$d \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2. \quad (3.4)$$

The bound (3.4) forms a generalization of the classical Singleton bound (Theorem 2.2.2), and reduces to it for the maximum value of locality $r = k$.

An upper bound on the dimension k that depends on q was derived by Cadambe and Mazumdar in [1]. Before we introduce it, let $k_q(n, d) := \log_q A_q(n, d)$ where we recall that $A_q(n, d)$ is the maximum cardinality of a length n q -ary code with minimum distance d .

Theorem 3.2.3. *For any $\text{LRC}_q^{(r)}(n, k, d)$,*

$$k \leq \min_{1 \leq j \leq \min(\lceil \frac{n}{r+1} \rceil, \lceil \frac{k}{r} \rceil)} \{jr + k_q(n - j(r+1), d)\}. \quad (3.5)$$

Note that this bound (3.5) also applies to general (non-linear) codes.

3.2.2 Bounds on $\text{LRC}_q^{(r,t)}(n, k, d)$

The bounds on LRCs in Subsection 3.2.1 are concerned only with the locality of the code but not the availability of recovering sets for each symbol. Having introduced the notion of availability in [14], Wang and Zhang derive an upper bound on the minimum distance of LRCs with locality and availability.

Theorem 3.2.4. *For any linear $\text{LRC}_q^{(r,t)}(n, k, d)$,*

$$d \leq n - k - \left\lceil \frac{t(k-1)+1}{t(r-1)+1} \right\rceil + 2. \quad (3.6)$$

The same bound (3.6) is derived by Dimakis et al. in [2]; they show that the bound holds for non-linear codes as well. We observe that this bound is an extension from one to many recovering sets since when $t = 1$, it reduces to (3.4).

Tamo and Barg extended the bound further by deriving an upper bound on the minimum distance of codes that have all symbol locality and availability in [11]. They obtain the following results:

Theorem 3.2.5. *Let $\mathcal{C}$ be a $\text{LRC}_q^{(r,t)}(n, k, d)$ with t disjoint recovering sets, each of size r . Then the rate of $\mathcal{C}$ satisfies*

$$\frac{k}{n} \leq \frac{1}{\prod_{j=1}^t (1 + \frac{1}{jr})}, \quad (3.7)$$

and the minimum distance of $\mathcal{C}$ is bounded above as follows:

$$d \leq n - \sum_{i=0}^t \left\lfloor \frac{k-1}{r^i} \right\rfloor. \quad (3.8)$$

Similar to (3.6), (3.8) is also an extension from one to many recovering sets since when $t = 1$, it also reduces to (3.4).

We shall use the upper bounds mentioned in this chapter to help us prove for the optimality of code sizes for certain LRCs in the following chapters.

Chapter 4

Linear Locally Recoverable Codes with Locality $r = 1$

The only known study of optimality of linear $\text{LRC}_q^{(r=1)}(n, k, d)$ states that by replicating each symbol once in an $[\frac{n}{2}, k, \frac{d}{2}]_q$ MDS code, we can obtain an optimal linear $\text{LRC}_q^{(r=1)}(n, k, d)$. Nonetheless, this code construction method only accounts for parameters of linear LRC with locality $r = 1$ where the values of length n and minimum distance d are even. Hence, this motivates us to look for optimal linear $\text{LRC}_q^{(r=1)}(n, k, d)$ beyond the values where n and d are even.

We will begin this chapter by stating some important properties of linear $\text{LRC}_q^{(r=1)}(n, k, d)$ that we will refer heavily to throughout the rest of this thesis. In Section 4.2, we will present the upper bounds on linear $\text{LRC}_q^{(r=1)}(n, k, d)$ by deriving them from the upper bounds found in Section 3.2 and also some lower bound constructions that were similar to those found in Section 2.3 for linear codes. In the last Section 4.3, we will then prove the optimality of linear $\text{LRC}_q^{(r=1)}(n, k, d)$ for some given length n and minimum distance d .

4.1 Properties of Linear $\text{LRC}_q^{(r=1)}(n, k, d)$

We start this section by proving the following important lemma for any linear $\text{LRC}_q^{(r=1)}(n, k, d)$.

Lemma 4.1.1. *Let $\mathcal{C}$ be a $[n, k, d]_q$ -linear code. $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ if and only if the symbols for all the coordinates of any codeword $\mathbf{x} \in \mathcal{C}$ are replicated at least once, i.e. the element of the recovering set of a symbol at a coordinate must be the same symbol from another coordinate for all codewords.*

Proof. We shall prove the above lemma using the definition of LRC. If $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$, we have the cardinality of the recovering set of x_i to be $|R_i| = 1$, and $R_i = j$ where $j \in [n] \setminus i$, hence $x_i = \mathbf{x}|_{R_i} = x_j \in \mathbb{F}_q$. Suppose that for any $i \in [n]$, $x_j \neq x_i$ for some $j \in [n] \setminus i$. Then the zero vector $\mathbf{0}$, cannot be a codeword in $\mathcal{C}$. This is a contradiction to the assumption that $\mathcal{C}$ is a linear code. If the symbols for all the coordinates of any codeword $\mathbf{x} \in \mathcal{C}$ are replicated at least once, then the symbols for all coordinates can be recovered from at least 1 other symbol for the other coordinates. Hence, $|R_i| \leq 1$ for all $i \in [n]$. By Definition 3.1.2, $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$. $\square$

Corollary 4.1.2. *Linear $\text{LRC}_q^{(r=1)}(1, 1)$ cannot exist.*

Proof. By Lemma 4.1.1, replication of a symbol at least once is required for linear $\text{LRC}_q^{(r=1)}(n, d)$, therefore we must have $n \geq 2$. $\square$

Lemma 4.1.1 does not have to hold for non-linear codes. Look at the following example.

Example 4.1.3. Consider the binary non-linear code with length $n = 2$, minimum distance $d = 2$ and cardinality 2,

$$\mathcal{C} = \left\{ \begin{array}{l} \mathbf{x} = 01, \\ \mathbf{y} = 10 \end{array} \right\}.$$

Since $\mathbf{x} = \bar{\mathbf{y}}$ and $\mathbf{y} = \bar{\mathbf{x}}$, by Definition 3.1.2, $\mathcal{C}$ is a $\text{LRC}^{(r=1)}(2, 1, 2)$. But $\mathcal{C}$ does not satisfy Lemma 4.1.1, since neither of the 2 only codewords $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ have symbols that replicate.

To allow us to study linear $\text{LRC}_q^{(r=1)}(n, k, d)$ more efficiently, we will need the following corollary.

Corollary 4.1.4. *Let G be a $k \times n$ generator matrix of a q -ary $[n, k, d]$ -linear code $\mathcal{C}$. $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ if and only if the columns of G are replicated at least once.*

Proof. Since the rows of G are codewords of $\mathcal{C}$, if $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$, then by Lemma 4.1.1, all codewords of $\mathcal{C}$ must have its symbols of all coordinates replicated at least once, hence the columns of G also have to be replicated at least once. If the columns of G are replicated at least once, then the row space of G forms $\mathcal{C}$ which satisfies Lemma 4.1.1, hence $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$. $\square$

Lemma 4.1.5. *Any linear $\text{LRC}_q^{(r=1)}(n, k, d)$ is also a linear $\text{LRC}_q^{(r=1)}(n, k, d - s)$ for $1 \leq s \leq d - 1$.*

Proof. By mathematical induction, it suffices to show that a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ is also a linear $\text{LRC}_q^{(r=1)}(n, k, d - 1)$. A linear $\text{LRC}_q^{(r=1)}(n, k, d)$ has minimum distance d and

hence the distance between any pair of codewords is at least d . Since $d \geq d-1$, the distance between each pair of codewords is also at least $d-1$. Hence, the codewords form a linear $\text{LRC}_q^{(r=1)}(n, k, d-1)$. $\square$

Example 4.1.6. The following is the generator matrix G of a linear $\text{LRC}_q^{(r=1)}(8, 3, 4)$.

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Notice that the weight of each codeword is 4 and the minimum distance between any pair of codewords is 4. Hence, the code generated by G is also a linear $\text{LRC}_q^{(r=1)}(8, 3, 3)$, $\text{LRC}_q^{(r=1)}(8, 3, 2)$ or $\text{LRC}_q^{(r=1)}(8, 3, 1)$ since it satisfies the definition that the minimum distance between any pair of codewords are at least 3, 2 or 1 respectively.

Corollary 4.1.7. Any linear $\text{LRC}_q^{(r=1)}(n, k, 2)$ is also a linear $\text{LRC}_q^{(r=1)}(n, k, 1)$.

Proof. This follows directly from Lemma 4.1.5. $\square$

For the rest of the thesis, we will only consider linear $\text{LRC}_q^{(r=1)}(n, k, d)$ where $d \geq 2$ since the optimality of linear $\text{LRC}_q^{(r=1)}(n, k, 1)$ follows from the optimality of linear $\text{LRC}_q^{(r=1)}(n, k, 2)$ by Corollary 4.1.7.

4.2 Bounds on Linear $\text{LRC}_q^{(r=1)}(n, k, d)$

In this section, we shall derive the upper bounds for linear $\text{LRC}_q^{(r=1)}(n, k, d)$ and also propose some lower bounds using propagation rules for linear $\text{LRC}_q^{(r=1)}(n, k, d)$ in the subsections.

4.2.1 Upper Bounds on Linear $\text{LRC}_q^{(r=1)}(n, k, d)$

Many of the upper bounds on linear $\text{LRC}_q^{(r=1)}(n, k, d)$ can be derived from the more general upper bounds on q -ary $[n, k, d]$ -linear codes. We will first state the trivial upper bound on the dimension of a linear $\text{LRC}_q^{(r=1)}(n, d)$.

Recall that $B_q(n, d)$ is the maximum size of a $[n, d]_q$ -linear code. Let us denote $M_q^{(r=1)}(n, d)$ to be the maximum dimension of a linear $\text{LRC}_q^{(r=1)}(n, d)$ and let $M_q(n, d) := \log_q B_q(n, d)$. Then for any linear $\text{LRC}_q^{(r=1)}(n, d)$,

$$M_q^{(r=1)}(n, d) \leq M_q(n, d). \quad (4.1)$$

From Theorems 3.2.2, 3.2.4 and 3.2.5 shown in Section 3.2, we can derive a better upper bound for linear $\text{LRC}_q^{(r=1)}(n, k, d)$.

Corollary 4.2.1. *For any linear $\text{LRC}_q^{(r=1)}(n, k, d)$, the minimum distance d satisfies*

$$d \leq n - 2k + 2, \quad (4.2)$$

and the dimension k is bounded above as follows:

$$k \leq \left\lfloor \frac{n - d}{2} \right\rfloor + 1. \quad (4.3)$$

Proof. For a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ where $r = 1$ and $t = 1$, each of the upper bounds (3.4), (3.6) and (3.8) reduces to (4.2). By making the integer dimension k a subject, we can rewrite (4.2) to obtain (4.3). $\square$

Lastly, from Theorem 4, we can derive an even tighter upper bound on the dimension of a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ as follows.

Corollary 4.2.2. *For any linear $\text{LRC}_q^{(r=1)}(n, k, d)$,*

$$k \leq \min_{1 \leq j \leq \left\lfloor \frac{n}{2} \right\rfloor} \{j + k_q(n - 2j, d)\}. \quad (4.4)$$

Proof. For a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ where $r = 1$ and $t = 1$, the upper bound (3.5) reduces to (4.4). $\square$

4.2.2 Lower Bounds on Linear $\text{LRC}_q^{(r=1)}(n, k, d)$

In this subsection, we extend the propagation rules for linear codes found in Section 2.3 to linear $\text{LRC}_q^{(r=1)}(n, k, d)$.

Theorem 4.2.3. *(Lengthening a linear LRC) Suppose there is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$. Then there must exist a linear $\text{LRC}_q^{(r=1)}(n + s, k, d)$ or linear $\text{LRC}_q^{(r=1)}(n + s, k, d + 1)$ for any $s \geq 1$.*

Proof. By mathematical induction, it suffices to show the existence of a linear $\text{LRC}_q^{(r=1)}(n + 1, k, d)$ or linear $\text{LRC}_q^{(r=1)}(n + 1, k, d + 1)$. We add a new coordinate that takes on the value of any of the other coordinates to all the codewords of $\mathcal{C}$ to form a new code,

$$\{(x_1, \dots, x_n, x_i) \mid 1 \leq i \leq n, (x_1, \dots, x_n) \in \mathcal{C}\}.$$

The above code satisfies Lemma 4.1.1 and is either a linear $\text{LRC}_q^{(r=1)}(n+1, k, d)$ or a linear $\text{LRC}_q^{(r=1)}(n+1, k, d+1)$. $\square$

Theorem 4.2.4. (Subcodes of a linear LRC) Suppose there is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$. Then there exists a linear $\text{LRC}_q^{(r=1)}(n, k-s, d)$ for any $1 \leq s \leq k-1$.

Proof. Let $\mathcal{C}$ be a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ and let $\mathbf{x}$ be a nonzero codeword of $\mathcal{C}$ with $wt(\mathbf{x}) = d$. We extend $\mathbf{x}$ to form a basis of $\mathcal{C}$: $\{\mathbf{x}_1 = \mathbf{x}, \dots, \mathbf{x}_k\}$. Consider the new code $\langle \{\mathbf{x}, \dots, \mathbf{x}_{k-s}\} \rangle$ spanned by the first $k-s$ codewords in the basis. These $k-s$ codewords can form the $k-s$ rows of a generator matrix G with rank $k-s$ and length n . Then G is the generator matrix of a linear $\text{LRC}_q^{(r=1)}(n, k-s, d)$. $\square$

Theorem 4.2.5. (Direct Sum) Let $\mathcal{C}_i$ be a linear $\text{LRC}_q^{(r=1)}(n_i, k_i, d_i)$ for $i = 1, 2$. Then the direct sum of $\mathcal{C}_1$ and $\mathcal{C}_2$ defined by

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(\mathbf{u}, \mathbf{v}) : \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$$

is a linear $\text{LRC}_q^{(r=1)}(n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\})$.

Proof. By Theorem 2.3.4 and Remark 2.3.5, we find that the generator matrix of $\mathcal{C}_1 \oplus \mathcal{C}_2$ satisfies Corollary 4.1.4, hence $\mathcal{C}_1 \oplus \mathcal{C}_2$ is linear $\text{LRC}_q^{(r=1)}(n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\})$. $\square$

Theorem 4.2.6. Let $\mathcal{C}$ be a linear $\text{LRC}_q^{(r=1)}(n, k, d)$. Then the code $\mathcal{C}$ defined by

$$\mathcal{C} = \{(\mathbf{u}, \mathbf{i} + \mathbf{u}) : \mathbf{u} \in \mathcal{C}, \text{ where } \mathbf{i} \text{ is the all-1 vector for } 0 \leq i \leq q-1\}$$

is a linear $\text{LRC}_q^{(r=1)}(2n, k+1, \min\{n, 2d\})$.

Proof. By considering Theorem 2.3.6 and taking $\mathcal{C}_1 = \mathcal{C}$ and $\mathcal{C}_2 = \{\mathbf{0}, \mathbf{1}, \dots, \mathbf{q}-\mathbf{1}\}$, we obtain a $[2n, k+1, \min\{n, 2d\}]_q$ -linear code defined as $\mathcal{C}$. From Remark 2.3.7, if we let G be a generator matrix of $\mathcal{C}$, we see that the matrix below

$$\begin{pmatrix} G & G \\ 0 \cdots 0 & 1 \cdots 1 \end{pmatrix}$$

is a generator matrix of $\mathcal{C}$. We observe that this generator matrix satisfies Corollary 4.1.4, therefore $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\})$. $\square$

Theorem 4.2.7. (Concatenation) Let $\mathcal{C}_i$ be a linear $\text{LRC}_q^{(r=1)}[n_i, k_i, d_i]$ for $i = 1, 2$ where $k_1 \leq k_2$. Then the concatenation of codewords of $\mathcal{C}_1$ and $\mathcal{C}_2$ form a code defined by

$$\mathcal{C}_1 \| \mathcal{C}_2 = \{(\mathbf{0}, \mathbf{0}), (\mathbf{x}_l, \mathbf{y}_l) : \mathbf{0}, \mathbf{x}_l \in \mathcal{C}_1, \mathbf{0}, \mathbf{y}_l \in \mathcal{C}_2 \text{ for } 1 \leq l \leq 2^{k_1} - 1\}$$

is a linear $\text{LRC}_q^{(r=1)}(n_1 + n_2, k_1, d_1 + e)$ where $e = \min_{1 \leq l \leq 2^{k_1} - 1} wt(\mathbf{y}_l)$.

Proof. Let G_i be a generator matrix of $\mathcal{C}_i$ for $i = 1, 2$ shown below where $k_1 \leq k_2$.

$$G_1 = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{k_1} \end{bmatrix}, \quad G_2 = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{k_2} \end{bmatrix}$$

From the above construction of code $\mathcal{C}_1 \parallel \mathcal{C}_2$, we see that the matrix

$$G = \begin{bmatrix} \mathbf{x}_1 & \mathbf{y}_1 \\ \mathbf{x}_2 & \mathbf{y}_2 \\ \vdots & \vdots \\ \mathbf{x}_{k_1} & \mathbf{y}_{k_1} \end{bmatrix}$$

forms a generator matrix of $\mathcal{C}_1 \parallel \mathcal{C}_2$. Since all rows in G_1 and G_2 are linearly independent, all rows of G are also linearly independent. Hence G is a $k_1 \times (n_1 + n_2)$ matrix with rank k_1 . The distance for $\mathcal{C}_1$ is d_1 hence $wt(\mathbf{x}) \geq d_1$ for $\mathbf{x} \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$. The minimum weight for all codewords $\mathbf{y}_l$ is $e = \min_{1 \leq l \leq 2^{k_1}-1} wt(\mathbf{y}_l)$. Therefore we must have $d(\mathcal{C}_1 \parallel \mathcal{C}_2) = wt((\mathbf{x}_l, \mathbf{y}_l)) \geq d_1 + e$ where $\mathbf{x}_l \in \mathcal{C}_1, \mathbf{y}_l \in \mathcal{C}_2$ for $1 \leq l \leq 2^{k_1} - 1$. Therefore, $\mathcal{C}_1 \parallel \mathcal{C}_2$ is a linear $LRC_q^{(r=1)}(n_1 + n_2, k_1, d_1 + e)$. $\square$

Remark 4.2.8. Theorem 4.2.7 should not be confused with concatenated codes.

4.3 Optimal Linear $LRC_q^{(r=1)}(n, k, d)$

In this section, we shall prove the optimal dimension k for certain values of length n and minimum distance d of a linear $LRC_q^{(r=1)}(n, d)$.

Theorem 4.3.1. $M_q^{(r=1)}(n, d) = 1$ for $2 \leq d \leq n < \left\lceil \frac{(q+1)d}{q} \right\rceil$.

Proof. $M_q^{(r=1)}(n, d) \geq 1$ for $2 \leq d \leq n < \left\lceil \frac{(q+1)d}{q} \right\rceil$ is proved by construction using a generator matrix G of a linear $LRC_q^{(r=1)}(n, d)$ for $2 \leq d \leq n < \left\lceil \frac{(q+1)d}{q} \right\rceil$. The weight of each codeword is d and the distance between any pair of codewords is d . Note that each non-zero symbol of each coordinate is replicated d times and each zero symbol is replicated $n - d$ times to satisfy Lemma (4.1.1). The generator matrix G is shown below:

$$G = \begin{bmatrix} \overbrace{1 \cdots 1}^d & \overbrace{0 \cdots 0}^{n-d} \end{bmatrix}$$

The exception here is when $d = n - 1$, in this case, we let the linear $LRC_q^{(r=1)}(n, n - 1)$ be the repetition code, linear $LRC_q^{(r=1)}(n, n)$ by Lemma 4.1.5.

The upper bound $M_q^{(r=1)}(n, d) \leq 1$ for $2 \leq d \leq n < \left\lceil \frac{(q+1)d}{q} \right\rceil$ can be found by using the Griesmer bound (2.6) from Theorem 2.2.6. It states that for any $[n, k, d]_q$ -linear code with

$k \geq 2$, the length of the code n has to satisfy the following:

$$\begin{aligned}
n &\geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil \\
&\geq d + \left\lceil \frac{d}{q} \right\rceil \\
&= \left\lceil d + \frac{d}{q} \right\rceil \\
&= \left\lceil \frac{(q+1)d}{q} \right\rceil.
\end{aligned}$$

Therefore for $2 \leq d \leq n < \left\lceil \frac{(q+1)d}{q} \right\rceil$, $k \leq 1$ and from (4.1), $M_q^{(r=1)}(n, d) \leq M_q(n, d) \leq 1$ for $2 \leq d \leq n < \left\lceil \frac{(q+1)d}{q} \right\rceil$. $\square$

Theorem 4.3.2. $M_q^{(r=1)}((q+1)m, qm) = 2$ for $m \geq 2$.

Proof. $M_q^{(r=1)}((q+1)m, qm) \geq 2$ for $m \geq 2$ is proved by construction of a generator matrix G of a linear $\text{LRC}_q^{(r=1)}((q+1)m, qm)$ for $m \geq 2$. The weight of each codeword is qm and the distance between any pair of codewords is qm . Note that it is necessary for $m \geq 2$ so that the symbols of each coordinate can be replicated to satisfy Lemma (4.1.1). The $2 \times (q+1)m$ generator matrix G is shown below:

$$G = \underbrace{\begin{bmatrix} \overbrace{1 \dots 1}^m & \overbrace{0 \dots 0}^m & \overbrace{1 \dots 1}^m & \overbrace{1 \dots 1}^m & \dots & \overbrace{1 \dots 1}^m \\ 0 \dots 0 & 1 \dots 1 & 1 \dots 1 & 2 \dots 2 & \dots & (q-1) \dots (q-1) \end{bmatrix}}_{(q+1) \text{ blocks of } m \text{ columns}}$$

The upper bound $M_q^{(r=1)}((q+1)m, qm) \leq 2$ for $m \geq 2$ can be found by using the Griesmer bound (2.6) from Theorem 2.2.6. It states that for any $[n, k, d]_q$ -linear code with $k \geq 3$ and $d = qm$ for $m \geq 2$, the length of the code n has to satisfy the following:

$$\begin{aligned}
n &\geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil \\
&\geq d + \left\lceil \frac{d}{q} \right\rceil + \left\lceil \frac{d}{q^2} \right\rceil \\
&= qm + m + \left\lceil \frac{m}{q} \right\rceil \\
&\geq (q+1)m + 1.
\end{aligned}$$

But $n = (q+1)m$, therefore, $M_q((q+1)m, qm) = k \leq 2$ and from (4.1), $M_q^{(r=1)}((q+1)m, qm) \leq M_q((q+1)m, qm) \leq 2$ for $m \geq 2$. $\square$

Theorem 4.3.3. $M_q^{(r=1)}((q+1)m + 1, qm) = 2$ for $m > q$.

Proof. $M_q^{(r=1)}((q+1)m + 1, qm) \geq 2$ for $m > q$ follows directly from Theorem (4.3.2) by replicating the symbols in only one particular coordinate of a linear $\text{LRC}_q^{(r=1)}((q+1)m, qm)$

for $m > q$. We show a $2 \times ((q+1)m+1)$ generator matrix G of a linear $\text{LRC}_q^{(r=1)}((q+1)m+1, qm)$ for $m > q$ below:

$$G = \underbrace{\begin{bmatrix} \overbrace{1 \dots 1}^m & \overbrace{0 \dots 0}^m & \overbrace{1 \dots 1}^m & \overbrace{1 \dots 1}^m & \dots & \overbrace{1 \dots 1}^{m+1} \\ \overbrace{0 \dots 0}^m & \overbrace{1 \dots 1}^m & \overbrace{1 \dots 1}^m & \overbrace{2 \dots 2}^m & \dots & \overbrace{(q-1) \dots (q-1)}^{m+1} \end{bmatrix}}_{q \text{ blocks of } m \text{ columns and a block of } m+1 \text{ column}}$$

Note that all the codewords have weight at least qm and distance between each pair of codewords is at least qm . The upper bound $M_q^{(r=1)}((q+1)m, qm) \leq 2$ can be found by using the Griesmer bound (2.6) from Theorem 2.2.6. It states that for any $[n, k, d]_q$ -linear code with $k \geq 3$ and $d = qm$ for $m > q$, the length of the code n has to satisfy the following:

$$\begin{aligned} n &\geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil \\ &\geq d + \left\lceil \frac{d}{q} \right\rceil + \left\lceil \frac{d}{q^2} \right\rceil \\ &= qm + m + \left\lceil \frac{m}{q} \right\rceil \\ &\geq (q+1)m + 2. \end{aligned}$$

But $n = (q+1)m+1$, therefore, $M_q((q+1)m+1, qm) = k \leq 2$ and from (4.1), $M_q^{(r=1)}((q+1)m+1, qm) \leq M_q((q+1)m+1, qm) \leq 2$ for $m > q$. $\square$

Theorem 4.3.4. $M_q^{(r=1)}((q+1)m-p, qm-p) = 2$ for $1 \leq p \leq q-1$ and $m \geq p+2$.

Proof. $M_q^{(r=1)}((q+1)m-p, qm-p) \geq 2$ for $m \geq p+2$ is proved by construction of a generator matrix G of a linear $\text{LRC}_q^{(r=1)}((q+1)m, qm-1)$ for $m \geq p+2$. The weight of each codeword is at least $qm-p$ and the distance between any pair of codewords is at least $qm-p$. Note that it is necessary for $m-p \geq 2$ so that the symbols of each coordinate can be replicated to satisfy Lemma 4.1.1. The $2 \times ((q+1)m-p)$ generator matrix G is shown below:

$$G = \underbrace{\begin{bmatrix} \overbrace{1 \dots 1}^m & \overbrace{0 \dots 0}^m & \overbrace{1 \dots 1}^{m-p} & \overbrace{1 \dots 1}^m & \dots & \overbrace{1 \dots 1}^m \\ \overbrace{0 \dots 0}^m & \overbrace{1 \dots 1}^m & \overbrace{1 \dots 1}^{m-p} & \overbrace{2 \dots 2}^m & \dots & \overbrace{(q-1) \dots (q-1)}^m \end{bmatrix}}_{q \text{ blocks of } m \text{ columns and a block of } m-p \text{ column}}$$

The upper bound $M_q^{(r=1)}((q+1)m-p, qm-p) \leq 2$ for $m \geq p+2$ can be found by using the Griesmer bound (2.6) from Theorem 2.2.6. It states that for any $[n, k, d]_q$ -linear code with $k \geq 3$ and $d = qm-p$ for $m \geq p+2$, the length of the code n has to satisfy the following:

$$\begin{aligned}
n &\geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil \\
&\geq d + \left\lceil \frac{d}{q} \right\rceil + \left\lceil \frac{d}{q^2} \right\rceil \\
&= qm - p + \left\lceil m - \frac{p}{q} \right\rceil + \left\lceil \frac{m - \frac{p}{q}}{q} \right\rceil \\
&\geq qm - p + m + \left\lceil \frac{m-1}{q} \right\rceil \quad \text{since } q \geq 2 \\
&\geq (q+1)m - p + 1.
\end{aligned}$$

But $n = (q+1)m - p$, therefore, $M_q((q+1)m - p, qm - p) = k \leq 2$ and from (4.1), $M_q^{(r=1)}((q+1)m - p, qm - p) \leq M_q((q+1)m - p, qm - p) \leq 2$ for $m \geq p+2$. $\square$

Theorem 4.3.5. $M_q^{(r=1)}(n, 2) = \lfloor \frac{n}{2} \rfloor$ for $n \geq 4$.

Proof. $M_q^{(r=1)}(n, 2) \geq \lfloor \frac{n}{2} \rfloor$ for $n \geq 4$ is proved by construction.

The maximum dimension of a $[m, 1]_q$ -linear code is $M_q(m, 1) = m$ for all $m \geq 1$ which is a $[m, m, 1]_q$ MDS code. By replicating each symbol of the coordinates of the $[m, m, 1]_q$ MDS code once, we will obtain a $[2m, m, 2]_q$ -linear code, which is also a linear $\text{LRC}_q^{(r=1)}(2m, m, 2)$, for $m \geq 2$. By replicating the symbols in only one particular coordinate of the linear $\text{LRC}_q^{(r=1)}(2m, m, 2)$ again, we will obtain a linear $\text{LRC}_q^{(r=1)}(2m+1, m, 2)$, for $m \geq 2$. Then we have

$$\begin{aligned}
M_q^{(r=1)}(2m, 2) &\geq m \text{ and } M_q^{(r=1)}(2m+1, 2) \geq m \text{ for all } m \geq 2, \\
\text{hence } M_q^{(r=1)}(n, 2) &\geq \frac{n}{2} \text{ for all even } n \geq 4, \\
\text{and } M_q^{(r=1)}(n, 2) &\geq \frac{n-1}{2} \text{ for all odd } n \geq 5, \\
\text{Therefore, } M_q^{(r=1)}(n, 2) &\geq \left\lfloor \frac{n}{2} \right\rfloor \text{ for } n \geq 4.
\end{aligned}$$

$M_q^{(r=1)}(n, 2) \leq \lfloor \frac{n}{2} \rfloor$ for $n \geq 4$ is derived using $d = 2$ from the upper bound (4.3) stated in Theorem 4.2.1. $\square$

Theorem 4.3.6. $M_q^{(r=1)}(n, 4) = \left\lfloor \frac{n-2}{2} \right\rfloor$ for $n \geq 6$.

Proof. $M_q^{(r=1)}(n, 4) \geq \left\lfloor \frac{n-2}{2} \right\rfloor$ for $n \geq 6$ is proved by construction.

The maximum dimension of a $[m, 2]_q$ -linear code is $M_q(m, 2) = m - 1$ for all $m \geq 2$ which is a $[m, m-1, 2]_q$ MDS code. By replicating each symbol of the coordinates of the $[m, m-1, 2]_q$ MDS code once, we will obtain a $[2m, m-1, 4]_q$ -linear code, which is also a linear $\text{LRC}_q^{(r=1)}(2m, m-1, 4)$, for $m \geq 3$. By replicating the symbols in only one particular coordinate of the linear $\text{LRC}_q^{(r=1)}(2m, m-1, 4)$ again, we will obtain a linear $\text{LRC}_q^{(r=1)}(2m+1, m-1, 4)$, for $m \geq 3$.

$1, m-1, 4)$, for $m \geq 3$. Then we have

$$\begin{aligned}
M_q^{(r=1)}(2m, 4) &\geq m-1 \text{ and } M_q^{(r=1)}(2m+1, 4) \geq m-1 \text{ for all } m \geq 3, \\
\text{hence } M_q^{(r=1)}(n, 4) &\geq \frac{n-2}{2} \text{ for all even } n \geq 6, \\
\text{and } M_q^{(r=1)}(n, 4) &\geq \frac{n-3}{2} \text{ for all odd } n \geq 7, \\
\text{Therefore, } M_q^{(r=1)}(n, 4) &\geq \left\lfloor \frac{n-2}{2} \right\rfloor \text{ for } n \geq 6.
\end{aligned}$$

$M_q^{(r=1)}(n, 4) \leq \left\lfloor \frac{n-2}{2} \right\rfloor$ for $n \geq 6$ is derived using $d = 4$ from the upper bound (4.3) stated in Theorem 4.2.1. $\square$

Theorem 4.3.7. $M_q^{(r=1)}(n, 3) = \frac{n-2}{2}$ for all even $n \geq 6$.

Proof. $M_q^{(r=1)}(n, 3) \geq \frac{n-2}{2}$ for all even $n \geq 6$ follows directly from Theorem 4.3.6 and Lemma 4.1.5 by letting $s = 1$.

$M_q^{(r=1)}(n, 3) \leq \frac{n-2}{2}$ for all even $n \geq 6$ is derived from the upper bound (4.3) stated in Theorem 4.2.1 by letting $d = 3$ and rewriting it for even $n \geq 6$. $\square$

Chapter 5

Code Construction using Partitions for Linear $\text{LRC}_q^{(r=1,t)}(n, d)$

After looking through Section 4.3, we realise that there are still not many optimal linear $\text{LRC}_q^{(r=1)}(n, k, d)$ known for large values of length n and minimum distance d . In this chapter, we will explore code constructions using partitions for linear $\text{LRC}_q^{(r=1,t)}(n, d)$ that proves to be a good lower bound for $M_q^{(r=1,t)}(n, d)$.

In Section 5.2, We will first present the code construction for linear $\text{LRC}_q^{(r=1)}(n, d)$ by considering partitions of n . This code construction technique allows us to find a good lower bound for $M_q^{(r=1)}(n, d)$ as we see that in Section 5.3, where we investigate the tightness of this bound on the binary linear $\text{LRC}^{(r=1)}(n, d)$ for $2 \leq d \leq n \leq 32$ by comparing with the upper bounds mentioned in the Section 4.2. Lastly, we will extend the code construction discussed for linear $\text{LRC}_q^{(r=1)}(n, d)$ to linear $\text{LRC}_q^{(r=1,t)}(n, d)$ by considering the availability t of the LRC.

5.1 Preliminaries

Definition 5.1.1. If n is a positive integer, then an *integer partition* or *partition* of n is a non-increasing sequence of positive integers $(p_1, p_2, \dots, p_m)$, where $m \leq n$, whose sum is n . Each p_i is called a *part* of the partition.

Remark 5.1.2. Note that permutation of coordinates of a linear $\text{LRC}_q^{(r,t)}(n, k, d)$ does not alter the property of locality or availability. By Definition 2.1.3, if a linear $\text{LRC}_q^{(r,t)}(n, k, d)$ can be obtained from another linear $\text{LRC}_q^{(r,t)}(n, k, d)$ by an arbitrary permutation of coordinates, they are equivalent. Hence, we can work with linear $\text{LRC}_q^{(r,t)}(n, k, d)$ with its relationship to partitions in a non-increasing sequence.

5.2 Code Construction using Partitions for Linear $\text{LRC}_q^{(r=1)}(n, d)$

Definition 5.2.1. Given n and d , define a vector $(p_1, p_2, \dots, p_N)$ to be an $(n, d; N, D)$ -partition, if

$$\begin{aligned} (i) \quad & \sum_{i=1}^N p_i = n; \\ (ii) \quad & p_i \geq 2, \quad \text{for } 1 \leq i \leq N; \\ (iii) \quad & \min_{I \in \binom{[N]}{D}} \sum_{i \in I} p_i \geq d \quad \text{where } D \leq N. \end{aligned}$$

Remark 5.2.2. A $(n, d; N, D)$ -partition must have $N \leq \lfloor \frac{n}{2} \rfloor$.

The code construction for a linear $\text{LRC}_q^{(r=1)}(n, k, d)$ is presented in the lemma below.

Lemma 5.2.3. (*Code Construction for a linear $\text{LRC}_q^{(r=1)}(n, k, d)$)* Given n and d , for any $(n, d; N, D)$ -partition $(p_1, p_2, \dots, p_N)$, we can use a $[N, k, D]_q$ -linear code to construct a linear $\text{LRC}_q^{(r=1)}(n, k, d)$.

Proof. Given n and d , for any $(n, d; N, D)$ -partition $(p_1, p_2, \dots, p_N)$, let H be the generator matrix of a $[N, k, D]_q$ -linear code with dimension k where g_i for $1 \leq i \leq N$ represents the N columns of H shown below.

$$H = \left[\begin{array}{c|c|c|c} | & | & & | \\ g_1 & g_2 & \cdots & g_N \\ | & | & & | \end{array} \right] \Bigg\}^k$$

By replicating each g_i column $p_i - 1$ times, we can obtain the following generator matrix G .

$$G = \left[\begin{array}{c|c|c|c} \overbrace{\begin{array}{c|c} | & | \end{array}}^{p_1} & \overbrace{\begin{array}{c|c} | & | \end{array}}^{p_2} & & \overbrace{\begin{array}{c|c} | & | \end{array}}^{p_N} \\ g_1 \cdots g_1 & g_2 \cdots g_2 & \cdots & g_N \cdots g_N \\ | & | & & | \end{array} \right] \Bigg\}^k \quad (5.1)$$

Let $\mathcal{C}$ be the linear code generated by G and we shall prove that $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1)}(n, k, d)$. For every i^{th} partition, the columns g_i are the same, hence the symbols of the coordinates restricted to the partition are the same. Since each partition $p_i \geq 2$ by Definition 5.2.1(ii), $\mathcal{C}$ satisfies the property of locality $r = 1$ by Lemma 4.1.1. The length of $\mathcal{C}$ is the sum of all p_i which is n by Definition 5.2.1(i), and the dimension of the code is k since each rows in G are still linearly independent. Since the minimum distance of the code generated by H is D , the weight of any arbitrary non-zero codeword $(x_1, x_2, \dots, x_N)$ in the code generated by H must be at least D . By the above code construction, any arbitrary non-zero codeword in $\mathcal{C}$ will be of the form $\mathbf{x} = \underbrace{x_1 \cdots x_1}_{p_1} \underbrace{x_2 \cdots x_2}_{p_2} \cdots \underbrace{x_N \cdots x_N}_{p_N}$ and will have weight at least d by Definition 5.2.1(iii). Therefore the minimum distance of $\mathcal{C}$ is d . $\square$

Theorem 5.2.4. *Given n and d , a lower bound by considering partitions of a linear $\text{LRC}_q^{(r=1)}(n, d)$ is given as:*

$$M_q^{(r=1)}(n, d) \geq \max_{(N, D) \in P} M_q(N, D), \quad (5.2)$$

where $P = \{(N, D) : \text{all } (n, d; N, D)\text{-partitions}\}$.

Proof. Given n, d we can find all the (N, D) pairs from the finite number of $(n, d; N, D)$ -partitions. Then by Lemma 5.2.3 we can construct any linear $\text{LRC}_q^{(r=1)}(n, k, d)$ from a $[N, k, D]_q$ -linear code. Hence, the lower bound for the maximum dimension of a linear $\text{LRC}_q^{(r=1)}(n, d)$ with length n and distance d follows from the maximum dimension of a $[N, D]_q$ -linear code with length N and distance D that satisfies Definition 5.2.1. $\square$

5.3 Tightness of Bound on Binary Linear $\text{LRC}^{(r=1)}(n, d)$

In this section, we investigate the tightness of the lower bound (5.2) on binary linear $\text{LRC}^{(r=1)}(n, d)$ by comparing it with upper bounds stated in Section 4.2. We start off with an example to use our lower bound (5.2) to determine the optimal dimension of a binary linear $\text{LRC}^{(r=1)}(n, d)$ with small parameters.

Example 5.3.1. In Example (2.1.2), we have a binary $[5, 2, 3]$ -linear code $\mathcal{C}$, represented by its generator matrix G .

$$G = \begin{pmatrix} & a & b & c & d & e \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Note that this is the only unique $[5, 2, 3]$ -linear code up to the equivalence of linear codes. We shall check if this is a linear $\text{LRC}^{(r=1)}(5, 2, 3)$ code.

We can observe from the generator matrix that symbols a and b can be recovered from one another, as well as d and e . But symbol c cannot be recovered from any of the other symbols. Since $\mathcal{C}$ does not satisfy Lemma 4.1.1, it is not a linear $\text{LRC}^{(r=1)}(5, 2, 3)$.

By Definition 5.2.1, there is only 1 possible partition for $n = 5$ and $d = 3$, which is the vector (5) and this is a $(5, 3; 1, 1)$ -partition. Hence from the above example and (5.2), the maximum dimension for a linear $\text{LRC}^{(r=1)}(5, 3)$ is 1.

We shall present a $(n, d; N, D)$ -partition for each value of $2 \leq d \leq n \leq 32$ and their respective maximum dimension $M(N, D)$ for a linear $\text{LRC}^{(r=1)}(n, d)$ (see lower bound (5.2)) with its corresponding values for (N, D) in the Tables 5.1 - 5.29. They can be transformed into a linear $\text{LRC}^{(r=1)}(n, d)$ using the generator matrix G in the proof of Lemma 5.2.3 and the generator matrix of an optimal $[N, D]$ -linear code.

We shall exclude all $(n, d; N, D)$ -partition where $M^{(r=1)}(n, d) = 1$ since their corresponding linear $\text{LRC}^{(r=1)}(n, d)$ are optimal by Theorem 4.3.1 and the optimal codes can be derived

using the construction in the proof in Theorem 4.3.1. The linear $\text{LRC}^{(r=1)}(5, 3)$ which we have shown in Example 5.3.1 is an exception for linear $\text{LRC}^{(r=1)}(n, d)$ that does not satisfy Theorem 4.3.1 but has $M^{(r=1)}(n, d) = 1$.

TABLE 5.1: $n = 4$

d	$(4, d; N, D)$ -partition	$M(N, D)$
2	$(4, 2; 2, 1) : (2, 2)$	$M(2, 1) = \mathbf{2}$

TABLE 5.2: $n = 5$

d	$(5, d; N, D)$ -partition	$M(N, D)$
2	$(5, 2; 2, 1) : (3, 2)$	$M(2, 1) = \mathbf{2}$
3	$(5, 3; 1, 1) : (5)$	$M(1, 1) = \mathbf{1}$

TABLE 5.3: $n = 6$

d	$(6, d; N, D)$ -partition	$M(N, D)$
2	$(6, 2; 3, 1) : (2, 2, 2)$	$M(3, 1) = \mathbf{3}$
3	$(6, 3; 2, 1) : (3, 3)$	$M(2, 1) = \mathbf{2}$
4	$(6, 4; 3, 2) : (2, 2, 2)$	$M(3, 2) = \mathbf{2}$

TABLE 5.4: $n = 7$

d	$(7, d; N, D)$ -partition	$M(N, D)$
2	$(7, 2; 3, 1) : (3, 2, 2)$	$M(3, 1) = \mathbf{3}$
3	$(7, 3; 2, 1) : (4, 3)$	$M(2, 1) = \mathbf{2}$
4	$(7, 4; 3, 2) : (3, 2, 2)$	$M(3, 2) = \mathbf{2}$

TABLE 5.5: $n = 8$

d	$(8, d; N, D)$ -partition	$M(N, D)$
2	$(8, 2; 4, 1) : (2, 2, 2, 2)$	$M(4, 1) = \mathbf{4}$
3	$(8, 3; 4, 2) : (2, 2, 2, 2)$	$M(4, 2) = \mathbf{2}$
4	$(8, 4; 4, 2) : (2, 2, 2, 2)$	$M(4, 2) = \mathbf{3}$
5	$(8, 5; 3, 2) : (3, 3, 2)$	$M(3, 2) = \mathbf{2}$

TABLE 5.6: $n = 9$

d	$(9, d; N, D)$ -partition	$M(N, D)$
2	$(9, 2; 4, 1) : (3, 2, 2, 2)$	$M(4, 1) = \mathbf{4}$
3	$(9, 3; 3, 1) : (3, 3, 3)$	$M(3, 1) = \mathbf{3}$
4	$(9, 4; 4, 2) : (3, 2, 2, 2)$	$M(4, 2) = \mathbf{3}$
5	$(9, 5; 3, 2) : (4, 3, 2)$	$M(3, 2) = \mathbf{2}$
6	$(9, 6; 3, 2) : (3, 3, 3)$	$M(3, 2) = \mathbf{2}$

TABLE 5.7: $n = 10$

d	$(10, d; N, D)$ -partition	$M(N, D)$
2	$(10, 2; 5, 1) : (2, 2, 2, 2, 2)$	$M(5, 1) = \mathbf{5}$
3	$(10, 3; 5, 2) : (2, 2, 2, 2, 2)$	$M(5, 2) = \mathbf{4}$
4	$(10, 4; 5, 2) : (2, 2, 2, 2, 2)$	$M(5, 2) = \mathbf{4}$
5	$(10, 5; 3, 2) : (5, 3, 2)$	$M(3, 2) = \mathbf{2}$
6	$(10, 6; 3, 2) : (4, 3, 3)$	$M(3, 2) = \mathbf{2}$

TABLE 5.8: $n = 11$

d	$(11, d; N, D)$ -partition	$M(N, D)$
2	$(11, 2; 5, 1) : (3, 2, 2, 2, 2)$	$M(5, 1) = \mathbf{5}$
3	$(11, 3; 5, 2) : (3, 2, 2, 2, 2)$	$M(5, 2) = \mathbf{4}$
4	$(11, 4; 5, 2) : (3, 2, 2, 2, 2)$	$M(5, 2) = \mathbf{4}$
5	$(11, 5; 4, 2) : (3, 3, 3, 2)$	$M(4, 2) = \mathbf{3}$
6	$(11, 6; 5, 3) : (3, 2, 2, 2, 2)$	$M(5, 3) = \mathbf{2}$
7	$(11, 7; 3, 2) : (4, 4, 3)$	$M(3, 2) = \mathbf{2}$

TABLE 5.9: $n = 12$

d	$(12, d; N, D)$ -partition	$M(N, D)$
2	$(12, 2; 6, 1) : (2, 2, 2, 2, 2, 2)$	$M(6, 1) = \mathbf{6}$
3	$(12, 3; 6, 2) : (2, 2, 2, 2, 2, 2)$	$M(6, 2) = \mathbf{5}$
4	$(12, 4; 6, 2) : (2, 2, 2, 2, 2, 2)$	$M(6, 2) = \mathbf{5}$
5	$(12, 5; 4, 2) : (4, 3, 3, 2)$	$M(4, 2) = \mathbf{3}$
6	$(12, 6; 4, 2) : (3, 3, 3, 3)$	$M(4, 2) = \mathbf{3}$
7	$(12, 7; 3, 2) : (5, 4, 3)$	$M(3, 2) = \mathbf{2}$
8	$(12, 8; 3, 2) : (4, 4, 4)$	$M(3, 2) = \mathbf{2}$

TABLE 5.10: $n = 13$

d	$(13, d; N, D)$ -partition	$M(N, D)$
2	$(13, 2; 6, 1) : (3, 2, 2, 2, 2, 2)$	$M(6, 1) = \mathbf{6}$
3	$(13, 3; 6, 2) : (3, 2, 2, 2, 2, 2)$	$M(6, 2) = \mathbf{5}$
4	$(13, 4; 6, 2) : (3, 2, 2, 2, 2, 2)$	$M(6, 2) = \mathbf{5}$
5	$(13, 5; 4, 2) : (5, 3, 3, 2)$	$M(4, 2) = \mathbf{3}$
6	$(13, 6; 4, 2) : (4, 3, 3, 3)$	$M(4, 2) = \mathbf{3}$
7	$(13, 7; 5, 3) : (3, 3, 3, 2, 2)$	$M(5, 3) = \mathbf{2}$
8	$(13, 8; 3, 2) : (5, 4, 4)$	$M(3, 2) = \mathbf{2}$

TABLE 5.11: $n = 14$

d	$(14, d; N, D)$ -partition	$M(N, D)$
2	$(14, 2; 7, 1) : (2, 2, 2, 2, 2, 2, 2)$	$M(7, 1) = \mathbf{7}$
3	$(14, 3; 7, 2) : (2, 2, 2, 2, 2, 2, 2)$	$M(7, 2) = \mathbf{6}$
4	$(14, 4; 7, 2) : (2, 2, 2, 2, 2, 2, 2)$	$M(7, 2) = \mathbf{6}$
5	$(14, 5; 5, 2) : (3, 3, 3, 3, 2)$	$M(5, 2) = \mathbf{4}$
6	$(14, 6; 7, 3) : (2, 2, 2, 2, 2, 2, 2)$	$M(7, 3) = \mathbf{4}$
7	$(14, 7; 7, 4) : (2, 2, 2, 2, 2, 2, 2)$	$M(7, 4) = \mathbf{3}$
8	$(14, 8; 7, 4) : (2, 2, 2, 2, 2, 2, 2)$	$M(7, 4) = \mathbf{3}$
9	$(14, 9; 3, 2) : (5, 5, 4)$	$M(3, 2) = \mathbf{2}$

TABLE 5.12: $n = 15$

d	$(15, d; N, D)$ -partition	$M(N, D)$
2	$(15, 2; 7, 1) : (3, 2, 2, 2, 2, 2, 2)$	$M(7, 1) = \mathbf{7}$
3	$(15, 3; 7, 2) : (3, 2, 2, 2, 2, 2, 2)$	$M(7, 2) = \mathbf{6}$
4	$(15, 4; 7, 2) : (3, 2, 2, 2, 2, 2, 2)$	$M(7, 2) = \mathbf{6}$
5	$(15, 5; 5, 2) : (4, 3, 3, 3, 2)$	$M(5, 2) = \mathbf{4}$
6	$(15, 6; 5, 2) : (3, 3, 3, 3, 3)$	$M(5, 2) = \mathbf{4}$
7	$(15, 7; 4, 2) : (4, 4, 4, 3)$	$M(4, 2) = \mathbf{3}$
8	$(15, 8; 7, 4) : (3, 2, 2, 2, 2, 2, 2)$	$M(7, 4) = \mathbf{3}$
9	$(15, 9; 3, 2) : (6, 5, 4)$	$M(3, 2) = \mathbf{2}$
10	$(15, 10; 3, 2) : (5, 5, 5)$	$M(3, 2) = \mathbf{2}$

TABLE 5.13: $n = 16$

d	$(16, d; N, D)$ -partition	$M(N, D)$
2	$(16, 2; 8, 1) : (2, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 1) = \mathbf{8}$
3	$(16, 3; 8, 2) : (2, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 2) = \mathbf{7}$
4	$(16, 4; 8, 2) : (2, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 2) = \mathbf{7}$
5	$(16, 5; 5, 2) : (4, 4, 3, 3, 2)$	$M(5, 2) = \mathbf{4}$
6	$(16, 6; 5, 2) : (4, 3, 3, 3, 3)$	$M(5, 2) = \mathbf{4}$
7	$(16, 7; 8, 4) : (2, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 4) = \mathbf{4}$
8	$(16, 8; 8, 4) : (2, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 4) = \mathbf{4}$
9	$(16, 9; 5, 3) : (4, 3, 3, 3, 3)$	$M(5, 3) = \mathbf{2}$
10	$(16, 10; 3, 2) : (6, 5, 5)$	$M(3, 2) = \mathbf{2}$

TABLE 5.14: $n = 17$

d	$(17, d; N, D)$ -partition	$M(N, D)$
2	$(17, 2; 8, 1) : (3, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 1) = \mathbf{8}$
3	$(17, 3; 8, 2) : (3, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 2) = \mathbf{7}$
4	$(17, 4; 8, 2) : (3, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 2) = \mathbf{7}$
5	$(17, 5; 6, 2) : (3, 3, 3, 3, 3, 2)$	$M(6, 2) = \mathbf{5}$
6	$(17, 6; 8, 3) : (3, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 3) = \mathbf{4}$
7	$(17, 7; 8, 4) : (3, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 4) = \mathbf{4}$
8	$(17, 8; 8, 4) : (3, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 4) = \mathbf{4}$
9	$(17, 9; 5, 3) : (4, 4, 3, 3, 3)$	$M(5, 3) = \mathbf{2}$
10	$(17, 10; 8, 5) : (3, 2, 2, 2, 2, 2, 2, 2)$	$M(8, 5) = \mathbf{2}$
11	$(17, 11; 3, 2) : (6, 6, 5)$	$M(3, 2) = \mathbf{2}$

TABLE 5.15: $n = 18$

d	$(18, d; N, D)$ -partition	$M(N, D)$
2	$(18, 2; 9, 1) : (2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 1) = \mathbf{9}$
3	$(18, 3; 9, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 2) = \mathbf{8}$
4	$(18, 4; 9, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 2) = \mathbf{8}$
5	$(18, 5; 6, 2) : (4, 3, 3, 3, 3, 2)$	$M(6, 2) = \mathbf{5}$
6	$(18, 6; 9, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 3) = \mathbf{5}$
7	$(18, 7; 9, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 4) = \mathbf{4}$
8	$(18, 8; 9, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 4) = \mathbf{4}$
9	$(18, 9; 6, 3) : (3, 3, 3, 3, 3, 3)$	$M(6, 3) = \mathbf{3}$
10	$(18, 10; 9, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 5) = \mathbf{2}$
11	$(18, 11; 3, 2) : (7, 6, 5)$	$M(3, 2) = \mathbf{2}$
12	$(18, 12; 3, 2) : (6, 6, 6)$	$M(3, 2) = \mathbf{2}$

TABLE 5.16: $n = 19$

d	$(19, d; N, D)$ -partition	$M(N, D)$
2	$(19, 2; 9, 1) : (3, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 1) = \mathbf{9}$
3	$(19, 3; 9, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 2) = \mathbf{8}$
4	$(19, 4; 9, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 2) = \mathbf{8}$
5	$(19, 5; 6, 2) : (4, 4, 3, 3, 3, 2)$	$M(6, 2) = \mathbf{5}$
6	$(19, 6; 9, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 3) = \mathbf{5}$
7	$(19, 7; 5, 2) : (4, 4, 4, 4, 3)$	$M(5, 2) = \mathbf{4}$
8	$(19, 8; 9, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(9, 4) = \mathbf{4}$
9	$(19, 9; 6, 3) : (4, 3, 3, 3, 3, 3)$	$M(6, 3) = \mathbf{3}$
10	$(19, 10; 7, 4) : (3, 3, 3, 3, 3, 2, 2)$	$M(7, 4) = \mathbf{3}$
11	$(19, 11; 6, 4) : (4, 4, 3, 3, 3, 2)$	$M(6, 4) = \mathbf{2}$
12	$(19, 12; 3, 2) : (7, 6, 6)$	$M(3, 2) = \mathbf{2}$

TABLE 5.17: $n = 20$

d	$(20, d; N, D)$ -partition	$M(N, D)$
2	$(20, 2; 10, 1) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 1) = \mathbf{10}$
3	$(20, 3; 10, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 2) = \mathbf{9}$
4	$(20, 4; 10, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 2) = \mathbf{9}$
5	$(20, 5; 7, 2) : (3, 3, 3, 3, 3, 3, 2)$	$M(7, 2) = \mathbf{6}$
6	$(20, 6; 10, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 3) = \mathbf{6}$
7	$(20, 7; 10, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 4) = \mathbf{5}$
8	$(20, 8; 10, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 4) = \mathbf{5}$
9	$(20, 9; 6, 3) : (4, 4, 3, 3, 3, 3)$	$M(6, 3) = \mathbf{3}$
10	$(20, 10; 10, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 5) = \mathbf{3}$
11	$(20, 11; 7, 4) : (3, 3, 3, 3, 3, 3, 2)$	$M(7, 4) = \mathbf{3}$
12	$(20, 12; 10, 6) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 6) = \mathbf{2}$
13	$(20, 13; 3, 2) : (7, 7, 6)$	$M(3, 2) = \mathbf{2}$

TABLE 5.18: $n = 21$

d	$(21, d; N, D)$ -partition	$M(N, D)$
2	$(21, 2; 10, 1) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 1) = \mathbf{10}$
3	$(21, 3; 10, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 2) = \mathbf{9}$
4	$(21, 4; 10, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 2) = \mathbf{9}$
5	$(21, 5; 7, 2) : (4, 3, 3, 3, 3, 3, 2)$	$M(7, 2) = \mathbf{6}$
6	$(21, 6; 10, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 3) = \mathbf{6}$
7	$(21, 7; 10, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 4) = \mathbf{5}$
8	$(21, 8; 10, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 4) = \mathbf{5}$
9	$(21, 9; 7, 3) : (3, 3, 3, 3, 3, 3, 3)$	$M(7, 3) = \mathbf{4}$
10	$(21, 10; 10, 5) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(10, 5) = \mathbf{3}$
11	$(21, 11; 7, 4) : (4, 3, 3, 3, 3, 3, 2)$	$M(7, 4) = \mathbf{3}$
12	$(21, 12; 7, 4) : (3, 3, 3, 3, 3, 3, 3)$	$M(7, 4) = \mathbf{3}$
13	$(21, 13; 3, 2) : (8, 7, 6)$	$M(3, 2) = \mathbf{2}$
14	$(21, 14; 3, 2) : (7, 7, 7)$	$M(3, 2) = \mathbf{2}$

TABLE 5.19: $n = 22$

d	$(22, d; N, D)$ -partition	$M(N, D)$
2	$(22, 2; 11, 1) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 1) = \mathbf{11}$
3	$(22, 3; 11, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 2) = \mathbf{10}$
4	$(22, 4; 11, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 2) = \mathbf{10}$
5	$(22, 5; 11, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 3) = \mathbf{7}$
6	$(22, 6; 11, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 3) = \mathbf{7}$
7	$(22, 7; 11, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 4) = \mathbf{6}$
8	$(22, 8; 11, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 4) = \mathbf{6}$
9	$(22, 9; 7, 3) : (4, 3, 3, 3, 3, 3, 3)$	$M(7, 3) = \mathbf{4}$
10	$(22, 10; 11, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 5) = \mathbf{4}$
11	$(22, 11; 7, 4) : (4, 4, 3, 3, 3, 3, 2)$	$M(7, 4) = \mathbf{3}$
12	$(22, 12; 11, 6) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 6) = \mathbf{3}$
13	$(22, 13; 3, 2) : (9, 7, 6)$	$M(3, 2) = \mathbf{2}$
14	$(22, 14; 3, 2) : (8, 7, 7)$	$M(3, 2) = \mathbf{2}$

TABLE 5.20: $n = 23$

d	$(23, d; N, D)$ -partition	$M(N, D)$
2	$(23, 2; 11, 1) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 1) = \mathbf{11}$
3	$(23, 3; 11, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 2) = \mathbf{10}$
4	$(23, 4; 11, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 2) = \mathbf{10}$
5	$(23, 5; 8, 2) : (3, 3, 3, 3, 3, 3, 3, 2)$	$M(8, 2) = \mathbf{7}$
6	$(23, 6; 11, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 3) = \mathbf{7}$
7	$(23, 7; 11, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 4) = \mathbf{6}$
8	$(23, 8; 11, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 4) = \mathbf{6}$
9	$(23, 9; 7, 3) : (4, 4, 3, 3, 3, 3, 3)$	$M(7, 3) = \mathbf{4}$
10	$(23, 10; 11, 5) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 5) = \mathbf{4}$
11	$(23, 11; 8, 4) : (3, 3, 3, 3, 3, 3, 3, 2)$	$M(8, 4) = \mathbf{4}$
12	$(23, 12; 11, 6) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(11, 6) = \mathbf{3}$
13	$(23, 13; 5, 3) : (5, 5, 5, 5, 3)$	$M(5, 3) = \mathbf{2}$
14	$(23, 14; 3, 2) : (9, 7, 7)$	$M(3, 2) = \mathbf{2}$
15	$(23, 15; 3, 2) : (8, 8, 7)$	$M(3, 2) = \mathbf{2}$

TABLE 5.21: $n = 24$

d	$(24, d; N, D)$ -partition	$M(N, D)$
2	$(24, 2; 12, 1) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 1) = \mathbf{12}$
3	$(24, 3; 12, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 2) = \mathbf{11}$
4	$(24, 4; 12, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 2) = \mathbf{11}$
5	$(24, 5; 12, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 3) = \mathbf{8}$
6	$(24, 6; 12, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 3) = \mathbf{8}$
7	$(24, 7; 12, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 4) = \mathbf{7}$
8	$(24, 8; 12, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 4) = \mathbf{7}$
9	$(24, 9; 8, 3) : (3, 3, 3, 3, 3, 3, 3, 3)$	$M(8, 3) = \mathbf{4}$
10	$(24, 10; 12, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 5) = \mathbf{4}$
11	$(24, 11; 8, 4) : (4, 3, 3, 3, 3, 3, 3, 2)$	$M(8, 4) = \mathbf{4}$
12	$(24, 12; 12, 6) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 6) = \mathbf{4}$
13	$(24, 13; 3, 2) : (11, 7, 6)$	$M(3, 2) = \mathbf{2}$
14	$(24, 14; 3, 2) : (10, 7, 7)$	$M(3, 2) = \mathbf{2}$
15	$(24, 15; 3, 2) : (9, 8, 7)$	$M(3, 2) = \mathbf{2}$
16	$(24, 16; 3, 2) : (8, 8, 8)$	$M(3, 2) = \mathbf{2}$

TABLE 5.22: $n = 25$

d	$(25, d; N, D)$ -partition	$M(N, D)$
2	$(25, 2; 12, 1) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 1) = \mathbf{12}$
3	$(25, 3; 12, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 2) = \mathbf{11}$
4	$(25, 4; 12, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 2) = \mathbf{11}$
5	$(25, 5; 12, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 3) = \mathbf{8}$
6	$(25, 6; 12, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 3) = \mathbf{8}$
7	$(25, 7; 12, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 4) = \mathbf{7}$
8	$(25, 8; 12, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(12, 4) = \mathbf{7}$
9	$(25, 9; 8, 3) : (4, 3, 3, 3, 3, 3, 3, 3)$	$M(8, 3) = \mathbf{4}$
10	$(25, 10; 5, 2) : (5, 5, 5, 5, 5)$	$M(5, 2) = \mathbf{4}$
11	$(25, 11; 8, 4) : (4, 4, 3, 3, 3, 3, 3, 2)$	$M(8, 4) = \mathbf{4}$
12	$(25, 12; 8, 4) : (4, 3, 3, 3, 3, 3, 3, 3)$	$M(8, 4) = \mathbf{4}$
13	$(25, 13; 7, 4) : (4, 4, 4, 4, 3, 3, 3)$	$M(7, 4) = \mathbf{3}$
14	$(25, 14; 3, 2) : (11, 7, 7)$	$M(3, 2) = \mathbf{2}$
15	$(25, 15; 3, 2) : (10, 8, 7)$	$M(3, 2) = \mathbf{2}$
16	$(25, 16; 3, 2) : (9, 8, 8)$	$M(3, 2) = \mathbf{2}$

TABLE 5.23: $n = 26$

d	$(26, d; N, D)$ -partition	$M(N, D)$
2	$(26, 2; 13, 1) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 1) = \mathbf{13}$
3	$(26, 3; 13, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 2) = \mathbf{12}$
4	$(26, 4; 13, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 2) = \mathbf{12}$
5	$(26, 5; 13, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 3) = \mathbf{9}$
6	$(26, 6; 13, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 3) = \mathbf{9}$
7	$(26, 7; 13, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 4) = \mathbf{8}$
8	$(26, 8; 13, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 4) = \mathbf{8}$
9	$(26, 9; 13, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 5) = \mathbf{5}$
10	$(26, 10; 13, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 5) = \mathbf{5}$
11	$(26, 11; 9, 4) : (3, 3, 3, 3, 3, 3, 3, 3, 2)$	$M(9, 4) = \mathbf{4}$
12	$(26, 12; 8, 4) : (4, 4, 3, 3, 3, 3, 3, 3)$	$M(8, 4) = \mathbf{4}$
13	$(26, 13; 7, 4) : (5, 4, 4, 4, 3, 3, 3)$	$M(7, 4) = \mathbf{3}$
14	$(26, 14; 7, 4) : (4, 4, 4, 4, 4, 3, 3)$	$M(7, 4) = \mathbf{3}$
15	$(26, 15; 3, 2) : (11, 8, 7)$	$M(3, 2) = \mathbf{2}$
16	$(26, 16; 3, 2) : (10, 8, 8)$	$M(3, 2) = \mathbf{2}$
17	$(26, 17; 3, 2) : (9, 9, 8)$	$M(3, 2) = \mathbf{2}$

TABLE 5.24: $n = 27$

d	$(27, d; N, D)$ -partition	$M(N, D)$
2	$(27, 2; 13, 1) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 1) = \mathbf{13}$
3	$(27, 3; 13, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 2) = \mathbf{12}$
4	$(27, 4; 13, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 2) = \mathbf{12}$
5	$(27, 5; 13, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 3) = \mathbf{9}$
6	$(27, 6; 13, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 3) = \mathbf{9}$
7	$(27, 7; 13, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 4) = \mathbf{8}$
8	$(27, 8; 13, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 4) = \mathbf{8}$
9	$(27, 9; 13, 5) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 5) = \mathbf{5}$
10	$(27, 10; 13, 5) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(13, 5) = \mathbf{5}$
11	$(27, 11; 7, 3) : (4, 4, 4, 4, 4, 4, 3)$	$M(7, 3) = \mathbf{4}$
12	$(27, 12; 9, 4) : (3, 3, 3, 3, 3, 3, 3, 3, 3)$	$M(9, 4) = \mathbf{4}$
13	$(27, 13; 7, 4) : (5, 5, 4, 4, 3, 3, 3)$	$M(7, 4) = \mathbf{3}$
14	$(27, 14; 7, 4) : (5, 4, 4, 4, 4, 3, 3)$	$M(7, 4) = \mathbf{3}$
15	$(27, 15; 7, 4) : (4, 4, 4, 4, 4, 4, 3)$	$M(7, 4) = \mathbf{3}$
16	$(27, 16; 3, 2) : (11, 8, 8)$	$M(3, 2) = \mathbf{2}$
17	$(27, 17; 3, 2) : (10, 9, 8)$	$M(3, 2) = \mathbf{2}$
18	$(27, 18; 3, 2) : (9, 9, 9)$	$M(3, 2) = \mathbf{2}$

TABLE 5.25: $n = 28$

d	$(28, d; N, D)$ -partition	$M(N, D)$
2	$(28, 2; 14, 1) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 1) = \mathbf{14}$
3	$(28, 3; 14, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 2) = \mathbf{13}$
4	$(28, 4; 14, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 2) = \mathbf{13}$
5	$(28, 5; 14, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 3) = \mathbf{10}$
6	$(28, 6; 14, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 3) = \mathbf{10}$
7	$(28, 7; 14, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 4) = \mathbf{9}$
8	$(28, 8; 14, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 4) = \mathbf{9}$
9	$(28, 9; 14, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 5) = \mathbf{6}$
10	$(28, 10; 14, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 5) = \mathbf{6}$
11	$(28, 11; 14, 6) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 6) = \mathbf{5}$
12	$(28, 12; 14, 6) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 6) = \mathbf{5}$
13	$(28, 13; 14, 7) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 7) = \mathbf{4}$
14	$(28, 14; 14, 7) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(14, 7) = \mathbf{4}$
15	$(28, 15; 7, 4) : (5, 4, 4, 4, 4, 4, 3)$	$M(7, 4) = \mathbf{3}$
16	$(28, 16; 7, 4) : (4, 4, 4, 4, 4, 4, 4)$	$M(7, 4) = \mathbf{3}$
17	$(28, 17; 3, 2) : (11, 9, 8)$	$M(3, 2) = \mathbf{2}$
18	$(28, 18; 3, 2) : (10, 9, 9)$	$M(3, 2) = \mathbf{2}$

TABLE 5.28: $n = 31$

d	$(31, d; N, D)$ -partition	$M(N, D)$
2	$(31, 2; 15, 1) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 1) = \mathbf{15}$
3	$(31, 3; 15, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 2) = \mathbf{14}$
4	$(31, 4; 15, 2) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 2) = \mathbf{14}$
5	$(31, 5; 15, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 3) = \mathbf{11}$
6	$(31, 6; 15, 3) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 3) = \mathbf{11}$
7	$(31, 7; 15, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 4) = \mathbf{10}$
8	$(31, 8; 15, 4) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 4) = \mathbf{10}$
9	$(31, 9; 15, 5) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 5) = \mathbf{7}$
10	$(31, 10; 15, 5) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 5) = \mathbf{7}$
11	$(31, 11; 15, 6) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 6) = \mathbf{6}$
12	$(31, 12; 15, 6) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 6) = \mathbf{6}$
13	$(31, 13; 15, 7) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 7) = \mathbf{5}$
14	$(31, 14; 15, 7) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 7) = \mathbf{5}$
15	$(31, 15; 15, 8) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 8) = \mathbf{4}$
16	$(31, 16; 15, 8) : (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 8) = \mathbf{4}$
17	$(31, 17; 3, 2) : (14, 9, 8)$	$M(3, 2) = \mathbf{2}$
18	$(31, 18; 3, 2) : (13, 9, 9)$	$M(3, 2) = \mathbf{2}$
19	$(31, 19; 3, 2) : (12, 10, 9)$	$M(3, 2) = \mathbf{2}$
20	$(31, 20; 3, 2) : (11, 10, 10)$	$M(3, 2) = \mathbf{2}$

TABLE 5.29: $n = 32$

d	$(32, d; N, D)$ -partition	$M(N, D)$
2	$(32, 2; 16, 1) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 1) = \mathbf{16}$
3	$(32, 3; 16, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 2) = \mathbf{14}$
4	$(32, 4; 16, 2) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 2) = \mathbf{14}$
5	$(32, 5; 16, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 3) = \mathbf{11}$
6	$(32, 6; 16, 3) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 3) = \mathbf{11}$
7	$(32, 7; 16, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 4) = \mathbf{10}$
8	$(32, 8; 16, 4) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 4) = \mathbf{10}$
9	$(32, 9; 16, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 5) = \mathbf{7}$
10	$(32, 10; 16, 5) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 5) = \mathbf{7}$
11	$(32, 11; 16, 6) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 6) = \mathbf{6}$
12	$(32, 12; 16, 6) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 6) = \mathbf{6}$
13	$(32, 13; 16, 7) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 7) = \mathbf{5}$
14	$(32, 14; 16, 7) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(16, 7) = \mathbf{5}$
15	$(32, 15; 15, 8) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 8) = \mathbf{5}$
16	$(32, 16; 15, 8) : (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$	$M(15, 8) = \mathbf{5}$
17	$(32, 17; 7, 4) : (5, 5, 5, 5, 4, 4, 4)$	$M(7, 4) = \mathbf{3}$
18	$(32, 18; 3, 2) : (14, 9, 9)$	$M(3, 2) = \mathbf{2}$
19	$(32, 19; 3, 2) : (13, 10, 9)$	$M(3, 2) = \mathbf{2}$
20	$(32, 20; 3, 2) : (12, 10, 10)$	$M(3, 2) = \mathbf{2}$
21	$(32, 21; 3, 2) : (11, 11, 10)$	$M(3, 2) = \mathbf{2}$

We will show and compare the values for $M^{(r=1)}(n, d)$ for $2 \leq d \leq n \leq 32$ in the following two tables. Table 5.30 shows all the upper bound values for $M^{(r=1)}(n, d)$, the maximum dimension of a linear LRC $^{(r=1)}(n, d)$, for $2 \leq d \leq n \leq 32$ derived using the 3 upper bounds, (4.3) in Corollary 4.2.1, (4.4) in Theorem 4.2.2 and the Griesmer Bound (2.6).

n\d	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
2	1																														
3	1	1																													
4	2	1	1																												
5	2	2	1	1																											
6	3	2	2	1	1																										
7	3	3	2	1	1	1																									
8	4	3	3	2	1	1	1																								
9	4	4	3	2	2	1	1	1																							
10	5	4	4	3	2	1	1	1	1																						
11	5	5	4	3	3	2	1	1	1	1																					
12	6	5	5	4	3	2	2	1	1	1	1																				
13	6	6	5	4	4	3	2	1	1	1	1	1																			
14	7	6	6	5	4	3	3	2	1	1	1	1	1																		
15	7	7	6	5	5	4	3	2	2	1	1	1	1	1																	
16	8	7	7	6	5	4	4	3	2	1	1	1	1	1	1																
17	8	8	7	6	6	5	4	3	3	2	1	1	1	1	1	1															
18	9	8	8	7	6	5	5	3	3	2	2	1	1	1	1	1	1														
19	9	9	8	7	7	6	5	4	3	3	2	1	1	1	1	1	1	1													
20	10	9	9	8	7	6	6	4	4	3	3	2	1	1	1	1	1	1	1												
21	10	10	9	8	8	7	6	5	4	3	3	2	2	1	1	1	1	1	1	1											
22	11	10	10	9	8	7	7	5	5	4	3	3	2	1	1	1	1	1	1	1	1										
23	11	11	10	9	9	8	7	6	5	4	4	3	3	2	1	1	1	1	1	1	1	1									
24	12	11	11	10	9	8	8	6	6	5	4	3	3	2	2	1	1	1	1	1	1	1	1								
25	12	12	11	10	10	9	8	7	6	5	5	3	3	3	2	1	1	1	1	1	1	1	1	1							
26	13	12	12	11	10	9	9	7	7	6	5	4	3	3	3	2	1	1	1	1	1	1	1	1	1						
27	13	13	12	11	11	10	9	8	7	6	6	4	4	3	3	2	2	1	1	1	1	1	1	1	1	1					
28	14	13	13	12	11	10	10	8	8	7	6	5	4	3	3	3	2	1	1	1	1	1	1	1	1	1	1				
29	14	14	13	12	12	11	10	9	8	7	7	5	5	4	3	3	3	2	1	1	1	1	1	1	1	1	1	1			
30	15	14	14	13	12	11	11	9	9	8	7	6	5	4	4	3	3	2	2	1	1	1	1	1	1	1	1	1	1		
31	15	15	14	13	13	12	11	10	9	8	8	6	6	5	4	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	
32	16	15	15	14	13	12	12	10	10	9	8	7	6	5	5	3	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1

TABLE 5.30: Upper Bound values for $M^{(r=1)}(n, d)$ for $2 \leq d \leq n \leq 32$

Table 5.31 summarises all the lower bound values for $M^{(r=1)}(n, d)$, the maximum dimension of a linear LRC $^{(r=1)}(n, d)$, for $2 \leq d \leq n \leq 32$ found by the code construction using $(n, d; N, D)$ -partitions and the lower bound (5.2).

n\d	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
2	1																														
3	1	1																													
4	2	1	1																												
5	2	1	1	1																											
6	3	2	2	1	1																										
7	3	2	2	1	1	1																									
8	4	3	3	2	1	1	1																								
9	4	3	3	2	2	1	1	1																							
10	5	4	4	2	2	1	1	1	1																						
11	5	4	4	3	2	2	1	1	1	1																					
12	6	5	5	3	3	2	2	1	1	1	1																				
13	6	5	5	3	3	2	2	1	1	1	1	1																			
14	7	6	6	4	4	3	3	2	1	1	1	1	1																		
15	7	6	6	4	4	3	3	2	2	1	1	1	1	1																	
16	8	7	7	4	4	4	4	2	2	1	1	1	1	1	1																
17	8	7	7	5	4	4	4	2	2	2	1	1	1	1	1	1															
18	9	8	8	5	5	4	4	3	2	2	2	1	1	1	1	1	1														
19	9	8	8	5	5	4	4	3	2	2	2	1	1	1	1	1	1	1													
20	10	9	9	6	6	5	5	3	3	3	2	2	1	1	1	1	1	1	1												
21	10	9	9	6	6	5	5	4	3	3	3	2	2	1	1	1	1	1	1	1											
22	11	10	10	7	7	6	6	4	4	4	3	3	2	2	1	1	1	1	1	1	1										
23	11	10	10	7	7	6	6	4	4	4	3	3	2	2	2	1	1	1	1	1	1	1									
24	12	11	11	8	8	7	7	4	4	4	4	2	2	2	2	1	1	1	1	1	1	1	1								
25	12	11	11	8	8	7	7	4	4	4	4	2	2	2	2	1	1	1	1	1	1	1	1	1							
26	13	12	12	9	9	8	8	5	5	4	4	3	3	2	2	2	1	1	1	1	1	1	1	1	1						
27	13	12	12	9	9	8	8	5	5	4	4	3	3	3	2	2	2	1	1	1	1	1	1	1	1	1					
28	14	13	13	10	10	9	9	6	6	5	5	4	4	3	3	2	2	2	1	1	1	1	1	1	1	1	1				
29	14	13	13	10	10	9	9	6	6	5	5	4	4	3	3	2	2	2	1	1	1	1	1	1	1	1	1	1			
30	15	14	14	11	11	10	10	7	7	6	6	5	5	4	4	2	2	2	2	1	1	1	1	1	1	1	1	1	1		
31	15	14	14	11	11	10	10	7	7	6	6	5	5	4	4	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	
32	16	15	15	11	11	11	11	8	8	7	7	5	5	5	5	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1

TABLE 5.31: Lower Bound values for $M^{(r=1)}(n, d)$ for $2 \leq d \leq n \leq 32$

Table 5.31 is split into 5 different coloured groups namely cyan, yellow, green, blue and red. All values that appear in the coloured cells other than the red cells in Table 5.31 represent the optimal dimensions for the respective (n, d) for a linear $\text{LRC}^{(r=1)}(n, k, d)$.

The optimality of the dimensions for (n, d) values that appear in the cyan cells was proven in Theorem 4.3.1. Those that appear in the yellow cells were proven in Theorems 4.3.2, 4.3.3 and 4.3.4. Those that appear in the green cells were proven in Theorems 4.3.5, 4.3.6 and 4.3.7. These were all shown in Section 4.3.

The optimal dimensions for (n, d) values that appear in the blue cells are those that attained the upper bound values in Table 5.30 and those that appear in the red cells did not attained this upper bound.

From Table 5.31, we see that a number of optimal linear $\text{LRC}^{(r=1)}(n, k, d)$ found using the $(n, d; N, D)$ -partitions code construction have either odd length n or odd minimum distance d . This is an improvement to the consideration of other publications that the optimal linear $\text{LRC}^{(r=1)}(n, k, d)$ are constructed from MDS codes which only considers even length n and even minimum distance d .

We shall end this section with a few observation from this study on linear $\text{LRC}^{(r=1)}(n, k, d)$.

We observed that the maximum dimension $M(N, D)$ of a linear $\text{LRC}^{(r=1)}(n, k, d)$ found using the lower bound (5.2) always occur when the parts of the $(n, d; N, D)$ -partition are evenly distributed. Hence, we come up with the following conjecture.

Conjecture 5.3.2. *Given n and d , if a $(n, d; N_1, D_1)$ -partition have $|p_i - p_j| \leq 1$ for all $i \neq j, i, j \in [N_1]$ i.e. the partitions of n are evenly distributed, and a $(n, d; N_2, D_2)$ -partition have $|p_i - p_j| > 1$ for some $i \neq j, i, j \in [N_2]$, then $M(N_1, D_1) \geq M(N_2, D_2)$.*

Throughout the study for the optimality of linear $\text{LRC}^{(r=1)}(n, k, d)$, we tested for the maximum dimension on the parameters for a few linear $\text{LRC}^{(r=1)}(n, k, d)$ by using a numerical brute force search but we could not find any improvements to the lower bound (5.2). Therefore, we also come up with the following conjecture.

Conjecture 5.3.3. *Given n and d , the optimal dimension of a linear $\text{LRC}^{(r=1)}(n, d)$ is given as:*

$$M^{(r=1)}(n, d) = \max_{(N, D) \in P} M(N, D), \quad (5.3)$$

where $P = \{(N, D) : \text{all } (n, d; N, D)\text{-partitions}\}$.

If the above conjecture is true, then there is a strong link between the optimality of linear codes and linear LRCs that we can investigate further since the optimal of linear codes is still a big open problem in coding theory.

5.4 Properties and Bounds of Linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$

We will end off this chapter with the last section on the properties and bounds of a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$. Many of the properties and bounds on linear $\text{LRC}_q^{(r=1)}(n, k, d)$ can be modified to apply for a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$. We want to highlight Subsection 5.4.3 on the code construction using partitions for a linear $\text{LRC}_q^{(r=1,t)}(n, d)$ which is an extension of Section 5.2 to find lower bounds for $M_q^{(r=1,t)}(n, d)$.

5.4.1 Properties of Linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$

Lemma 5.4.1. *Let $\mathcal{C}$ be a q -ary $[n, k, d]$ -linear code. $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$ if and only if the symbols for all the coordinates of any codeword $\mathbf{x} \in \mathcal{C}$ are replicated at least t times, i.e. the element of the recovering set of a symbol at a coordinate must be the same symbol from t other coordinates for all codewords.*

Proof. Proof is the same as the proof in Lemma 4.1.1 with the exception that each symbol is replicated at least t times. $\square$

Corollary 5.4.2. *Let G be a $k \times n$ generator matrix of a q -ary $[n, k, d]$ -linear code $\mathcal{C}$. $\mathcal{C}$ is a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$ if and only if the columns of G are replicated at least t times.*

Proof. Proof is the same as Corollary 4.1.4 with the exception that each symbol is replicated at least t times. $\square$

Remark 5.4.3. A $\text{LRC}_q^{(r=1,t)}(n, k, d - s)$ for $1 \leq s \leq d - 1$ has any pair of its codewords differ in at least $d - s$ places, hence any $\text{LRC}_q^{(r=1,t)}(n, k, d)$ is also a $\text{LRC}_q^{(r=1,t)}(n, k, d - s)$ for $1 \leq s \leq d - 1$.

5.4.2 Upper Bounds on Linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$

From Theorems 3.2.4 and 3.2.5 shown in Section 3.2, we can derive the upper bounds for any linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$.

Corollary 5.4.4. *For any linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$, the minimum distance d satisfies*

$$d \leq n - (t + 1)(k - 1), \quad (5.4)$$

and the dimension k is bounded above as follows:

$$k \leq \left\lfloor \frac{n-d}{t+1} \right\rfloor + 1. \quad (5.5)$$

Proof. For a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$ where $r = 1$, the upper bound (3.6) and (3.8) each reduces to (5.4). By making the dimension k (an integer) a subject, we can rewrite (5.4) to obtain (5.5). $\square$

5.4.3 Code Construction using Partitions for Linear $\text{LRC}_q^{(r=1,t)}(n, d)$

Definition 5.4.5. Given, n, d and t , define a vector $(p_1, p_2, \dots, p_N)_t$ to be a $(n, d, t; N, D)$ -partition, if

$$\begin{aligned} (i) \quad & \sum_{i=1}^N p_i = n; \\ (ii) \quad & p_i \geq t+1, \quad \text{for } 1 \leq i \leq N; \\ (iii) \quad & \min_{I \in \binom{[N]}{D}} \sum_{i \in I} p_i \geq d \quad \text{where } D \leq N. \end{aligned}$$

Note that this is a generalization of Definition 5.2.1 for arbitrary t .

Remark 5.4.6. A $(n, d, t; N, D)$ -partition must have $N \leq \lfloor \frac{n}{t+1} \rfloor$.

Lemma 5.4.7. Given n, d and t , for any $(n, d, t; N, D)$ -partition $(p_1, p_2, \dots, p_N)_t$, we can use a $[N, k, D]_q$ -linear code to construct a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$.

Proof. Similar to the proof for Lemma 5.2.3, with the exception that for every i^{th} partition, instead of having each partition where $p_i \geq 2$, we have $p_i \geq t+1$, hence there are at least $t+1$ symbols of the coordinates of the $[n, k, d]_q$ -linear code $\mathcal{C}$ that are the same. Therefore, by Lemma 5.4.1, $\mathcal{C}$ satisfies the property of locality $r = 1$ and availability t and is a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$. $\square$

Theorem 5.4.8. Given n, d and t , a lower bound by considering partitions of a $\text{LRC}_q^{(r=1,t)}(n, d)$ is given as:

$$M_q^{(r=1,t)}(n, d) \geq \max_{(N, D) \in P} M_q(N, D), \quad (5.6)$$

where $P = \{(N, D) : \text{all } (n, d, t; N, D)\text{-partitions}\}$.

Proof. Similar proof as that of Theorem 5.2.4. $\square$

Chapter 6

Conclusion

This thesis summarised and analysed the properties and bounds related to linear locally recoverable codes with locality $r = 1$ and availability t . We introduced some propagation techniques for LRCs that were derived from propagation rules of linear codes. Optimal codes for certain parameters of n and d were proven and a linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$ code construction method was presented using the concept of partitions for the length n . This construction method provided us with a good lower bound and more optimal codes were found for the binary linear locally recoverable codes.

We also provided some insights to the optimality of binary linear locally recoverable codes and further work on proving Conjectures 5.3.2 and 5.3.3 can be done to improve the tightness of bound for binary linear locally recoverable codes with locality $r = 1$. More work can also be done to explore the tightness of bound for linear $\text{LRC}_q^{(r=1,t)}(n, k, d)$ since it was only briefly addressed in this thesis.

Finding the optimality of linear codes have been a long standing open problem in coding theory. Through our code construction techniques using partitions, there seems to be a link between optimal linear codes and optimal linear locally recoverable codes. Hence, any future discovery on one type of code can also help shed light and improve the optimality of the other.

Bibliography

- [1] V. Cadambe and A. Mazumdar. “An Upper Bound on the Size of Locally Recoverable Codes”. In: *2013 International Symposium on Network Coding (NetCod)*. June 2013, pp. 1–5.
- [2] A. G. Dimakis et al. “Locality and Availability in Distributed Storage”. In: *2014 IEEE International Symposium on Information Theory (ISIT)*. 2014, pp. 681–685.
- [3] P. Gopalan et al. “Explicit Maximally Recoverable Codes With Locality”. In: *IEEE Transactions on Information Theory* 60.9 (Sept. 2014), pp. 5245–5256.
- [4] P. Gopalan et al. “On the Locality of Codeword Symbols”. In: *IEEE Transactions on Information Theory* 58.11 (Nov. 2012), pp. 6925–6934.
- [5] C. Huffman and V. Pless. *Fundamentals of Error-Correcting Codes*. Cambridge University Press, 2003.
- [6] S. Ling and C. Xing. *Coding Theory: A First Course*. Cambridge University Press, 2004.
- [7] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North-Holland Publishing Co., Amsterdam, 1977.
- [8] V. Lalitha N. Prakash and P. V. Kumar. “Codes with Locality for Two Erasures”. In: *2014 IEEE International Symposium on Information Theory (ISIT)*. June 2014, pp. 1962–1966.
- [9] D. S. Papailiopoulos and A. G. Dimakis. “Locally Repairable Codes”. In: *IEEE Transactions on Information Theory* 60.10 (2014), pp. 5843–5855.
- [10] I. Tamo and A. Barg. “A Family of Optimal Locally Recoverable Codes”. In: *2014 IEEE International Symposium on Information Theory (ISIT)*. June 2014, pp. 686–690.
- [11] I. Tamo and A. Barg. “Bounds on Locally Recoverable Codes with Multiple Recovering Sets”. In: *2014 IEEE International Symposium on Information Theory (ISIT)*. 2014, pp. 691–695.
- [12] I. Tamo, A. Barg, and A. Frolov. “Bounds on the Parameters of Locally Recoverable Codes”. In: *IEEE Transactions on Information Theory* pp.99 (2016), pp. 1–1.

- [13] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis. “Optimal Locally Repairable Codes and Connections to Matroid Theory”. In: *2013 IEEE International Symposium on Information Theory (ISIT)*. July 2013, pp. 1814–1818.
- [14] A. Wang and Z. Zhang. “Repair Locality With Multiple Erasure Tolerance”. In: *IEEE Transactions on Information Theory* 60.11 (2014), pp. 6979–6987.
- [15] H. Weatherspoon and J. Kubiatowicz. “Erasure Coding vs. Replication: A Quantitative Comparison”. In: *Proc. 1st Int. Workshop Peer-to-Peer Syst.* (2002), pp. 328–337.