Error Detection and Correction for Symmetric and Asymmetric Channels

Irina Petkova Naydenova

Selmer Center
Department of Informatics
University of Bergen, Norway

Bergen 2007

To Ivan

Contents

Acknowledgements

In the process of preparing this thesis I have enjoyed the help, support and encouragement of many people. Thank you to all of you!

First of all, I would like to express my sincere gratitude to my supervisor Professor Torleiv Kløve for sharing his vast experience and knowledge, for putting so much trust in me, for his helpful discussions, constructive ideas and suggesting interesting problems.

I would like to thank Professor Stefan Dodunekov for his help with my professional orientation and for his encouragement.

I am very grateful to Tsonka Baicheva for introducing me in Coding Theory, for her help, support, patience and for the benefits of our collaboration.

I thank to all my colleagues and friends in the Selmer Center, who have provided a very good environment to work in. The high level of the research done in the Selmer Center is a good inspiration and it is a good place to learn from others. I thank to the entire Department of Informatics for the friendly and worm atmosphere that made things so much easier.

I am sincerely grateful to my family for always showing confidence in me and especially to my husband Ivan Naydenov for his unconditional love, support and patience when I was explaining to him all the problems which I worked on!

Abstract

When a message is sent over a channel errors can occur due to noise during the transmission. So it is very important to know the error correction and detection capability of the code, which is used to encode the message.

There are different types of channels in terms of the memory. In this work we consider channels without memory. Depending on the symbol probability, the channels are symmetric or non-symmetric. The properties of the both types of channels are discussed. For symmetric channels we present codes which are good and proper for error detection. Necessary conditions for codes to be good for error detection are derived and it is shown that large codes are proper for error detection.

For the non-symmetric channels some optimal binary and ternary codes which can correct up to t errors and detect all unidirectional errors are constructed. The method includes a generalized lower bound for the length of a code over arbitrary alphabet size. First we make this generalization and then we give a method which is used for the construction.

A generalization of the Bose-Lin codes, a class of codes detecting asymmetric errors is made and the minimum weight of an undetected error is determined.

Chapter 1

Introduction

In recent years, there has been an increasing demand for efficient and reliable digital data transmission and storage systems. The demand has been accelerated by the emergence of large-scale, high-speed data networks for the exchange, processing and storage of digital information in many spheres. To design such systems a merging of communications and computer technology is required. A major concern of the system designer is the control of errors so that the data can be reliably reproduced. There are many books and papers devoted to the error-free data transmission. This introduction follows parts of [19], [23], [3], [29], as well as an Internet encyclopedia.

1.1 Applications of symmetric and asymmetric channels

Error-control codes play a central role in communication systems, providing the ability to protect digital data against errors. The well-developed theory of error-control codes has found applications in many different areas: data storage, multiple-access communications, synchronization, intentional interference, secret sharing and authentication. A conventional error-control code is designed to provide the entire message with the best possible error protection.

Error-control codes are used to send information via a communication channel. The channel is usually modelled as a triple consisting of an input alphabet, an output alphabet and for each pair (i, o) of input and output elements, a transition probability $p(i, o)$. Semantically, the transition probability is the probability that the symbol o is received given that i was transmitted over the channel.

Error-detection codes are used in all sorts of digital communication applications to enable the receiver of a message transmitted over a noisy channel to determine whether the message has been corrupted during the transmission.

In coding theory, an idealized model of a communication channel that sends bits is called a *binary symmetric channel*. It can transmit only one of two symbols (usually called 0 and 1). (A *non-binary channel* is capable of transmitting more that two symbols). The probability of a 1 becoming a 0, and of 0 becoming a 1 are assumed to be the same. In this case we say that we have *symmetric errors*. The channel is often used by theorists because it is one of the simplest noisy channels to analyze. Many problems in communication theory can be reduced to a binary symmetric channel. On the other hand, being able to transmit effectively over binary symmetric channel can give rise to solutions also for more complicated channels.

Most classes of codes have been designed for use on *symmetric channels*. However, in certain applications, such as optical communications, the error probability from 1 to 0 is significantly higher than the error probability from 0 to 1. These applications can be modeled by an *asymmetric channel*, on which only $1 \to 0$ transitions can occur (*asymmetric errors*).

One such example in [29] is the photon communication systems, in which photons are used to transmit the information. Due to energy losses in the channel a photon may not be received. Since the number of received photons does not exceed the number of transmitted photons, the photon channel is an asymmetric channel.

Further, some other memory systems behave like an *unidirectional channel*, on which, even though both $1 \to 0$ and $0 \to 1$ errors are possible, all errors within the message are of the same type (increasing or decreasing) when sending a certain message (*unidirectional errors*).

In this work we consider both *symmetric* and *non-symmetric channels*. The codes which are used to encode the message, sent over these channels, are called respectively *symmetric* and *non-symmetric codes*. We investigate which symmetric codes are *good* and *proper* for error detection and give some necessary conditions for this. We also construct some optimal codes which can correct up to t errors and detect all unidirectional errors. We have made a generalization of the Bose-Lin codes, a class of codes detecting asymmetric errors.

1.2 Outline

Since we work with symmetric and non-symmetric channels it is needed to give some basic definitions and theorems for both channels. The next chapter is divided into two parts. In the first part we give some basic knowledge on symmetric codes, which can be linear and non-linear. Some of the properties of linear codes are properties also for the non-linear codes, but this is not always the case. This is discussed in the chapter. The second part is devoted to the non-symmetric channels. From the codes which are used on these channels we present t-EC-AUED codes and generalized Bose-Lin codes and some basic notations and definitions are given.

We focus on the error detection capability of q-ary symmetric codes. For the error detection, codes can be good and proper for error detection. In Chapter 3 necessary conditions for codes to be good for error detection are presented. Usually to determine whether a code is good for error detection is a very hard computational problem. Using these conditions it is sometimes much more easier to see if the code is good or not for error detection. Parts of this chapter have been presented at the following conferences:

◇ 2005 IEEE International Symposium on Information Theory, Australia, September 2005, [15],

◇ Annual Workshop of Coding Theory and Applications, Bulgaria, December 2005, [25].

It is usually very difficult to determine whether a code is proper for error detection, too. But large codes are proper. In Chapter 4 we discuss how large. The computations are for q-ary symmetric codes. The conditions which we derive when a code is proper or not depend just on the length of the code and the alphabet size. Parts of this chapter have been presented at the

◇ 2006 IEEE Information Theory Workshop, China, October 2006, [27].

In Chapter 5 we present a function $S(m, n)$, which is the number of sequences of length m and weight n. Some very important properties of this function are derived and they are later used in Chapter 7. Some of the properties of $S(m, n)$ are published in:

$\diamond$ IEEE Transactions on Information Theory, vol. 53, pp. 1188-1193, March 2007, [28].

Another type of channels are the non-symmetric channels which are presented in this work. In Chapter 6 we construct some binary and ternary codes which are able to correct up to t errors and detect all unidirectional errors. But first we give a generalized lower bound for the length of such codes and using it we construct some optimal codes. The method of construction is described and the codes are presented (as matrices). Parts of this chapter have been presented at the following conferences:

$\diamond$ Fourth International Workshop on Optimal Codes and Related Topics, Bulgaria, June 2005, [16],

$\diamond$ 2006 International Symposium on Information Theory and its Applications, Korea, October 2006, [26].

Some other class of codes which is used when we consider non-symmetric channels are the codes presented by Bose and Lin. This class of codes can detect binary asymmetric errors. Based on this class we make a generalization to arbitrary alphabet size. We call the codes generalized Bose-Lin codes. The description of these codes is given in Chapter 7. The case when we have an undetectable error is discussed and the minimum weight of an undetectable error is determined. We take a closer look at the function which presents the maximum weight of the detected errors. Parts of this chapter have been presented at the

$\diamond$ Fourth International Workshop on Optimal Codes and Related Topics, Bulgaria, June 2005, [17]

and have been published in the following journal:

$\diamond$ IEEE Transactions on Information Theory, vol. 53, pp. 1188-1193, March 2007, [28].

Finally in Chapter 8 we make a summary of what we have done and discuss some open problems.

Chapter 2

Basics on Error-Control Channels

As it was mentioned in the introduction, channels can be divided into channels without memory and channels with memory. In this thesis we work just with channels without memory. These channels can be symmetric or non-symmetric (asymmetric and unidirectional) depending on the symbol transition probability.

2.1 Symmetric channel

The main problem when a message is transmitted over a channel is that some errors may have occurred during the transmission because of noise. The basic idea is to have a redundancy which is used to detect and, for some applications, to correct the errors. First the information should be encoded, then sent through the channel and then decoded by the receiver. We first focus on the encoding. Let $F_q = \{0, 1, ..., q-1\}$. The message will be encoded into vectors of symbols from F_q. Suppose that we have a set L of M possible messages which can be sent. An $(n, M; q)$ code is a subset of the set F_q^n, containing M vectors with length n, where F_q^n is the set of all vectors of length n with symbols from F_q. An encoding is one-to-one function from L to the code. The elements of the code are called *codewords*.

2.1.1 Basic terms

Definition 2.1.1 ***The Hamming weight*** $w_H(\mathbf{x})$ *of a vector* $\mathbf{x} \in F_q^n$ *is the number of non-zero positions in* $\mathbf{x}$*, that is*

$$w_H(\mathbf{x}) = \#\{i \mid 1 \leq i \leq n \text{ and } x_i \neq 0\}.$$

Definition 2.1.2 ***The Hamming distance*** $d_H(\mathbf{x}, \mathbf{y})$ *between two vectors* $\mathbf{x}, \mathbf{y} \in F_q^n$ *is the number of the positions where they differ, that is*

$$d_H(\mathbf{x}, \mathbf{y}) = \#\{i \mid 1 \leq i \leq n \text{ and } x_i \neq y_i\}.$$

It follows that

$$d_H(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} - \mathbf{y}).$$

When a vector $\mathbf{x}$ is sent and e errors have occurred during the transmission, then the received vector $\mathbf{y}$ will differ from $\mathbf{x}$ in e positions. So, $d(\mathbf{x}, \mathbf{y}) = e$.

Let C be an $(n, M; q)$ code, then

Definition 2.1.3 ***The minimum distance*** *of a code* C *is*

$$d_{min}(C) = min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}.$$

An $(n, M; q)$ code with minimum distance d_{min} is also called an $(n, M, d_{min}; q)$ code. Very often instead of d_{min} we just write only d, so the code will be denoted as $(n, M, d; q)$ code.

Distance and weight distribution

Definition 2.1.4 *Let* C *be an* $(n, M; q)$ *code and*

$$A_i = A_i(C) = \frac{1}{M}\#\{(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C \text{ and } d_H(\mathbf{x}, \mathbf{y}) = i\}.$$

The sequence $A_0, A_1, ..., A_n$ *is called* ***the distance distribution*** *of* C *and*

$$A_C(z) = \sum_{i=0}^{n} A_i z^i$$

is ***the distance distribution function*** *of* C*.*

Note that, if the minimum distance of the code C is d_{min}, then $A_0 = 1$; $A_i = 0$, for $0 < i < d_{min}$ and $A_{d_{min}} > 0$.

Definition 2.1.5 *Let C be an $(n, M; q)$ code and denote by A_i^w the number of codewords with weight i, that is*

$$A_i^w = A_i^w(C) = \#\{\mathbf{x} \in C \mid w_H(\mathbf{x}) = i\}.$$

The sequence $A_0^w, A_1^w, ..., A_n^w$ is called ***the weight distribution*** *of C and*

$$A_C^w(z) = \sum_{i=0}^{n} A_i^w z^i$$

is ***the weight distribution function*** *of C.*

Error detection

Error detection is the ability to detect errors which are made due to noise during a transmission from the transmitter to the receiver.

Suppose that $\mathbf{x} \in C$ is sent over a noisy channel and the received vector $\mathbf{y} \in F_q^n$ differs from $\mathbf{x}$ in e positions which means that e errors have occurred $(d_H(\mathbf{x}, \mathbf{y}) = e)$. Let the minimum distance of the code C be d_{min}. This means that any two different codewords from C differ in at least d_{min} positions. So if we have $d_{min} - 1$ or fewer errors, the received vector will be not a codeword. Then the receiver detects that the received vector $\mathbf{y}$ is not a codeword and the errors are detected. Hence:

Theorem 2.1.1 *A code C with minimum distance d_{min} is capable of detecting $d_{min} - 1$ or fewer errors. [23]*

So a code with minimum distance d_{min} guarantees detection of all $d_{min} - 1$ or fewer errors. It is also capable of detecting a large fraction of d_{min} or more errors. In fact, an $(n, M; q)$ code is capable of detecting $q^n - M$ errors. From all q^n possible vectors, there are M vectors that are identical to the M codewords. We are sending a codeword $\mathbf{x}$. If we receive a vector $\mathbf{y} \in F_q^n$, not identical to a codeword, we have an error. There are exactly $q^n - M$ vectors that are not identical to the codewords of an $(n, M; q)$ code. So these $q^n - M$ errors are *detectable*. But if the vector $\mathbf{y}$ is identical to any of the codewords, but different from the sent one $\mathbf{x}$, the transmitted codeword $\mathbf{x}$ is altered into another codeword $\mathbf{y}$. In this case the decoder accepts the received vector $\mathbf{y}$ as the transmitted codeword and performs an incorrect decoding. Such an error is called an *undetectable error*. Therefore there are $M-1$ undetectable errors.

Let C be used only for error detection on a discrete memoryless channel with q inputs and q outputs. Any transmitted symbol has a probability $1-p$ of being received correctly and a probability $\frac{p}{q-1}$ of being transformed into each of the other $q-1$ symbols, where p is the transition probability of the channel. The probability that the decoder will fail to detect errors can be computed from the distance distribution of C. Let $P_{ue}(C)$ be the probability of an undetected error. Then

$$P_{ue}(C) = \sum_{i=1}^{n} A_i \left(\frac{p}{q-1}\right)^i (1-p)^{n-i}. \tag{2.1}$$

To compute the probability of undetected error is equivalent to compute the distance distribution of the code, which is known only for a few classes of codes and it is known to be a hard computational problem. An easier problem is to find some bounds on $P_{ue}(C)$. Even if the probability of undetected error is known, a criterion is needed to decide if the code is suitable for error detection. The following criteria were introduced in [21].

Definition 2.1.6 *If*

$$P_{ue}(C,p) \leq P_{ue}\left(\frac{q-1}{q}\right) = \frac{M-1}{q^n}$$

for all $p \in \left[0, \frac{q-1}{q}\right]$*, then* C *is* ***good*** *for error detection.*

Definition 2.1.7 *If* $P_{ue}(C,p)$ *is an increasing function in* $p \in \left[0, \frac{q-1}{q}\right]$*, the code is* ***proper*** *for error detection.*

Definition 2.1.8 *The code* C *is* ***bad*** *for error detection if*

$$P_{ue}(C,p) > \frac{M-1}{q^n}$$

for some $p \in \left[0, \frac{q-1}{q}\right]$*.*

Definition 2.1.9 *The code C is* ***ugly*** *for error detection if*

$$P_{ue}(C,p) \geq \frac{M}{q^n}$$

for some $p \in \left[0, \frac{q-1}{q}\right]$.

Clearly, being ugly is a stronger condition than not bad. We note that most codes are either good or ugly, but a code may be neither.

To determine whether a code is good, proper, bad or ugly for error detection, using the definitions above, it is equivalent to compute the distance distribution and the probability of undetected error, which is, as it was said before, a very hard computational problem for most of the codes. In the next chapter we derive some conditions which show if the code is good or not for error detection and the conditions depend just on the size, the minimum distance of the code and a lower bound on A_d. In Chapter 4 we discuss proper codes.

Error correction

Another question is how many errors a code is able to correct if it is used for random-error correction. Error correction has the additional feature that enables localization of the errors and correcting them.

Let C has minimum distance d_{min}, which is even or odd, so:

$$2t+1 \leq d_{min} \leq 2t+2$$

for some integer positive t.

Let $\mathbf{x}, \mathbf{z} \in C$ and $\mathbf{y} \in F_q^n$. Again $\mathbf{x}$ is the transmitted codeword and $\mathbf{y}$ is the received vector. The Hamming distance among $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ satisfies the triangle inequality:

$$d_H(\mathbf{x},\mathbf{y}) + d_H(\mathbf{y},\mathbf{z}) \geq d_H(\mathbf{x},\mathbf{z}). \tag{2.2}$$

Suppose that t' errors have occurred during the transmission, so:

$$d_H(\mathbf{x},\mathbf{y}) = t'. \tag{2.3}$$

Since the minimum distance of the code is d_{min} it follows that

$$d_H(\mathbf{x},\mathbf{z}) \geq d_{min} \geq 2t+1. \tag{2.4}$$

If $t' \leq t$, then $d(\mathbf{y}, \mathbf{z}) > t$, using (2.2), (2.3), (2.4).

This inequality says that if t or fewer errors occur during the transmission, then the received vector $\mathbf{y}$ is closer (in Hamming distance) to the sent codeword $\mathbf{x}$ than to any other codeword in C. Which means that $\mathbf{y}$ will be decoded into $\mathbf{x}$, which is the actual transmitted codeword. The decoding will be correct and the errors will be corrected.

Theorem 2.1.2 *A code with minimum distance d_{min} guarantees correction of $t = \lfloor (d_{min} - 1)/2 \rfloor$ or fewer errors. The parameter t is called the **random-error-correction capability** of the code and the code is t**-error-correcting code**. [23]*

The code is not capable of correcting all l errors, $l > t$, for there is at least one case in which l errors result in a received vector that is closer to an incorrect codeword than to the transmitted codeword. Usually a code with random-error-correcting capability t is capable of correcting q^n/M errors, including those with t or fewer errors.

2.1.2 Linear codes

The symmetric codes can be divided into two in terms of the vector set. They can be *linear* or *non-linear*. For linear codes F_q is identical to the Galois field $GF(q)$, so F_q^n is a vector space, where q is some prime power. The most important codes for a symmetric channel are the linear codes.

Definition 2.1.10 ***A linear code** is a k-dimensional subspace of the vector space F_q^n.*

The code is denoted by $[n, k; q]$, where n is the *length* of the codewords, k is the *dimension* of the code and q is the *alphabet size*. If the minimum distance of the code is d_{min}, the code can be denoted also as $[n, k, d_{min}; q]$ or $[n, k, d; q]$. If $q = 2$ we just write $[n, k]$, $[n, k, d_{min}]$ or $[n, k, d]$ code, if the minimum distance is known. For a linear code the *size* of the code is $M = q^k$.

It is obvious that for linear codes we have

$$A_i = A_i^w \text{ and } A_C(z) = A_C^w(z).$$

So for convenience, when we consider linear codes we drop w in the notations for weight distribution and weight distribution function and we can use just one of the terms: distance or weight distribution and distance or weight distribution function. So in the formula for the probability of undetected error $P_{ue}(C, p)$ instead of distance distribution we can use the weight distribution.

Generator matrix

Since a linear code C is a k-dimensional subspace of F_q^n, the vector space can be presented in terms of k vectors $\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_k$ as a basis and then C is the set of all possible linear combinations of these vectors.

Definition 2.1.11 *A* ***generator matrix*** *G of the code C is a $k \times n$ matrix whose rows $\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_k$ are a basis.*

Dual code

The space of all vectors which are orthogonal to all the codewords of the linear code C form a new linear code. Since the ground field is finite it is possible for all the codewords in a code to be orthogonal to themselves.

Definition 2.1.12 *If C is an $[n, k; q]$ code, then* **the dual code** *$C^\perp$ of C is an $[n, n-k; q]$ code defined by:*

$$C^\perp = \{\mathbf{v} \in F_q^n \mid \mathbf{v} \cdot \mathbf{x} = 0, \forall \mathbf{x} \in C\},$$

where

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + ... + x_n y_n \in F_q,$$

for all $\mathbf{x} = (x_1, x_2, ..., x_n)$ and $\mathbf{y} = (y_1, y_2, ..., y_n)$.

Parity-check matrix

Definition 2.1.13 *A* ***parity-check matrix*** *H of a code C is an $(n-k) \times n$ matrix of rank $n-k$ which is a generator matrix of the dual code.*

Note that $GH^t = 0$, where H^t is the transposed of H.

Systematic code

Definition 2.1.14 *A code C is called* ***systematic*** *if the generator matrix has the form $G = (I_k|P)$, where I_k is a $k \times k$ identity matrix and P is some $k \times (n-k)$ matrix.*

Since $(I_k|P)(-P^t|I_{n-k})^t = -P + P = 0$, $H = (-P^t|I_{n-k})$ is a parity-check matrix of C.

Equivalent codes

Definition 2.1.15 *If G is any matrix with entries in a field F_q, then replacing any row of G by (a) its sum with another row, or (b) its scalar multiple with any non-zero element of F_q, is called an* ***elementary row operation****. If G is any matrix with entries in a field F_q then (a) swapping any two columns, or (b) replacing any column of G by its scalar multiple with any non-zero element of F_q, is called a* ***simple column operation****.*

Definition 2.1.16 *If G is a generator matrix for a linear code C and C' is the linear code with generator matrix G', where G' is any matrix obtained from G by elementary row operations or simple column operations, then we say that the codes C and C' are* ***equivalent****.*

If C and C' are equivalent linear codes then $A_i(C) = A_i(C')$. For the study of the weight distribution, we may therefore, without loss of generality assume that the code is systematic.

2.1.3 Non-linear codes

Linear codes satisfy the property that any sum of codewords gives again a codeword. But not every code is linear. We have many more non-linear codes than linear ones.

When we have a linear code we have a linear vector space with a basis and to describe the code we need only k codewords, since linear combinations of these k codewords span the k-dimensional subspace. However with a non-linear code we need to list all the codewords, that is M. This is part of the motivation for working with linear codes instead of non-linear codes.

On the other hand, the main problem with linear codes is that the number of codewords for a given code length is restricted. One cannot just add a new vector which is at the appropriate distance from all the other codewords, one has also to check the sum of this vector with all codewords in the code for the minimum distance property. The best non-linear codes for a given length usually have more codewords than their linear counterparts. Thus information can be sent more quickly or stored more compactly using non-linear codes.

It turns out that many properties of non-linear codes are very similar to those of linear codes.

Let $F_q = \{0, 1, ...q - 1\}$, where q is some positive number and let M be the set of all possible messages which can be sent.

Definition 2.1.17 *A **non-linear** $(n, M; q)$ code is a subset of F_q^n, containing M vectors with length n.*

For non-linear codes the weight and the distance distribution are not necessary the same. We may have that:

$$A_i \neq A_i^w \text{ and } A_C(z) \neq A_C^w(z).$$

The necessary conditions for codes to be good for error detection, which are given in the next chapter, are valid for linear and non-linear codes.

2.2 Non-symmetric channels

As it was mentioned in the introduction there are other channels than the symmetric channels. Examples are the *asymmetric* and the *unidirectional* channels.

Let $F_q = \{0, 1, 2, ..., q - 1\}$ for $q \geq 2$.

Definition 2.2.1 *The **q-ary asymmetric channel** is the channel on which the only transitions that can occur are $x \rightarrow y$, where $0 \leq y \leq x \leq q - 1$.*

If all $y \leq x$ are possible as a received symbol, we call the channel *complete*. As an example of a noncomplete channel is the channel introduced by Ahlswede and Aydinian [2], on which when x is sent only 0 and x can be received. In this work we assume that the channel is complete, when we considering an asymmetric channel.

Definition 2.2.2 *The **q-ary unidirectional channel** is the channel on which all errors within a codeword are of the same type (all increasing or all decreasing).*

The codes, which are used to encode the message, sent over these channels, are called *q-ary asymmetric codes* and *q-ary unidirectional codes*, respectively.

Parameters of the non-symmetric code

Let C be a code over F_q^n. Let $\mathbf{x}, \mathbf{y} \in F_q^n$ and let $N(\mathbf{x}, \mathbf{y})$ denote the number of positions i where $x_i > y_i$. If $N(\mathbf{y}, \mathbf{x}) = 0$ the vector $\mathbf{x}$ is said to *cover* the vector $\mathbf{y}$ ($\mathbf{x} > \mathbf{y}$). If $\mathbf{x} \geq \mathbf{y}$ or $\mathbf{y} \geq \mathbf{x}$ the vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be *ordered*, otherwise they are *unordered*. The *Hamming distance* $d_H(\mathbf{x}, \mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$ is the sum of $N(\mathbf{x}, \mathbf{y})$ and $N(\mathbf{y}, \mathbf{x})$:

$$d_H(\mathbf{x}, \mathbf{y}) = N(\mathbf{x}, \mathbf{y}) + N(\mathbf{y}, \mathbf{x}) = \#\{i \mid x_i \neq y_i\}.$$

Equivalent codes

Two non-symmetric codes can also be *equivalent*. We say that two codes are called *equivalent* if they differ only in the order of the coordinates. Hence equivalent codes have the same parameters n, M and d_{min}.

Types of errors, error correction and error detection

There are three different types of errors: *symmetric*, *asymmetric* and *unidirectional errors*. When a codeword $\mathbf{x} \in C$ is sent and a vector $\mathbf{y} \in F_q^n$ is received, we say that $\mathbf{x}$ has suffered t *asymmetric errors* if $\mathbf{x}$ covers $\mathbf{y}$ and $d_H(\mathbf{x}, \mathbf{y}) = t$, that $\mathbf{x}$ has suffered t *unidirectional errors* if $\mathbf{x}$ covers $\mathbf{y}$ or is covered by $\mathbf{y}$ and $d_H(\mathbf{x}, \mathbf{y}) = t$, and that $\mathbf{x}$ has suffered t *symmetric errors* if $d_H(\mathbf{x}, \mathbf{y}) = t$. [29]

In [29] are constructed three kinds of spheres, with which help are explained the error correction and detection capabilities of the code. Let the spheres have a radius t and $\mathbf{x} \in F_q^n$:

$$\begin{aligned} S_{Sy}(\mathbf{x}, t) &= \{\mathbf{y} \in F_q^n | d_H(\mathbf{x}, \mathbf{y}) \leq t\}, \\ S_U(\mathbf{x}, t) &= \{\mathbf{y} \in F_q^n | d_H(\mathbf{x}, \mathbf{y}) \leq t \text{ and } (\mathbf{x} \leq \mathbf{y} \text{ or } \mathbf{y} \geq \mathbf{x})\}, \\ S_{As}(\mathbf{x}, t) &= \{\mathbf{y} \in F_q^n | d_H(\mathbf{x}, \mathbf{y}) \leq t \text{ and } \mathbf{x} \geq \mathbf{y}\}. \end{aligned}$$

Each sphere $S_X(\mathbf{x}, \mathbf{y})$ contains the vectors that can be received when a codeword $\mathbf{x}$ is sent suffering t or fewer errors of type X (X=Sy(mmetric), X=As(ymmetric) or X=U(nidirectional), respectively). Hence

Definition 2.2.3 *A code C can* ***correct*** *up to t errors of type X if the spheres $S_X(\mathbf{x}, t)$ are disjoint for any two distinct codewords. [29]*

Definition 2.2.4 *A code C can* ***detect*** *up to t errors of type X if the sphere $S_X(\mathbf{x}, t)$ does not contain codewords different from* $\mathbf{x}$ *for all* $\mathbf{x} \in C$. *[29]*

From the error detection point of view, the asymmetric and the unidirectional codes are equivalent, that is, a code capable of detecting up to t asymmetric errors is also capable of detecting t unidirectional errors.

Necessary and sufficient conditions for correcting and detecting errors of each of the three types are known [29]. However, sometimes a combination of correction and detection is required or even correction and/or detection of errors of different type. In this work in Chapter 6 we will discuss codes which are able to correct up to t symmetric errors and detect all unidirectional errors. Such a code is called a *t-EC-AUED* code.

2.2.1 t-EC-AUED codes

Binary t-EC-AUED codes, in particular for $t = 1$, have been extensively studied. We construct some optimal binary and ternary t-EC-AUED codes for $t \geq 1$.

A characterization when a code is a t-EC-AUED code is known, [9]:

Theorem 2.2.1 *A code C is a t-EC-AUED code if and only if $N(\mathbf{x}, \mathbf{y}) \geq t+1$ and $N(\mathbf{y}, \mathbf{x}) \geq t+1$, for all distinct $\mathbf{x}, \mathbf{y} \in C$.*

There are many papers describing constructions of t-EC-AUED codes (see e.g. [3]). Let $A(n, t)$ denote the maximal size of a t-EC-AUED code of length n. We give a brief survey of bounds on the number of codewords in a binary t-EC-AUED code:

◇ **Lower bounds**:

Graham and Sloan [18]: $A(n,t) \geq \dfrac{\binom{n}{\lfloor n/2 \rfloor}}{(q+1)^t}$, $t > 1$, where q is the smallest prime power not smaller than n.

Bose and Rao [9]: $A(n,1) \geq \dfrac{\binom{n}{\lfloor n/2 \rfloor}}{(n+1)}$.

◇ **Upper bounds**:

Lin and Bose [22]: $A(n,1) \leq \dfrac{2}{n}\binom{n}{\lfloor n/2 \rfloor}$.

Zhang and Xia [30]: $A(n,t) \leq \dfrac{(t+1)\binom{n}{\lceil n/2 \rceil}}{\Omega(n,t)}$, $0 \leq t \leq n$, where $\Omega(n,t) = \sum_{i=0}^{t} \sum_{j=max(0,2i-t)}^{i} \binom{n-2i}{t-2i+j}\binom{2i}{j}$

Let $n_q(a, t+1)$ denote the length of the shortest t-EC-AUED code of size a over F_q^n. We call a t-EC-AUED code of length $n_q(a, t+1)$ and size a *optimal.*

Böinck and van Tilborg gave a Plotkin type lower bound for the length of a binary t-EC-AUED code [4]:

$$n_2(a, t+1) \geq \left\lceil \left(4 - \frac{2}{\lceil a/2 \rceil}\right)(t+1) \right\rceil .$$

In Chapter 6 we make a generalization of this bound and using it and some lemmas, which we will present, we construct some optimal binary and ternary t-EC-AUED codes.

We remark that for $t = 0$ it has been shown by de Bruijn et al. [10] that for given n the largest 0-EC-AUED code of length n is the code of all $S(n, \lceil n(q-1)/2 \rceil)$ codewords of weight $\lceil n(q-1)/2 \rceil)$, where the function $S(x, y)$ is presented in Chapter 5 and denotes the number of vectors with length x and weight y. There is no simple formula for $S(n, \lceil n(q-1)/2 \rceil)$ in general, for $q = 2$ it is $S(n, \lceil n/2 \rceil) = \binom{n}{\lceil n/2 \rceil}$. For larger q the size of the codes is discused in Chapter 6.

2.2.2 Generalized Bose-Lin codes

We have an undetected error if a codeword is sent and another codeword, different from the sent one, is received. Bose and Lin [7] introduced a class of systematic binary codes for detection of asymmetric errors. The probability of undetected error for these codes was determined in [20]. The class of codes is conjectured to contain the best possible systematic binary codes for error detection, but this has not been verified.

The only known systematic codes for detecting q-ary asymmetric errors for arbitrary q are the codes introduced by Bose and Pradhan [8] and the codes introduced by Bose, Elmougy, and Tallini [5], [12]. For $q = 2$, both these classes of codes are subclasses of the Bose-Lin codes.

In Chapter 7 we introduce a more general class of q-ary systematic error detecting codes. The codes by Bose and Pradhan [8] and the codes by Bose, Elmougy, and Tallini [5], [12] are subclasses. For $q = 2$, our class of codes coincides exactly with the Bose-Lin codes. Therefore, we call our class of codes

Generalized Bose Lin Codes (GBL codes). Since the binary codes are conjectured to contain the best possible systematic codes, this may be the case also for the GBL codes.

The undetectable errors for GBL codes are characterized. The undetectable errors of minimum weight are explicitly described. Chapter 7 follows parts of the presentation in [20]. Some of the generalization is more or less straight forward, in other parts, the generalization presents new problems.

Chapter 3

Necessary Conditions for Codes to be Good for Error Detection

This and the next chapter are devoted to the symmetric channel. In this chapter we present bounds on the length of codes to be good for error detection on a q-ary symmetric channel. The purpose is to show that, for given κ and d, there exists a value $\mu(d,\kappa)$ such that if $n \geq \mu(d,\kappa)$ and C is an $(n, M, d; q)$ code such that $A_d \geq A$, where $\ln A = \ln M - \kappa$, then C is not good for error detection. Further, we give approximations of $\mu(d,\kappa)$.

3.1 Existence of $\mu(d,\kappa)$

The probability of undetected error for C on a symmetric channel with symbol error probability p is given by (see e.g. [19]) and it was presented as (2.1) in Chapter 2. Define

$$P_{\mathrm{ue}}^{\perp}(C,p) = \frac{1}{M}\sum_{i=0}^{n} A_i(1-Qp)^i - (1-p)^n, \tag{3.1}$$

where $Q = q/(q-1)$.

If C is linear,

$$P_{\mathrm{ue}}(C^{\perp},p) = P_{\mathrm{ue}}^{\perp}(C,p). \tag{3.2}$$

Recall that C is *good* if $P_{\mathrm{ue}}(C,p) \leq (M-1)q^{-n}$ for all $p \in (0,(q-1)/q)$, C is *bad* (for error detection) if $P_{\mathrm{ue}}(C,p) > (M-1)q^{-n}$ for some $p \in (0,(q-1)/q)$, and C is *ugly* if $P_{\mathrm{ue}}(C,p) \geq Mq^{-n}$ for some $p \in (0,(q-1)/q)$.

Lemma 3.1.1 *Let C be an $(n, M; q)$ code and suppose that $P_{\rm ue}^{\perp}(C, p) \geq 1/M$ for some $p \in (0, (q-1)/q)$. Define π by*

$$1 - Qp = \frac{\pi}{(q-1)(1-\pi)}.$$

Then $P_{\rm ue}(C, \pi) \geq Mq^{-n}$.

Proof: We have

$$1 - p = \frac{1}{q(1-\pi)}$$

and so

$$\begin{aligned} 0 &\leq MP_{\rm ue}^{\perp}(C,p) - 1 \\ &= \sum_{i=1}^{n} A_i (1-Qp)^i - M(1-p)^n \\ &= (1-\pi)^{-n} \left\{ \sum_{i=1}^{n} A_i \left(\frac{\pi}{q-1} \right)^i (1-\pi)^{n-i} - Mq^{-n} \right\} \\ &= (1-\pi)^{-n} \{ P_{\rm ue}(C,\pi) - Mq^{-n} \}. \end{aligned}$$

Hence $P_{\rm ue}(C, \pi) \geq Mq^{-n}$. □

Since $\pi \in (0, (q-1)/q)$, we immediately get the following corollary.

Corollary 3.1.1 *If C is an $(n, M; q)$ code and $P_{\rm ue}^{\perp}(C, p) \geq 1/M$ for some $p \in (0, (q-1)/q)$, then C is ugly.*

Corollary 3.1.2 *If C is linear, then C is ugly if and only if $C^{\perp}$ is ugly.*

Proof: The if part follows directly from (3.2) and Corollary 3.1.1. Since $C^{\perp\perp} = C$ we get the if and only if. □

Remark 3.1 *For $q = 2$, Corollary 3.1.2 is Theorem 3.4.2, part 1 in [19]. The proof for general q given above is a generalization of the proof for $q = 2$ given in [19].*

Remark 3.2 *It is not the case, for linear codes, that C bad implies that $C^{\perp}$ is bad.*

We want to find sufficient conditions for a code to be ugly. For a linear code, a general lower bound on A_d is $q-1$, and for a non-linear code a general lower bound on A_d is $2/M$. Now, let A be some positive number. We will consider $(n, M, d; q)$ codes for which $A_d \geq A$. In the rest of the chapter we also use the notation

$$\kappa = \ln(M/A) = \ln M - \ln A.$$

By definition,

$$P_{\text{ue}}^{\perp}(C,p) \geq \frac{1}{M} + \frac{A}{M}(1-Qp)^d - (1-p)^n.$$

Hence, if

$$\frac{A}{M}(1-Qp)^d \geq (1-p)^n, \tag{3.3}$$

then $P_{\text{ue}}^{\perp}(C,p) \geq \frac{1}{M}$. Taking logarithms in (3.3), we get the equivalent condition

$$-\kappa + d\ln(1-Qp) \geq n\ln(1-p).$$

Combining this with Corollary 3.1.1, we get the following lemma.

Lemma 3.1.2 *If C is an $(n, M, d; q)$ code and*

$$n \geq h(p) = \frac{d\ln(1-Qp) - \kappa}{\ln(1-p)},$$

then C is ugly.

Any choice of p, $0 < p < (q-1)/q$ now gives a proof of the existence of a $\mu(d,\kappa)$ such that if $n \geq \mu(d,\kappa)$ and C is an $(n, M, d; q)$ code with $A_d \geq A$, then C is ugly for error detection. To get the strongest result from the lemma, we want to find the p that minimizes $h(p)$. We can not find a closed formula for this, but we consider approximations.

We will use the notations

$$f(p) = \frac{\ln(1-Qp)}{\ln(1-p)}, \text{ and } g(p) = \frac{-1}{\ln(1-p)}.$$

Then

$$h(p) = d\,f(p) + \kappa\,g(p). \tag{3.4}$$

The function $f(p)$ is increasing on $(0;(q-1)/q)$, it approaches the value Q when $p \to 0+$, and it approaches infinity when $p \to (q-1)/q-$. Moreover,

$$f'(p) = \frac{-Q(1-p)\ln(1-p) + (1-Qp)\ln(1-Qp)}{(1-p)(1-Qp)\ln(1-p)^2},$$

and

$$f''(p) = \frac{f_1(p)}{-(1-p)^2(1-Qp)^2(\ln(1-p))^3},$$

where

$$\begin{aligned} f_1(p) &= Q^2(1-p)^2(\ln(1-p))^2 + 2Q(1-p)(1-Qp)\ln(1-p) \\ &\quad -2(1-Qp)^2\ln(1-Qp) - (1-Qp)^2\ln(1-p)\ln(1-Qp) > 0 \end{aligned}$$

for all $p \in (0;(q-1)/q)$. Hence f is convex on $(0;(q-1)/q)$. Similarly, the function $g(p)$ is decreasing on $(0;(q-1)/q)$, it approaches infinity when $p \to 0+$, and it takes the value $-1/\ln q$ for $p = (q-1)/q$. Moreover,

$$g'(p) = \frac{-1}{(1-p)\ln(1-p)^2} = \frac{-(1-Qp)}{(1-p)(1-Qp)\ln(1-p)^2},$$

$$g''(p) = \frac{-(2+\ln(1-p))}{(1-p)^2(\ln(1-p))^3} > 0$$

for all $p \in (0;(q-1)/q)$, and so $g(p)$ is also convex on $(0;(q-1)/q)$. This implies that the combined function $h(p)$ is also convex on $(0;(q-1)/q)$ since $\kappa > 0$, and it takes its minimum somewhere in $(0;(q-1)/q)$. We denote this minimum by $\mu(d,\kappa)$.

From Corollary 3.1.1 and Lemma 3.1.2 we get the following necessary condition for a code to be good.

Corollary 3.1.3 *If C is good for error detection, then $n < \mu(d,\kappa)$.*

We next consider $d \geq \kappa$ and $\kappa \geq d$ separately. In particular, we find approximations for $\mu(d,\kappa)$ when $d \gg k$ or $k \gg d$. We denote by p_m the value of p where $h(p)$ has its minimum; this minimum is by definition $\mu(d,\kappa)$.

3.2 On $\mu(d,\kappa)$ when $d \geq \kappa$

In this section, we let $\kappa = \alpha d$, where α is a parameter, $0 \leq \alpha \leq 1$. Then

$$h(p) = d\,\frac{\ln(1-Qp) - \alpha}{\ln(1-p)}$$

and

$$\begin{aligned}\frac{h'(p)}{d} &= \frac{-Q(1-p)\ln(1-p) + (1-Qp)\ln(1-Qp)}{(1-p)(1-Qp)\ln(1-p)^2} \\ &\quad - \frac{\alpha}{(1-p)\ln(1-p)^2}.\end{aligned}$$

In particular $h'(p) = 0$ if (and only if)

$$\alpha = \frac{-Q(1-p)\ln(1-p) + (1-Qp)\ln(1-Qp)}{1-Qp}. \tag{3.5}$$

We want to solve this for p in terms of α. There is no closed form of this solution. However, we can find good approximations. For $\alpha \to 0+$, we see that $p \to 0$ and $h(p) \to Q$. We will first study this important case in more details. We note that $\alpha \to 0+$ implies that $d \to \infty$. The parameter κ may also grow, but then at a slower rate (since $d/\kappa \to 0$).

Theorem 3.2.1 *Let*

$$y = \sqrt{\frac{\alpha}{2Q(Q-1)}}.$$

There exist numbers a_i and b_i for $i = 1, 2, \ldots$ such that, for any $r \geq 0$,

$$p_m = \sum_{i=1}^{r} a_i y^i + O(y^{r+1}),$$

and

$$\mu(d,\alpha d) = dQ\left\{1 + 2(Q-1)\sum_{i=1}^{r} b_i y^i + O(y^{r+1})\right\}$$

when $y \to 0$ (that is $\alpha \to 0$). The first few a_i and b_i are given by the following table:

$$\begin{aligned} a_1 &= 2, \\ a_2 &= -(8Q+2)/3, \\ a_3 &= (26Q^2 + 22Q - 1)/9, \end{aligned}$$

$$\begin{aligned}
a_4 &= -(368Q^3 + 708Q^2 - 12Q + 8)/135, \\
a_5 &= (1252Q^4 + 4600Q^3 + 480Q^2 + 100Q - 23)/540, \\
a_6 &= -(15424Q^5 + 98192Q^4 + 33568Q^3 + 3328Q^2 - 1600Q + 304)/8505, \\
a_7 &= (449608Q^6 + 4667496Q^5 + 3382044Q^4 + 368432Q^3 - 168006Q^2 \\
&\quad +67866Q - 11237)/340200, \\
a_8 &= -(23104Q^7 + 375664Q^6 + 483984Q^5 + 89744Q^4 - 25840Q^3 + 16272Q^2 \\
&\quad -5648Q + 832)/25515, \\
a_9 &= (43153232Q^8 + 1068845248Q^7 + 2220153152Q^6 + 728346016Q^5 \\
&\quad -119425048Q^4 + 112747312Q^3 - 61064752Q^2 + 18694504Q \\
&\quad -2482411)/73483200, \\
b_1 &= 1, \\
b_2 &= (2Q - 1)/3, \\
b_3 &= (2Q^2 - 2Q - 1)/18, \\
b_4 &= -2(Q - 2)(2Q - 1)(Q + 1)/135, \\
b_5 &= (4Q^4 - 8Q^3 - 48Q^2 + 52Q - 23)/1080, \\
b_6 &= (16Q^5 - 40Q^4 + 352Q^3 - 488Q^2 + 464Q - 152)/8505, \\
b_7 &= -(1112Q^6 - 3336Q^5 + 22116Q^4 - 38672Q^3 + 61806Q^2 \\
&\quad -43026Q + 11237)/680400, \\
b_8 &= (16Q^7 - 56Q^6 + 552Q^5 - 1240Q^4 + 3128Q^3 - 3480Q^2 \\
&\quad +1912Q - 416)/25515, \\
b_9 &= -(9136Q^8 - 36544Q^7 + 1853248Q^6 - 5431840Q^5 + 21387736Q^4 \\
&\quad -33765040Q^3 + 29284048Q^2 - 13300744Q + 2482411)/146966400.
\end{aligned}$$

Proof: First we note that $\alpha = 2Q(Q-1)y^2$ and so

$$h(p) = d\,\frac{\ln(1-Qp) - 2Q(Q-1)y^2}{\ln(1-p)},$$

and

$$h'(p) = d\,\frac{H(p,y)}{(1-p)(1-Qp)(\ln(1-p))^2},$$

where

$$H(p,y) = -Q(1-p)\ln(1-p) + (1-Qp)\ln(1-Qp) - 2Q(Q-1)y^2(1-Qp).$$

Hence $h'(p) = 0$ if $H(p,y) = 0$. Taking the Taylor expansion of $H(\sum a_i y^i, y)$ we get

$$H\left(\sum a_i y^i, y\right) = \frac{a_1^2 - 4}{4}y^2 + \frac{a_1}{6}(Qa_1^2 + a_1^2 + 6a_2 + 12Q)y^3 + \cdots$$

All coefficients for $i \leq r$ should be zero. In particular, the coefficient of y^2 shows that $a_1^2 = 4$. Since $a_1 y^2$ is the dominating term in the expression for p when y is small and $p > 0$, we must have $a_1 > 0$ and so $a_1 = 2$. Next the coefficient of y^3 shows that $a_2 = -(16Q+4)/6$. In general, we get equations in the a_i which can be used to determine the a_i recursively. The recursions seems to be quite complicated in general and we have not found an explicit general expression for a_i. Substituting the expression for p into $h(p)$ and taking Taylor expansion, we get the expression for $\mu(d,\kappa)$. □

Remark 3.3 *We do not know when the infinite series $\sum_{i=1}^{\infty} a_i y^i$ and $\sum_{i=1}^{\infty} b_i y^i$ converge (we believe that they always do, but it may depend on q).*

Assuming that $\kappa\alpha \to 0$ and taking the first three terms of approximation, we get

$$\mu(d,\kappa) \approx dQ + \sqrt{2d\kappa Q(Q-1)} + \frac{2Q-1}{3}\kappa, \tag{3.6}$$

(the other terms go to zero with y).

Because of the big O term in Theorem 3.2.1, we do not know if the approximation is smaller or larger than the exact value. Only for approximations larger than $\mu(d,\kappa)$ we can be sure that the corresponding code is ugly. We will call such approximations *upper* approximations.

By definition, $h(p)$ is an upper approximation for any p. One way to get a good upper approximation is to choose for p a good approximation for p_m. For example, taking the first term in the approximation for p_m, that is, $p = \sqrt{2\alpha/(Q(Q-1))}$, we get

$$\mu(d,\kappa) \leq h\left(\sqrt{2\alpha/(Q(Q-1))}\right). \tag{3.7}$$

Example 3.2.1 *Consider $q = 2$, $A = 1$ (valid for all linear codes), $d = 1000$ and $M = 4$. Then $\kappa = 2\ln 2$ and $\alpha \approx 0.001386$. Solving $h'(p) = 0$ numerically, we get $p \approx 0.0352540$ and $\mu(1000, 2\ln 2) \approx 2075.8565430$. Taking one, two, three, and four terms respectively in the expression for $\mu(1000, 2\ln 2)$ in Theorem 3.2.1, as well as the bound (3.7), we get the following approximations:*

no. of terms		*value*
1		2000
2		2074.4659482
3	*= (3.6)*	2075.8522426
4		2075.8565439
	(3.7)	2075.9808954

We have computed the integers $\lceil \mu(d,\kappa) \rceil$, $\lceil (3.6) \rceil$, and $\lceil (3.7) \rceil$ for a number of different values of d. When d is large then these values differ by at most one, and when d decreases the difference between the values increases slowly.

The approximation (3.6) is good for $\alpha \approx 0$. When $\alpha > 0$, the terms after the third term do not go to zero. However, the approximation in Theorem 3.2.1 may still be quite good. We have made some numeric computations that illustrate this. We use the following notations:

$$\eta(\alpha) = \frac{\mu(d, \alpha d)}{d} \text{ and } \nu_r(\alpha) = Q + 2Q(Q-1)\sum_{i=1}^{r} b_i y^i \tag{3.8}$$

From the table in Theorem 3.2.1, we see that $b_2 > 0$ for all Q (remember that $Q = q/(q-1) \in (1,2]$). Further, $b_3 > 0$ for $q = 2$ and $q = 3$, but $b_3 < 0$ for $q \geq 4$. Finally, $b_4 = 0$ for $q = 2$ and $b_4 > 0$ for $q > 2$. Hence, for all $\alpha > 0$

$$\begin{array}{ll} \text{for } q = 2, & \nu_2(\alpha) < \nu_3(\alpha) = \nu_4(\alpha), \\ \text{for } q = 3, & \nu_2(\alpha) < \nu_3(\alpha) < \nu_4(\alpha), \\ \text{for } q \geq 4, & \nu_3(\alpha) < \min\{\nu_2(\alpha), \nu_4(\alpha)\}. \end{array}$$

A simple calculation shows that for $q \geq 4$ we have $\nu_4(\alpha) > \nu_2(\alpha)$ if and only if

$$\alpha > \omega_q \overset{\text{def}}{=} \frac{135^2(2Q^2 - 2Q - 1)^2 Q(Q-1)}{2 \cdot 18^2 (Q-2)^2 (2Q-1)^2 (Q+1)^2}.$$

When q increases, the value of ω_q increases from $\omega_4 \approx 0.023$ to $\omega_9 \approx 0.378$ and then decreases and approaches zero when $q \to \infty$.

Example 3.2.2 *In this example we consider $q = 2$. For all values of α we have computed, $\nu_2(\alpha) < \eta(\alpha) < \nu_3(\alpha)$. Some examples are given in the following table.*

α	$\eta(\alpha)$	$\nu_2(\alpha) - \eta(\alpha)$	$\nu_3(\alpha) - \eta(\alpha)$
10^{-6}	2.00200100008	$-8.33 \cdot 10^{-11}$	$1.28 \cdot 10^{-17}$
10^{-3}	2.06424818804	$-2.63 \cdot 10^{-6}$	$3.95 \cdot 10^{-10}$
0.1	2.73505913571	$-2.60 \cdot 10^{-3}$	$3.16 \cdot 10^{-5}$
1	5.07687209474	$-7.68 \cdot 10^{-2}$	$6.46 \cdot 10^{-3}$
10	20.16721044856	-1.84	$7.93 \cdot 10^{-1}$

We see that $\nu_3(\alpha)$ gives a quite good upper approximation to $\eta(\alpha)$ for all $\alpha \leq 1$ and even for larger values of α.

Example 3.2.3 *Now, consider $q = 10$. We have $\omega_{10} \approx 0.37435$. For all values of α we have computed, $\nu_3(\alpha) < \eta(\alpha) < \min\{\nu_2(\alpha), \nu_4(\alpha)\}$. Some examples are given in the following table.*

α	$\eta(\alpha)$	$\nu_2(\alpha) - \eta(\alpha)$	$\nu_3(\alpha) - \eta(\alpha)$	$\nu_4(\alpha) - \eta(\alpha)$
10^{-6}	1.11160842243	$8.41 \cdot 10^{-11}$	$-1.37 \cdot 10^{-13}$	$2.21 \cdot 10^{-16}$
10^{-3}	1.12722947097	$2.53 \cdot 10^{-6}$	$-1.31 \cdot 10^{-7}$	$6.63 \cdot 10^{-9}$
0.1	1.30724978395	$1.74 \cdot 10^{-3}$	$-9.26 \cdot 10^{-4}$	$4.50 \cdot 10^{-4}$
0.37435	1.55839493272	$9.26 \cdot 10^{-3}$	$-1.00 \cdot 10^{-2}$	$9.26 \cdot 10^{-3}$
1	1.98651890138	$2.89 \cdot 10^{-2}$	$-5.53 \cdot 10^{-2}$	$8.23 \cdot 10^{-2}$
10	6.50984287366	$2.47 \cdot 10^{-1}$	-2.42	11.345

Also for $q = 10$, we get good upper approximations even for $\alpha = 10$.

3.3 On $\mu(d,k)$ when $\kappa \geq d$

In this section, we let $d = \beta\kappa$, where β is a parameter, $0 \leq \beta \leq 1$. Then

$$h(p) = \kappa \frac{\beta \ln(1-Qp) - 1}{\ln(1-p)}$$

and

$$\begin{aligned} \frac{h'(p)}{\kappa} &= \beta \frac{(-Q(1-p)\ln(1-p) + (1-Qp)\ln(1-Qp))}{(1-p)(1-Qp)\ln(1-p)^2} \\ &= -\frac{1}{(1-p)\ln(1-p)^2}. \end{aligned}$$

In particular $h'(p) = 0$ if (and only if)

$$\beta = \frac{1-Qp}{-Q(1-p)\ln(1-p) + (1-Qp)\ln(1-Qp)}. \tag{3.9}$$

We want to solve this for p in terms of β, but again there is no closed form of this solution. We first consider approximations for $\beta \to 0+$. This implies that $\kappa \to \infty$; d may also increase, but then at a slower rate. For $\beta \to 0+$, we see that $p_m \to 1/Q$ (that is, $1 - Qp_m \to 0$) and $h(p_m) \to \kappa/\ln q$.

Theorem 3.3.1 *Let*

$$\lambda = \ln q \quad \textit{and} \quad \Lambda = \ln(\beta\lambda/(q-1)).$$

There exist polynomials $A_i(x)$ and $B_i(x)$ for $i = 0, 1, 2, \ldots$ such that, for any $r \geq 0$,

$$1 - Qp_m = \frac{\lambda}{q-1} \sum_{i=1}^{r} A_i(\Lambda)\beta^i + O(\beta^{r+1}(\ln\beta)^{\deg A_{r+1}}),$$

and

$$\mu(\beta\kappa, \kappa) = \frac{\kappa}{\lambda} \left\{ \sum_{i=0}^{r} B_i(\Lambda)\beta^i + O(\beta^{r+1}(\ln\beta)^{\deg B_{r+1}}) \right\}$$

for $\beta \to 0$. The first few $A_i(x)$ and $B_i(x)$ are given by the following table:

$$\begin{aligned}
A_1(x) &= x + \lambda - 1, \\
A_2(x) &= (2x^2 + (4\lambda - 2)x + 2\lambda^2 - 3\lambda)/2, \\
A_3(x) &= (6x^3 + 3(6\lambda - 1)x^2 + 3(6\lambda^2 - 5\lambda - 2)x + 6\lambda^3 - 11\lambda^2 + 3)/6, \\
A_4(x) &= (12x^4 + (48\lambda + 4)x^3 + (72\lambda^2 - 24\lambda - 30)x^2 \\
&\quad +(48\lambda^3 - 52\lambda^2 - 30\lambda + 12)x - 25\lambda^3 + 12\lambda^4 + 12\lambda + 2)/12, \\
A_5(x) &= (120x^5 + (600\lambda + 170)x^4 + (1200\lambda^2 + 80\lambda - 460)x^3 \\
&\quad +(1200\lambda^3 - 580\lambda^2 - 990\lambda)x^2 \\
&\quad +(600\lambda^4 - 770\lambda^3 - 520\lambda^2 + 300\lambda + 220)x \\
&\quad +120\lambda^5 - 274\lambda^4 + 175\lambda^2 + 70\lambda - 50)/120, \\
A_6(x) &= (120x^6 + (720\lambda + 324)x^5 + (1800\lambda^2 + 720\lambda - 520)x^4 \\
&\quad +(2400\lambda^3 + 40\lambda^2 - 1900\lambda - 460)x^3 \\
&\quad +(1800\lambda^4 - 1110\lambda^3 - 2140\lambda^2 - 90\lambda + 660)x^2 \\
&\quad +(720\lambda^5 - 1044\lambda^4 - 770\lambda^3 + 530\lambda^2 + 720\lambda - 80)x \\
&\quad +120\lambda^6 - 294\lambda^5 + 225\lambda^3 + 135\lambda^2 - 110\lambda - 44)/120, \\
A_7(x) &= (5040x^7 + (35280\lambda + 20916)x^6 + (105840\lambda^2 + 72576\lambda - 16968)x^5 \\
&\quad +(176400\lambda^3 + 78540\lambda^2 - 109410\lambda - 55650)x^4 \\
&\quad +(176400\lambda^4 - 7980\lambda^3 - 208040\lambda^2 - 88620\lambda + 45360)x^3 \\
&\quad +(105840\lambda^5 - 77868\lambda^4 - 159810\lambda^3 - 11970\lambda^2 + 102060\lambda + 15960)x^2 \\
&\quad +(35280\lambda^6 - 56196\lambda^5 - 43848\lambda^4 + 33810\lambda^3 + 61950\lambda^2 - 2100\lambda \\
&\quad -16296)x + 5040\lambda^7 - 13068\lambda^6 + 11368\lambda^4 + 8715\lambda^3 - 6580\lambda^2 \\
&\quad -7266\lambda + 1638)/5040, \\
B_0(x) &= 1, \\
B_1(x) &= -(x - 1), \\
B_2(x) &= -(2x + \lambda - 2)/2, \\
B_3(x) &= ((3\lambda - 6)x + 2\lambda^2 - 6\lambda + 6)/6,
\end{aligned}$$

$$\begin{aligned}
B_4(x) &= -(12x^2+(4\lambda^2+12\lambda-12)x+3\lambda^3+\lambda^2-6\lambda)/12,\\
B_5(x) &= -(60x^3+(120\lambda-30)x^2-(15\lambda^3-115\lambda^2+90\lambda+60)x\\
&\quad -12\lambda^4+50\lambda^3-75\lambda^2+30)/60,\\
B_6(x) &= -(360x^4+(1080\lambda-180)x^3+(1440\lambda^2-1440\lambda)x^2\\
&\quad +(72\lambda^4+780\lambda^3-1950\lambda^2+1260\lambda-540)x\\
&\quad +60\lambda^5+86\lambda^4-645\lambda^3+930\lambda^2-720\lambda+360)/360,\\
B_7(x) &= -(2520x^5+(10080\lambda-420)x^4+(17640\lambda^2-13020\lambda)x^3\\
&\quad +(15120\lambda^3-26460\lambda^2+11340\lambda-5040)x^2\\
&\quad -(420\lambda^5-6958\lambda^4+19005\lambda^3-16170\lambda^2+6300\lambda-1680)x\\
&\quad -360\lambda^6+1764\lambda^5-5292\lambda^4+6720\lambda^3-2835\lambda^2\\
&\quad -840\lambda+1260)/2520.
\end{aligned}$$

Proof: Let $\eta = q-1$ and $\pi = 1-Qp$. Then $1-p = (1+\eta\pi)/q$ and $Q(1-p) = (1+\eta\pi)/\eta$. Hence

$$h'(p) = \kappa \frac{G(\pi,\beta)}{(1-p)(1-Qp)\ln(1-p)^2},$$

where

$$G(\pi,\beta) = -\beta\frac{1+\eta\pi}{\eta}\ln\Big(\frac{1+\eta\pi}{q}\Big)+\beta\pi\ln\pi-\pi.$$

Therefore, $h'(p)=0$ if and only if $G(\pi,\beta)=0$.

If $\pi \to 0+$, then $\ln((1+\eta\pi)/q) \to -\ln q$ and $\pi\ln\pi \to 0$.

Hence, for small π,

$$0 = \frac{G(\pi,\beta)}{\pi} \approx \frac{\ln q}{\eta} - \frac{\pi}{\beta}.$$

Therefore, $\pi \approx \beta\lambda/\eta$. We write $\pi = \beta\lambda(1+y)/\eta$ (where y will depend on β). Then

$$\ln\pi = \Lambda + \ln(1+y).$$

Hence, if $\Gamma(y) = G(\pi,\beta)$ we get

$$\begin{aligned}
\Gamma(y) &= \frac{\beta}{\eta}\Big\{-(1+\beta\lambda(1+y))\ln(1+\beta\lambda(1+y))+(1+\beta\lambda(1+y))\lambda\\
&\quad +\beta\lambda(1+y)(\Lambda+\ln(1+y))-\lambda(1+y)\Big\}.
\end{aligned}$$

We now write $y = \sum_{i=1}^{r} \alpha_i(\Lambda)\beta^i + O(\beta^{r+1})$. Formally treating Λ as if it were a constant, we can take the Taylor expansion of $\Gamma(y)$ in terms of β and we get an expansion of the form $\sum_{i=1}^{\infty} c_i\beta^i$, where the c_i are polynomials of Λ. Since these polynomials c_i must be identically zero, we get equations to determine the polynomials $A_i(x)$. Substituting the series of p_m into $h(p)$ and taking the Taylor expansion, we get the series of $\mu(\beta\kappa, \kappa)$. □

Remark 3.4 *To formally justify that we treat Λ as if it were a constant, we should prove that Λ is algebraically independent of the other quantities involved, that is, there is no non-trivial polynomial equation in Λ with coefficients expressed as rational functions of the remaining quantities. We have not done this, but it highly likely that it is true.*

Remark 3.5 *Note that $\Lambda \to -\infty$ when $\beta \to 0+$.*

As to convergence, the situation is similar to Theorem 3.2.1.

Assuming that $d\beta \ln \beta \to 0$ and taking the first two terms of the approximation we get

$$\mu(d,\kappa) \approx \frac{\kappa}{\ln q} - \frac{d}{\ln q}\left(\ln\left(\frac{d \ln q}{\kappa(q-1)}\right) - 1\right), \tag{3.10}$$

(the other terms go to zero with β).

Again because of the big O term in Theorem 3.3.1, we do not know if the approximation is smaller or larger than the exact value, but we have shown that the approximation is good when $\beta \to 0$.

Taking a good approximation for p_m and inserting this into $h(p)$ we get good upper approximations. For example, taking the first term in the approximation for $1 - Qp_m$, that is, $1 - Qp = \lambda\beta/(q-1)$, then we get

$$\mu(\beta\kappa,\kappa) \leq h\left(\frac{q-1-\beta\ln q}{q}\right) \tag{3.11}$$

Example 3.3.1 *Consider $q = 2$, $d = 2$, $A = 1$, and $M = 2^{1000}$. Then $\kappa = 1000\ln 2$ and $\beta \approx 0.002885$. Solving $h'(p) = 0$ numerically, we get $p \approx 0.4990185$ and $\mu(2, 1000\ln 2) \approx 1020.8737393$. Taking one, two, three, and* *four terms respectively in the expression for $\mu(2, 1000 \ln 2)$ in Theorem 3.3.1, as well as the bound (3.11), we get the following approximations:*

no. of terms		*value*
1		1000
2	= *(3.10)*	1020.8169587
3		1020.8741384
4		1020.8737363
	(3.11)	1021.219184

We have made numeric calculations for a number of $\beta > 0$. The terms after the second do not go to zero, but we may still get good approximations. We take two, three and four terms of the approximation. Here the notations are the following:

$$\sigma(\beta) = \frac{\mu(\beta\kappa, \kappa)}{\kappa} \text{ and } \xi_r(\beta) = \frac{1}{\ln q} \sum_{i=0}^{r} B_i(\Lambda)\beta^i \tag{3.12}$$

Simple calculations show that

$$B_1(\Lambda) \geq 0 \text{ if and only if } \beta \leq \frac{e(q-1)}{\ln(q)},$$

where as usual $e = \exp(1)$,

$$B_2(\Lambda) \geq 0 \text{ if and only if } \beta \leq \frac{e(q-1)}{\sqrt{q}\ln(q)},$$

and $B_3(\Lambda) \geq 0$ if and only if $q \leq 31$ and

$$\frac{q-1}{\sqrt{q}\ln(q)} \exp(-\Omega_q) \leq \beta \leq \frac{q-1}{\sqrt{q}\ln(q)} \exp(\Omega_q)$$

where $\Omega_q = \sqrt{1 - \frac{(\ln(q))^2}{12}}$.

Example 3.3.2 *Consider $q = 2$. We have $B_1(\Lambda) \geq 0$ for $\beta \leq 3.9217$, $B_2(\Lambda) \geq 0$ for $\beta \leq 2.7730$, and $B_3(\Lambda) \geq 0$ for $0.3829 \leq \beta \leq 2.7175$. For all the values of β we have computed, $\xi_1(\beta) < \sigma(\beta)$. For $\beta \lesssim 0.686$ we have $\xi_2(\beta) > \sigma(\beta)$. For $\beta \gtrsim 0.0859$ we have $\xi_3(\beta) > \sigma(\beta)$. Some selected values are given in the following table as an illustration.*

β	$\sigma(\beta)$	$\xi_1(\beta)-\sigma(\beta)$	$\xi_2(\beta)-\sigma(\beta)$	$\xi_3(\beta)-\sigma(\beta)$
10^{-6}	1.4427169439	$-2.14\cdot 10^{-11}$	$1.37\cdot 10^{-16}$	$-1.11\cdot 10^{-21}$
10^{-3}	1.4546436900	$-1.14\cdot 10^{-5}$	$3.38\cdot 10^{-8}$	$-1.15\cdot 10^{-10}$
10^{-1}	2.0167210449	$-4.47\cdot 10^{-2}$	$3.25\cdot 10^{-3}$	$4.83\cdot 10^{-5}$
0.5	3.5174115967	$-5.89\cdot 10^{-1}$	$2.88\cdot 10^{-2}$	$6.96\cdot 10^{-2}$
1	5.0768720947	-1.66	$-1.91\cdot 10^{-1}$	$5.01\cdot 10^{-1}$

We note that

$$\eta(\alpha)=\frac{\mu(d,\alpha d)}{d}=\frac{\mu(\kappa/\alpha,\kappa)}{\kappa/\alpha}=\alpha\sigma(1/\alpha).$$

Therefore, we can compare the approximations given for $\eta(\alpha)$ and $\sigma(\beta)$. For example, for $\alpha=1$, the best upper approximation we got for $\eta(1)$ has an error of $6.46\cdot 10^{-3}$ whereas the best upper approximation of $\sigma(1)$ has an error of $5.01\cdot 10^{-1}$. Even for $\alpha=2.5$ we get a better upper approximation using the best upper approximation to $\eta(2.5)$. However, for $\alpha=3$, the upper approximation of $3\sigma(1/3)$ is better.

Example 3.3.3 *For $q=10$ the situation is similar to $q=2$ and we give some select values without further comments.*

β	$\sigma(\beta)$	$\xi_1(\beta)-\sigma(\beta)$	$\xi_2(\beta)-\sigma(\beta)$	$\xi_3(\beta)-\sigma(\beta)$
10^{-6}	0.4343015082	$-6.53\cdot 10^{-12}$	$4.26\cdot 10^{-17}$	$-3.54\cdot 10^{-22}$
10^{-3}	0.4383243187	$-3.52\cdot 10^{-6}$	$1.08\cdot 10^{-8}$	$-3.94\cdot 10^{-11}$
10^{-1}	0.6509842874	$-1.41\cdot 10^{-2}$	$1.21\cdot 10^{-3}$	$-4.61\cdot 10^{-5}$
0.5	1.2842347128	$-1.86\cdot 10^{-1}$	$2.06\cdot 10^{-2}$	$1.35\cdot 10^{-2}$
1	1.9865189014	$-5.26\cdot 10^{-1}$	$4.18\cdot 10^{-4}$	$1.12\cdot 10^{-1}$

3.4 The redundancy of linear codes

For a linear $[n,k]$ code, $\rho=n-k$ is its *redundancy*, and the relation $n\geq\mu(d,\kappa)$ is of course equivalent to $\rho\geq\mu(d,\kappa)-k$. When A is relatively large, then $\mu(d,\kappa)-k$ is relatively small. In this section, we consider this situation in some details when $\beta\to 0+$, that is, $k\gg d$.

We assume that d is fixed, $n=\mu=\mu(d,\kappa)$ and $A=\gamma n^\delta$ for some fixed positive γ and $\delta\leq d$. Then

$$\kappa=k\ln q-\ln\gamma-\delta\ln\mu<k\lambda. \tag{3.13}$$

Combining (3.13) and Theorem 3.3.1 (for $r=1$) we get

$$\mu=k-\frac{\delta\ln\mu}{\lambda}-\frac{\ln\gamma}{\lambda}+(1-\Lambda)\frac{d}{\lambda}+O(\beta\ln\beta), \tag{3.14}$$

since $\kappa\beta^2 \ln \beta = d\beta \ln \beta$ and d is fixed. In equation (3.14), μ appears both on the left hand side and on the right hand side (both explicit and implicit in Λ). However, the right hand side only contains $\ln \mu$ so by bootstrapping we can show that $\mu \approx k$ and $\kappa \to \infty$. We get $\beta \ln \beta = \frac{d}{\kappa}(\ln d - \ln \kappa)$ and so $O(\beta \ln \beta) = O(\frac{d}{\kappa} \ln \kappa) = O(\ln k/k)$. We note that $1 - \Lambda > 0$ for $\kappa > \frac{\lambda d}{e(q-1)}$ and

$$1 - \Lambda = \ln \kappa + O(1)$$

and so

$$1 < \frac{\mu}{k} < 1 + \frac{d \ln \kappa}{\lambda k} + O\left(\frac{\ln k}{k^2}\right)$$

and

$$0 \leq \ln \mu - \ln k \leq O\left(\frac{\ln k}{k}\right),$$

that is, $\ln \mu = \ln k + O\left(\frac{\ln k}{k}\right)$. Substituting this in (3.13) and (3.14) and simplifying, we get the following theorem.

Theorem 3.4.1 *Let C be a linear $[n, k]$ code with minimum distance d. Assume that $A_d \geq \gamma n^\delta$, where $\delta \leq d$ and γ is some fixed positive number. If the redundancy $\rho = n - k$ of the code satisfies*

$$\rho \geq \frac{d-\delta}{\ln q} \ln k + \frac{1}{\ln q}(d + d\ln(q-1) - \ln \gamma - d \ln d) + O\left(\frac{\ln k}{k}\right), \qquad (3.15)$$

then the code is ugly for error detection.

Remark 3.6 *When $\delta = d$, then the bound on the redundancy is a fixed number (it does not grow with k). Note also that $A_d \leq \binom{n}{d} \leq n^d/d!$. Hence*

$$-\ln \gamma \geq \ln d! \approx d \ln d - d + \frac{1}{2} \ln(2\pi d)$$

(by Stirling's formula) when $\delta = d$. Therefore, the right hand side of (3.15) is lower bounded by

$$\frac{1}{\ln q} \left\{ d \ln(q-1) + \frac{1}{2} \ln(2\pi d) \right\}.$$

Example 3.4.1 *Let C be a $[\nu, \zeta]$ code with minimum distance $d_C \geq 2$. Let H is a parity-check matrix for C. Let C_t be the $[t\nu, (t-1)\nu + \zeta]$ code with parity-check matrix $H_t = H|H| \cdots |H$ (repeated $t > 1$ times). The minimum distance of C_t is clearly 2, and it is easy to find a lower bound on A_2: first choose j such that $1 \leq j \leq \nu$ (this can be done in ν ways); next choose a* *pair (u, v) where $0 \leq u < v < t$ (this can be done in $t(t-1)/2$ ways); finally choose $a \in GF(q) \setminus \{0\}$ (this can be done in $q-1$ ways). In all, there are $(q-1)\nu t(t-1)/2$ possible choices of j, u, v, a. Let $\mathbf{x}_i$, $i = 0, 1 \dots t\nu - 1$ be the columns of H_t. For each choice of j, u, v, a we have*

$$a\mathbf{x}_{u\nu+j} + (-a)\mathbf{x}_{v\nu+j} = \mathbf{0}$$

(since $\mathbf{x}_{u\nu+j} = \mathbf{x}_{v\nu+j}$). Hence

$$A_2 \geq A = (q-1)\nu t(t-1)/2 = \frac{(q-1)(1-1/t)}{2\nu} n^2.$$

The redundancy of C_t is $\nu - \zeta$. We note that

$$-\ln \gamma = \ln 2 + \ln \nu - \ln(q-1) - \ln(1-1/t).$$

By Theorem 3.4.1, if $t \to \infty$ and

$$\nu - \zeta \geq \frac{1}{\ln q} \left\{ 2 + \ln(q-1) - \ln 2 + \ln \nu - \ln(1-1/t) + O\left(\frac{\ln t}{t}\right) \right\},$$

then C_t is ugly. Since $\ln(1-1/t) \approx 1/t = o((\ln t)/t)$ we can conclude that if

$$\nu - \zeta > \frac{1}{\ln q} \left\{ 2 + \ln(q-1) - \ln 2 + \ln \nu \right\}, \tag{3.16}$$

then C_t is ugly for t sufficiently large.

As an example, let C be the binary extended l error correcting BCH code of length $\nu = 2^m$ (where $l \geq 2$). For this code, $\nu - \zeta \geq 2m$ and the right hand side of (3.16) simplifies to $m - 1 + 2/\ln 2 \approx m + 1.9$. Hence, (3.16) is satisfied for all $m \geq 3$ and so C_t is ugly for t sufficiently large.

Chapter 4

Large Codes are Proper for Error Detection

It is simple to show that large codes are proper for error detection. In this chapter we discuss how large is large. Using some known lemmas and theorems for proper codes we give a lower bound for the size of the code to be proper for error detection.

4.1 Some known results

The probability of undetected error is given by (2.1), as it was presented in Chapter 2.

Define

$$A_i(C, \mathbf{x}) = \# \left\{ \mathbf{c} \in C | d_H(\mathbf{c}, \mathbf{x}) = i \right\},$$

so using this notation we can rewrite the distance distribution as:

$$A_i(C) = \frac{1}{M} \sum_{\mathbf{x} \in C} A_i(C, \mathbf{x}).$$

We see that

$$A_i(F_q^n, \mathbf{x}) = \binom{n}{i} (q-1)^i \tag{4.1}$$

for all $\mathbf{x} \in F_q^n$ and $0 \leq i \leq n$. In particular,

$$A_i(F_q^n) = \binom{n}{i} (q-1)^i$$

and so

$$P_{ue}(F_q^n, p) = 1 - (1-p)^n.$$

This is clearly an increasing function on $[0,1]$ and so F_q^n is proper. Hence there exists a bound $B(q,n) \leq q^n$ such that if $M \geq B(q,n)$, then *any* $(n, M; q)$ code is proper. The goal of this chapter is to give non-trivial bounds on $B(q,n)$.

Our main tools to obtain an upper bound are two known results which we refer here as Lemmas 4.1.1 and 4.1.2.

Lemma 4.1.1 *Let C be a code of length n over F_q^n. If*

$$\sum_{i=1}^{l} \frac{l_{(i)}}{n_{(i)}} A_i(C) \geq q \sum_{i=1}^{l-1} \frac{(l-1)_{(i)}}{n_{(i)}} A_i(C), \tag{4.2}$$

for $l = 2, 3, \ldots, n$, where

$$m_{(i)} = m(m-1)...(m-i+1),$$

then C is proper.

This lemma is essentially due to Dodunekova and Dodunekov [11]. They considered only linear codes (over finite fields), however, their proof carries over to the more general situation we consider here. For completeness we give the proof:

Proof: First we define the functions $\Lambda_l(p)$:

$$\Lambda_l(p) = \sum_{j=l}^{n} \binom{n}{j} \Big(\frac{qp}{q-1}\Big)^j \Big(1 - \frac{qp}{q-1}\Big)^{n-j}.$$

Then we have

$$\begin{aligned}
\frac{d\Lambda_l(p)}{dp} &= \sum_{j=l}^{n} \binom{n}{j} \Big\{ j \left(\frac{qp}{q-1}\right)^{j-1} \frac{q}{q-1} \left(1-\frac{qp}{q-1}\right)^{n-j} \\
&\quad + \left(\frac{qp}{q-1}\right)^{j} (n-j) \left(1-\frac{qp}{q-1}\right)^{n-j-1} \left(\frac{-q}{q-1}\right) \Big\} \\
&= \frac{q}{q-1} \sum_{j=l}^{n} \binom{n}{j} j \left(\frac{qp}{q-1}\right)^{j-1} \left(1-\frac{qp}{q-1}\right)^{n-j} \\
&\quad - \frac{q}{q-1} \sum_{j=l+1}^{n} \binom{n}{j-1} (n-j+1) \left(\frac{qp}{q-1}\right)^{j-1} \left(1-\frac{qp}{q-1}\right)^{n-j} \\
&= \frac{q}{q-1} \binom{n}{l} l \Big(\frac{qp}{q-1}\Big)^{l-1} \Big(1-\frac{qp}{q-1}\Big)^{n-l} \\
&\quad + \frac{q}{q-1} \sum_{j=l+1}^{n} \Big\{ \binom{n}{j} j - \binom{n}{j-1} (n-j+1) \Big\} \\
&\quad \cdot \left(\frac{qp}{q-1}\right)^{j-1} \left(1-\frac{qp}{q-1}\right)^{n-j}
\end{aligned}$$

Since $\binom{n}{j} j = \binom{n}{j-1}(n-j+1)$ we get

$$\frac{d\Lambda_l(p)}{dp} = \frac{ql}{q-1} \binom{n}{l} \Big(\frac{qp}{q-1}\Big)^{l-1} \Big(1-\frac{qp}{q-1}\Big)^{n-l} > 0$$

for all $p \in (0, \frac{q-1}{q})$. Hence, the functions $\Lambda_l(p)$ are increasing. Next, rewriting the expression for $P_{ue}(C,p)$ we get:

$$\begin{aligned}
P_{ue}(C,p) &= \sum_{i=1}^{n} A_i(C) \Big(\frac{p}{q-1}\Big)^{i} \Big(\frac{p}{q-1} + 1 - \frac{qp}{q-1}\Big)^{n-i} \\
&= \sum_{i=1}^{n} A_i(C) \Big(\frac{p}{q-1}\Big)^{i} \cdot \sum_{j=0}^{n-i} \binom{n-i}{j} \Big(\frac{p}{q-1}\Big)^{j} \Big(1-\frac{qp}{q-1}\Big)^{n-i-j}
\end{aligned}$$

Now we set $l = i + j$ to get

$$\begin{aligned}
P_{ue}(C,p) &= \sum_{l=1}^{n} \Big(\frac{p}{q-1}\Big)^{l}\Big(1-\frac{qp}{q-1}\Big)^{n-l} \cdot \sum_{i=1}^{l} \binom{n-i}{l-i} A_i(C) \\
&= \sum_{l=1}^{n} \frac{1}{q^l\binom{n}{l}} \Big\{\Lambda_l(p) - \Lambda_{l+1}(p)\Big\} \cdot \sum_{i=1}^{l} \binom{n-i}{l-i} A_i(C) \\
&= \frac{A_1(C)}{qn}\{\Lambda_1(p) - \Lambda_2(p)\} \\
&\quad + \sum_{l=2}^{n} \Big\{ q^{-l}\Lambda_l(p) \cdot \sum_{i=1}^{l} \frac{\binom{n-i}{l-i}}{\binom{n}{l}} A_i(C) - q^{-l}\Lambda_{l+1}(p) \cdot \sum_{i=1}^{l} \frac{\binom{n-i}{l-i}}{\binom{n}{l}} A_i(C) \Big\} \\
&= \frac{A_1(C)}{qn}\{\Lambda_1(p) - \Lambda_2(p)\} \\
&\quad + \sum_{l=2}^{n} \Big\{ q^{-l}\Lambda_l(p) \cdot \sum_{i=1}^{l} \frac{\binom{n-i}{l-i}}{\binom{n}{l}} A_i(C) \\
&\qquad - q^{-l+1}\Lambda_l(p) \cdot \sum_{i=1}^{l-1} \frac{\binom{n-i}{l-i-1}}{\binom{n}{l-1}} A_i(C) \Big\} \\
&= \frac{A_1(C)}{qn}\{\Lambda_1(p) - \Lambda_2(p)\} \\
&\quad + \sum_{l=2}^{n} q^{-l}\Lambda_l(p) \cdot \Big\{ \sum_{i=1}^{l} \frac{\binom{n-i}{l-i}}{\binom{n}{l}} A_i(C) - q \sum_{i=1}^{l-1} \frac{\binom{n-i}{l-i-1}}{\binom{n}{l-1}} A_i(C) \Big\} \\
&= \frac{A_1(C)}{qn}\{\Lambda_1(p) - \Lambda_2(p)\} \\
&\quad + \sum_{l=2}^{n} q^{-l}\Lambda_l(p) \cdot \Big\{ \sum_{i=1}^{l} \frac{l_{(i)}}{n_{(i)}} A_i(C) - q \sum_{i=1}^{l-1} \frac{(l-1)_{(i)}}{n_{(i)}} A_i(C) \Big\}.
\end{aligned}$$

Under the conditions of the lemma, all terms in this sum are increasing functions in p. □

The second result we use is the following lemma.

Lemma 4.1.2 *Let C_1 be an $(n, M_1; q)$ code and C_2 an $(n, M_2; q)$ code such that C_1 and C_2 are disjoint and $C_1 \cup C_2 = F_q^n$. Then*

$$M_2 A_i(C_2) = M_1 A_i(C_1) + (M_2 - M_1)\binom{n}{i}(q-1)^i$$

for $0 \le i \le n$.

For $q = 2$, this lemma is essentially due to Fu, Kløve and Wei [14]. Their proof can be generalized to arbitrary q. However, a simpler proof (which also applies to a more general situation) was given by Abdel-Ghaffar [1]. For completeness we give his proof slightly modified.

Using (4.1), for any $\mathbf{x} \in F_q^n$ we get

$$\binom{n}{i}(q-1)^i = A_i(C_1, \mathbf{x}) + A_i(C_2, \mathbf{x}).$$

Summing over all $\mathbf{x}_1 \in C_1$ we get

$$M_1\binom{n}{i}(q-1)^i = M_1 A_i(C_1) + \sum_{\mathbf{x}_1 \in C_1} A_i(C_2, \mathbf{x}_1). \tag{4.3}$$

Similarly,

$$M_2\binom{n}{i}(q-1)^i = M_2 A_i(C_2) + \sum_{\mathbf{x}_2 \in C_2} A_i(C_1, \mathbf{x}_2). \tag{4.4}$$

Since

$$\begin{aligned}\sum_{\mathbf{x}_1 \in C_1} A_i(C_2, \mathbf{x}_1) &= \#\{(\mathbf{x}_1, \mathbf{x}_2) \mid \mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2, d_H(\mathbf{x}_1, \mathbf{x}_2) = i\} \\ &= \sum_{\mathbf{x}_2 \in C_2} A_i(C_1, \mathbf{x}_2),\end{aligned}$$

the lemma follows by subtracting (4.3) from (4.4). □

From Lemma 4.1.2 we immediately get the following corollary.

Corollary 4.1.1 *For C_1 and C_2 as in Lemma 4.1.2 we have*

$$M_2 P_{ue}(C_2, p) = M_1 P_{ue}(C_1, p) + (M_2 - M_1)(1 - (1-p)^n).$$

4.2 Upper bounds on $B(q,n)$

From Corollary 4.1.1 we see that if $M_2 \ge M_1$ and C_1 is proper, then C_2 is proper. However, C_2 may well be proper even if C_1 is not (assuming $M_2 \ge M_1$). Let

$$\beta(q,n) = \frac{-(qn - n + q) + s}{2(qn - n - q)},$$

where

$$s = \sqrt{(qn - n + q)^2 + 4n(q-1)(qn - n - q)q^n}.$$

Our main explicit result is, in fact, the following lemma.

Lemma 4.2.1 *For C_1 and C_2 as in Lemma 4.1.2, if*

$$M_1 \leq \beta(q,n),$$

then C_2 is proper.

We note that if $M_2 \geq q^n - \beta(q,n)$, then

$$M_1 \leq \beta(q,n).$$

Hence from the lemma we get the following equivalent result.

Theorem 4.2.1 *For $q \geq 2$ and $n \geq 3$ we have*

$$B(q,n) \leq q^n - \beta(q,n);$$

that is, any q-ary code of size at least $q^n - \beta(q,n)$ is proper.

Proof of Lemma 4.2.1: For the proof, we simplify the notations a little and let $A_i = A_i(C_1)$ and $M = M_1$. It is easy to verify that

$$\beta(q,n) < q^n/2$$

for $n \geq 3$, and so

$$M_2 - M_1 = q^n - 2M > 0.$$

Combining Lemmas 4.1.1 and 4.1.2, a sufficient condition for C_2 to be proper is

$$\begin{aligned} &\sum_{i=1}^{l} \frac{l_{(i)}}{n_{(i)}} \{MA_i + (q^n - 2M)\binom{n}{i}(q-1)^i\} \\ &\geq \; q\sum_{i=1}^{l-1} \frac{(l-1)_{(i)}}{n_{(i)}} \{MA_i + (q^n - 2M)\binom{n}{i}(q-1)^i\} \end{aligned}$$

We also note that

$$\sum_{i=1}^{l} \frac{l_{(i)}}{n_{(i)}} \binom{n}{i}(q-1)^i = \sum_{i=1}^{l} \binom{l}{i}(q-1)^i = q^l - 1$$

for $2 \leq l \leq n$. Hence, the sufficient condition for C_2 to be proper can be written as:

$$\begin{aligned} & M \sum_{i=1}^{l} \frac{l_{(i)}}{n_{(i)}} A_i + (q^n - 2M)(q^l - 1) \\ \geq \; & qM \sum_{i=1}^{l-1} \frac{(l-1)_{(i)}}{n_{(i)}} A_i + (q^n - 2M)q(q^{l-1} - 1) \end{aligned} \tag{4.5}$$

for $2 \leq l \leq n$. Omitting the term for $i = l$ in the sum on the left hand side and rearranging, we get the following stronger sufficient condition for C_2 to be proper (stronger in the sense that if (4.6) is satisfied, then (4.5) is satisfied):

$$(q^n - 2M)(q-1) \geq M \sum_{i=1}^{l-1} \frac{(l-1)_{(i)}}{n_{(i)}} \left(q - \frac{l}{l-i} \right) A_i \tag{4.6}$$

for $2 \leq l \leq n$. Since

$$\begin{aligned} \frac{(l-1)_{(i)}}{n_{(i)}} \left(q - \frac{l}{l-i} \right) &= \frac{1}{n} \frac{(l-1)_{(i-1)}}{(n-1)_{(i-1)}} (ql - qi - l) \\ &\leq \frac{1}{n}(ql - q - l), \end{aligned}$$

we get

$$\begin{aligned} \sum_{i=1}^{l-1} \frac{(l-1)_{(i)}}{n_{(i)}} \left(q - \frac{l}{l-i} \right) A_i &\leq \frac{1}{n}(ql - l - q) \sum_{i=1}^{l-1} A_i \\ &\leq \frac{1}{n}(ql - l - q)(M-1) \end{aligned} \tag{4.7}$$

Combining (4.7) with (4.6), we see that a yet stronger sufficient condition for C_2 to be proper is

$$(q^n - 2M)(q-1) \geq \frac{1}{n}(ql - l - q)M(M-1) \tag{4.8}$$

for $2 \leq l \leq n$. The strongest condition is imposed for $l = n$ and this condition is

$$(q^n - 2M)(q-1) \geq \frac{1}{n}(qn - n - q)M(M-1).$$

This is equivalent to

$$M \leq \beta(q,n).$$

This completes the proof of Lemma 4.2.1 and Theorem 4.2.1. □

Using different estimates, we can obtain a stronger bound than Theorem 4.2.1; this bound is not explicit, however, but needs some computation. We refer to it as the algorithmic bound. In (4.6), the terms in the sum are non-positive for $q - l/(l-i) < 0$, that is $i > l(q-1)/q$. If we omit these terms, we get the stronger conditions

$$(q^n - 2M)(q-1) \geq M \sum_{i=1}^{\lfloor l(q-1)/q \rfloor} \frac{(l-1)_{(i)}}{n_{(i)}} \left(q - \frac{l}{l-i} \right) A_i \tag{4.9}$$

for $2 \leq l \leq n$. We note that the condition for $l+1$ is stronger than the condition for l since

$$\frac{(l)_{(i)} \left(q - \frac{l+1}{l+1-i} \right)}{(l-1)_{(i)} \left(q - \frac{l}{l-i} \right)} = \frac{l(ql+q-qi-l-1)}{(l+1-i)(ql-qi-l)} \geq 1$$

for $i \leq l(q-1)/q$. Hence, the strongest condition is the condition for $l = n$, namely

$$(q^n - 2M)(q-1) \geq M \sum_{i=1}^{\lfloor n(q-1)/q \rfloor} \frac{1}{n}(q(n-i)-n)A_i. \tag{4.10}$$

We have $A_i \leq \binom{n}{i}(q-1)^i$. We can use this bound on A_i, but must take into account that $\sum_{i=1}^{n} A_i = M - 1$. Let

$$M_m = \sum_{i=1}^{m} \binom{n}{i} (q-1)^i.$$

Assume that $M > M_{r-1}$ where $r < \lfloor n(q-1)/q \rfloor$, and let

$$S = \sum_{i=1}^{r-1} (q(n-i)-n) \binom{n}{i} (q-1)^i + (q(n-r)-n)(M-1-M_{r-1}).$$

Then

$$\begin{aligned}
S &\geq \sum_{i=1}^{r-1}(q(n-i)-n)\binom{n}{i}(q-1)^i + (q(n-r)-n)\sum_{i=1}^{\lfloor n(q-1)/q\rfloor} A_i \\
&\quad -(q(n-r)-n)\sum_{i=1}^{r-1}\binom{n}{i}(q-1)^i \\
&= \sum_{i=1}^{r-1} q(r-i)\binom{n}{i}(q-1)^i + (q(n-r)-n)\sum_{i=1}^{\lfloor n(q-1)/q\rfloor} A_i \\
&= \sum_{i=1}^{\lfloor n(q-1)/q\rfloor} (q(n-i)-n)A_i + \sum_{i=1}^{r-1} q(r-i)\Big(\binom{n}{i}(q-1)^i - A_i\Big) \\
&\quad + \sum_{i=r+1}^{\lfloor n(q-1)/q\rfloor} q(i-r)A_i \\
&\geq \sum_{i=1}^{\lfloor n(q-1)/q\rfloor} (q(n-i)-n)A_i.
\end{aligned}$$

Hence, by (4.10) we get the following stronger sufficient condition for C_2 to be proper:

$$\begin{aligned}
(q^n-2M)(q-1) &\geq \frac{M}{n}\sum_{i=1}^{r-1}(q(n-i)-n)\binom{n}{i}(q-1)^i \\
&\quad + \frac{M}{n}(q(n-r)-n)(M-1-M_{r-1}).
\end{aligned}$$

Solving this inequality, we get a maximal value which we denote by $\beta_r(q,n)$. Provided $\beta_r(q,n) > M_{r-1}$ (this was a requirement for the derivation above) we have

$$B(q,n) \leq q^n - \beta_r(q,n).$$

In particular, $\beta_1(q,n) = \beta(q,n)$. We know that we can use r determined by $M_{r-1} < \beta_r(q,n) \leq M_r$. Usually, this $\beta_r(q,n)$ is maximal.

We illustrate the procedure by some numerical examples.

For $q = 3$ and $n = 20$, we get the following values:

r	M_{r-1}	$\beta_r(3, 20)$
1	0	61395.65
2	40	64045.19
3	800	67033.73
4	9920	70002.06
5	87440	69030.75

From $M_{r-1} < \beta_r(q, n) \leq M_r$, it follows that $r = 4$.

For $q = 3$ and $n = 21$, we get the following values:

r	M_{r-1}	$\beta_r(3, 21)$
1	0	106136.11
2	42	110468.15
3	882	115339.98
4	11522	120392.84
5	107282	121081.17
6	758450	91120.30

From $M_{r-1} < \beta_r(q, n) \leq M_r$, it follows that $r = 5$.

For $q = 3$ and $n = 30$, we get the following values:

r	M_{r-1}	$\beta_r(3, 30)$
1	0	14721667.41
2	60	15125073.74
3	1800	15563519.44
4	34280	16041435.64
5	472760	16551735.16
6	5032952	16953441.99
7	43034552	16027100.38

From $M_{r-1} < \beta_r(q, n) \leq M_r$, it follows that $r = 6$.

For $q = 4$ and $n = 25$, we get the following values:

r	M_{r-1}	$\beta_r(4,25)$
1	0	34486676.40
2	75	35501204.40
3	2775	36610795.63
4	64875	37829286.84
5	1089525	39141074.74
6	14000115	40100659.55
7	143106015	36183958.73

From $M_{r-1} < \beta_r(q,n) \leq M_r$, it follows that $r = 6$.

4.3 A lower bound on $B(q,n)$

A lower bound on $B(q,n)$ is obtained by giving an explicit code which is not proper. Let

$$\begin{aligned}\gamma(2,n) &= 2^{\lfloor (n+3)/2 \rfloor}\\ \gamma(q,n) &= q^{\lfloor (n+2)/2 \rfloor} \text{ for } q \geq 3.\end{aligned}$$

In [13], Fu and Kløve showed that a result that implies that

$$B(2,n) \geq 2^n - \gamma(2,n)$$

for $n \geq 7$ (the paper considered *good* binary linear codes, and codes that are not good are also not proper). The construction from [13] can be generalized to arbitrary q to show the following result.

Theorem 4.3.1 *For $q \geq 2$ and $n \geq 4$ we have*

$$B(q,n) > q^n - \gamma(q,n).$$

Proof: Let C_1 be the $(n, q^k; q)$ code

$$\{(\mathbf{x}, \mathbf{0}) \in F_q^n \mid \mathbf{x} \in F_q^k\}.$$

It is easy to see that

$$A_i = \binom{k}{i}(q-1)^i$$

for all i. Hence,

$$\begin{aligned} P_{\mathrm{ue}}(C_1,p) &= \sum_{i=1}^{k}\binom{k}{i}p^i(1-p)^{k-i}(1-p)^{n-k} \\ &= (1-p)^{n-k}-(1-p)^n. \end{aligned}$$

Therefore, if $f(p)=(q^n-q^k)P_{\mathrm{ue}}(C_2,p)$, Corollary 4.1.1 shows that

$$\begin{aligned} f(p) &= q^k((1-p)^{n-k}-(1-p)^n) \quad +(q^n-2q^k)(1-(1-p)^n) \\ &= q^n-2q^k+q^k(1-p)^{n-k}-(q^n-q^k)(1-p)^n. \end{aligned}$$

Hence,

$$f'(p)=-(n-k)q^k(1-p)^{n-k-1}+n(q^n-q^k)(1-p)^{n-1},$$

and so

$$f'\Big(\frac{q-1}{q}\Big)=-(n-k)q^{2k+1-n}+n(q^n-q^k)q^{-n+1}<0$$

if

$$-q^k(n-k)+n(q^{n-k}-1)<0. \tag{4.11}$$

Let $k=(n+\alpha)/2$ (where $\alpha=1,2,3$). Then (4.11) is equivalent to

$$q^{(n-\alpha)/2}\Big(n-q^\alpha\frac{n-\alpha}{2}\Big)-n<0.$$

For $\alpha=3$ and $q\ge 2$ we have

$$n-q^\alpha\frac{n-\alpha}{2}\le n-2^3\frac{n-3}{2}\le 0 \text{ for } n\ge 4.$$

For $\alpha=2$ and $q\ge 2$ we have

$$n-q^\alpha\frac{n-\alpha}{2}\le n-2^2\frac{n-2}{2}\le 0 \text{ for } n\ge 4.$$

For $\alpha=1$ and $q\ge 3$ we have

$$n-q^\alpha\frac{n-\alpha}{2}\le n-3\frac{n-1}{2}\le 0 \text{ for } n\ge 3.$$

Hence, $f'\Big(\frac{q-1}{q}\Big)<0$ for $q=2$, $k=\lfloor(n+3)/3\rfloor$, and $n\ge 4$; and also for $q\ge 3$, $k=\lfloor(n+2)/3\rfloor$, and $n\ge 4$. This completes the proof. □

We note that

$$\beta(q,n) \approx -\frac{qn-n+q}{2(qn-n-q)} + q^{n/2}\sqrt{\frac{qn-n}{qn-n-q}}.$$

As a numerical example, for $q = 3$ and $n = 20$, we get

$$\beta(3,20) = 61395.6455950\ldots$$

whereas the approximation above is 61395.6455923.

On the other hand, $\gamma(q,n)$ is also of the order $q^{n/2}$ (e.g. for $q \geq 3$ and n odd, $\gamma(q,n) = q^{n/2}\sqrt{q}$). Hence, $B(q,n) = q^n - q^{n/2}\omega(q,n)$, where

$$\sqrt{\frac{qn-n}{qn-n-q}} \lesssim \omega(q,n) \leq q$$

for $q \geq 3$ or n even (for $q = 2$ and n odd, our upper bound is $2\sqrt{2}$). Moreover, the algorithmic bound improves the lower bound on $\omega(q,n)$.

Chapter 5

The Function $S(m,n)$

5.1 Computing $S(m,n)$

For $\mathbf{x} = (x_0, x_1, \ldots, x_{k-1}) \in F_q^k$, let

$$w(\mathbf{x}) = \sum_{i=0}^{k-1} x_i, \text{ the } \textit{weight} \text{ of } \mathbf{x}.$$

For integers $m \geq 0$, $S(m,n)$ is defined by

$$S(m,n) = \#\{\mathbf{x} \in F_q^m \mid w(\mathbf{x}) = n\}.$$

Note that for $S(m,n) = 0$ for $n < 0$. A sequence $\mathbf{x}$ such that $w(\mathbf{x}) = n$ is known as a *composition* of n. Compositions have been studied by a number of people, starting with MacMahon, see [24].

For $q = 2$ we get $S(m,n) = \binom{m}{n}$. Explicit expressions for $S(m,n)$ for general q seems to be complicated. However, we can find a number of useful results on $S(m,n)$.

Lemma 5.1.1 *For all $m \geq 0$ and $n \geq 0$ we have*

$$S(m,n) = \sum_{\substack{(m_0,m_1,\ldots,m_{q-1}) \\ \sum m_i = m \text{ and } \sum i m_i = n}} \frac{m!}{m_0! m_1! \cdots m_{q-1}!}. \tag{5.1}$$

Proof: Using generating functions, we see that

$$\sum_{n=0}^{\infty} S(m,n) x^n = (1 + x + \cdots + x^{q-1})^m,$$

a relation given by MacMahon [24, p.151] (in a different notation).

Hence, (5.1) follows by the multinomial theorem. □

For each term in (5.1) we have

$$\sum_{i=0}^{q-1} im_i = \sum_{i=1}^{q-1} im_i \le (q-1)\sum_{i=1}^{q-1} m_i \le (q-1)m.$$

Hence, if $n > (q-1)m$, the sum is empty and we get the following corollary.

Corollary 1 *For $m \ge 0$ and $n > (q-1)m$ we have*

$$S(m,n) = 0. \tag{5.2}$$

We also have a symmetry.

Corollary 2 *If $m \ge 0$ and $0 \le n \le (q-1)m$, then*

$$S(m,n) = S(m,(q-1)m-n). \tag{5.3}$$

Proof: Let $(m_0, m_1, \ldots, m_{q-1})$ satisfy $\sum m_i = m$ and $\sum im_i = n$, and let $m'_i = m_{q-1-i}$ for $0 \le i \le q-1$. Then $\sum m'_i = m$ and

$$\sum im'_i = \sum (q-1-i)m_i = (q-1)\sum m_i - \sum im_i = (q-1)m - n.$$

Since $m'_0!m'_1!\cdots m'_{q-1}! = m_0!m_1!\cdots m_{q-1}!$, this means that for each term in the sum (5.1) for $S(m,n)$, there is an equal corresponding term in the sum for $S(m,(q-1)m-n)$. □

For actual computation of $S(m,n)$, a recursion is usually more convenient. This recursion will also be used to obtain further results.

Lemma 5.1.2

$$S(0,0) = 1 \textit{ and } S(0,n) = 0 \textit{ for } n > 0.$$

Further, for $m \ge 1$ we have

$$S(m,n) = \sum_{j=0}^{q-1} S(m-1,n-j). \tag{5.4}$$

Proof: By definition, $S(m,n)$ is the number of $(x_1, x_2, \cdots x_m)$ such that $x_1 + x_2 + \cdots + x_m = n$ and where $0 \leq x_i \leq q-1$. The number of these where $x_m = j$ is therefore $S(m-1, n-j)$. Summing for all j we get (5.4). □

For integers $l \geq 0, a, b$ where $a \leq b$ (one or both may be negative) we have the following well known sum.

$$\sum_{j=a}^{b} \binom{l+j}{l} = \binom{l+b+1}{l+1} - \binom{l+a}{l+1}. \tag{5.5}$$

An easy induction using (5.5) and (5.4) shows that

$$S(m,n) = \binom{m+n-1}{m-1} \text{ for } m \geq 1 \text{ and } 0 \leq n \leq q-1. \tag{5.6}$$

Combining (5.4) and (5.6) we get the following result.

Lemma 5.1.3 *If* $m \geq 2$ *and* $0 \leq n_1 < n_2 \leq (q-1)m/2$*, then*

$$S(m, n_1) < S(m, n_2).$$

Proof: It is sufficient to prove the result for $n_2 = n_1 + 1$; the general result then follows immediately by induction. We write $n_1 = n$ and $n_2 = n+1$ and prove the result by induction. For $m = 2$, (5.6) shows that

$$S(2,n) = n+1 < n+2 = S(2, n+1) \text{ for } 0 \leq n < (q-1).$$

For $m > 2$, (5.4) implies that

$$\begin{aligned} S(m, n+1) - S(m,n) &= \sum_{j=0}^{q-1} S(m-1, n+1-j) - \sum_{j=0}^{q-1} S(m-1, n-j) \\ &= S(m-1, n+1) - S(m-1, n-q+1). \end{aligned}$$

To complete the induction, we consider two cases.

Case I: $n+1 \leq (q-1)(m-1)/2$. Then

$$S(m-1, n-q+1) < S(m-1, n+1)$$

by the induction hypothesis.

Case II: $(q-1)(m-1)/2 < n+1 \leq (q-1)m/2$. Then

$$(m-1)(q-1)-n-1 < (q-1)(m-1)/2.$$

Further $2n+1 < 2n+2 \leq (q-1)m$ and so

$$n-q+1 < (q-1)(m-1)-n-1.$$

Hence

$$\begin{array}{rcl} S(m-1,n-q+1) & < & S(m-1,(m-1)(q-1)-n-1) \\ & = & S(m-1,n+1). \end{array}$$

□

Another very useful result that follows from (5.4) and (5.5) is the following.

Lemma 5.1.4 *For $m \geq 1$, $b \geq 0$, and $0 \leq l \leq m-1$, there exist integers $A_{m,b,l}$ such that*

$$S(m,n) = \sum_{l=0}^{m-1} A_{m,b,l} \binom{n-bq+l}{l} \tag{5.7}$$

for $b(q-1) < n \leq (b+1)(q-1)$. For $b=0$ we have

$$A_{m,0,l} = 0 \text{ for } l < m-1 \text{ and } A_{m,0,m-1} = 1. \tag{5.8}$$

For $b \geq 1$ we have

$$A_{m,b,0} = \sum_{j=0}^{m-2} A_{m-1,b-1,j} \binom{q-b+j+1}{j+1} - \sum_{j=0}^{m-2} A_{m-1,b,j} \binom{j-b+1}{j+1},$$

and

$$A_{m,b,l} = A_{m-1,b,l-1} - A_{m-1,b-1,l-1} \tag{5.9}$$

for $1 \leq l \leq m-1$.

Proof: First we note that Corollary 1 implies that we have $A_{m,b,l} = 0$ for all l when $b \geq m$. We prove the lemma by induction. For $b = 0$ we have (5.8) which follows immediately from (5.6). This also completes the proof for $m = 1$. For the induction step, let $m \geq 2$ and $b \geq 1$ and suppose (5.7) is true for smaller values of the parameters. By (5.4) we get

$$\begin{aligned}
S(m,n) &= \sum_{j=n-q+1}^{b(q-1)} S(m-1,j) + \sum_{j=b(q-1)+1}^{n} S(m-1,j) \\
&= \sum_{j=n-q+1}^{b(q-1)} \sum_{l=0}^{m-2} A_{m-1,b-1,l} \binom{j-(b-1)q+l}{l} \\
&\quad + \sum_{j=b(q-1)+1}^{n} \sum_{l=0}^{m-2} A_{m-1,b,l} \binom{j-bq+l}{l} \\
&= \sum_{l=0}^{m-2} A_{m-1,b-1,l} \sum_{j=n-q+1}^{b(q-1)} \binom{j-(b-1)q+l}{l} \\
&\quad +\sum_{l=0}^{m-2} A_{m-1,b,l} \sum_{j=b(q-1)+1}^{n} \binom{j-bq+l}{l} \\
&= \sum_{l=0}^{m-2} A_{m-1,b-1,l} \left\{ \binom{b(q-1)-(b-1)q+l+1}{l+1} \right. \\
&\qquad \left. -\binom{n-q-(b-1)q+l+1}{l+1} \right\} \\
&\quad +\sum_{l=0}^{m-2} A_{m-1,b,l} \left\{ \binom{n-bq+l+1}{l+1} \right. \\
&\qquad \left. -\binom{b(q-1)-bq+l+1}{l+1} \right\} \\
&= \sum_{l=0}^{m-2} \left\{ A_{m-1,b-1,l} \binom{-b+q+l+1}{l+1} \right. \\
&\qquad \left. -A_{m-1,b,l} \binom{-b+l+1}{l+1} \right\} \\
&\quad + \sum_{l=0}^{m-2} (A_{m-1,b,l} - A_{m-1,b-1,l}) \binom{n-bq+l+1}{l+1}.
\end{aligned}$$

This proves the lemma. Note that we have used that $\binom{-b+l+1}{l+1} = 0$ for $l+1-b \geq 0$, that is, $l \geq b-1$. □

Next we determine $A_{m,b,l}$ explicitly for $b=1$ and $b=2$.

Corollary 3 *For $b = 1 < m$ we have*

$$\begin{aligned} A_{m,1,l} &= \binom{q+m-l-2}{m-l-1} \text{ for } 0 \le l < m-1, \\ A_{m,1,m-1} &= -(m-1). \end{aligned}$$

Proof: We use Lemma 5.1.4. For $l = 0$ we get

$$A_{m,1,0} = \sum_{j=0}^{m-2} A_{m-1,0,j} \binom{q+j}{j+1} = \binom{q+m-2}{m-1}.$$

For $1 \le l < m-1$ we get $A_{m,1,l} = A_{m-1,1,l-1}$, and so by induction,

$$A_{m,1,l} = A_{m-l,1,0} = \binom{q+m-l-2}{m-l-1}.$$

Finally, for $l = m-1$ we get $A_{m,1,m-1} = A_{m-1,1,m-2} - 1$. Since $A_{1,1,0} = 0$, we get $A_{m,1,m-1} = -(m-1)$ by induction. □

Corollary 4 *For $b = 2 < m$ we have*

$$\begin{aligned} A_{m,2,l} &= \sum_{i=2}^{m-l-1} \sum_{j=0}^{i-2} \binom{q+i-j-2}{i-j-1} \binom{q-1+j}{j+1} \\ &\quad - \sum_{i=2}^{m-l-1} (i-1) \binom{q+i-2}{i} - l \binom{q+m-l-2}{m-l-1} \end{aligned}$$

for $0 \le l < m-1$,

$$A_{m,2,m-1} = \binom{m-1}{2}.$$

Proof: We use Lemma 5.1.4. For $l = 0$ we get

$$A_{m,2,0} = \sum_{j=0}^{m-2} A_{m-1,1,j} \binom{q-1+j}{j+1} + A_{m-1,2,0}.$$

By induction

$$\begin{aligned} A_{m,2,0} &= \sum_{i=2}^{m-1} \sum_{j=0}^{i-1} A_{i,1,j} \binom{q-1+j}{j+1} \\ &= \sum_{i=2}^{m-1} \sum_{j=0}^{i-2} \binom{q+i-j-2}{i-j-1} \binom{q-1+j}{j+1} - \sum_{i=2}^{m-1} (i-1) \binom{q+i-2}{i}. \end{aligned}$$

For $0 < l < m-1$, we get

$$A_{m,2,l} = A_{m-1,2,l-1} - A_{m-1,1,l-1}.$$

By induction we get

$$A_{m,2,l} = A_{m-l,2,0} - \sum_{j=1}^{l} A_{m-j,1,l-j} = A_{m-l,2,0} - l\binom{q+m-l-2}{m-l-1}.$$

Finally,

$$A_{m,2,m-1} = A_{m-1,2,m-2} - A_{m-1,1,m-2} = A_{m-1,2,m-2} + (m-2).$$

By induction, $A_{m,2,m-1} = \binom{m-1}{2}$. □

Similar explicit expressions may be found for $b = 3, 4, \ldots$; they become increasingly complicated.

Chapter 6

t-EC-AUED Codes

This chapter and Chapter 7 are devoted to the non-symmetric channels. Here we will present t-EC-AUED codes. Böinck and van Tilborg gave a bound on the length of binary t-EC-AUED codes. In this chapter a generalization of this bound to arbitrary alphabet size is given. This generalized Böinck - van Tilborg bound, combined with constructions, is used to determine the length of some optimal binary and ternary t-EC-AUED codes. As it was mentioned in Chapter 2, the size of optimal 0-EC-AUED codes is the value of the function $S(n, \lceil n(q-1)/2 \rceil)$, the function $S(m,n)$ was presented in Chapter 5. So we will make computations for $t > 0$, but for completeness, we give also the codes for $t = 0$.

6.1 A generalized Böinck-van Tilborg bound

In this chapter the expression $t+1$ occurs on many places, so we find it convenient to use the notation $T \overset{\text{def}}{=} t+1$.

The lower bound which was derived by Böinck and van Tilborg for the length of a binary t-EC-AUED codes, rewritten in our notations is:

$$n_2(a,T) \geq \left\lceil \left(4 - \frac{2}{\lceil a/2 \rceil}\right) T \right\rceil . \tag{6.1}$$

In this section we generalize Böinck - van Tilborg bound to non-binary codes. Let

$$f(m_0, m_1, \ldots, m_{q-1}) = \sum_{0 \leq i < j \leq (q-1)} m_i m_j,$$

$$\begin{aligned} S_1 &= m_0 + m_1 + ... + m_{q-1} = \sum_{i=0}^{q-1} m_i, \\ S_2 &= m_0^2 + m_1^2 + ... + m_{q-1}^2 = \sum_{i=0}^{q-1} m_i^2. \end{aligned}$$

Then $S_1^2 = S_2 + 2f(m_0, m_1, \ldots, m_{q-1})$ and so

$$f(m_0, m_1, .., m_{q-1}) = \frac{1}{2}(S_1^2 - S_2).$$

Let $\lambda(a)$ be the maximum of $f(m_0, m_1, .., m_{q-1})$ over $(m_0, m_1, \ldots, m_{q-1})$, where $m_0, m_1, ..., m_{q-1}$ are non-negative integers such that $S_1 = a$.

Lemma 6.1.1 *If C is an $(T-1)$-EC-AUED code of length n and size a, then*

$$n \geq \frac{a(a-1)T}{\lambda(a)}.$$

Proof: Consider $\sum_{\substack{\mathbf{x},\mathbf{y}\in C \\ \mathbf{x}\neq\mathbf{y}}} N(\mathbf{x}, \mathbf{y})$. Since C is a $(T-1)$-EC-AUED code, $N(\mathbf{x}, \mathbf{y}) \geq T$ for all distinct $\mathbf{x}, \mathbf{y} \in C$, and so

$$\sum_{\substack{\mathbf{x},\mathbf{y}\in C \\ \mathbf{x}\neq\mathbf{y}}} N(\mathbf{x}, \mathbf{y}) \geq a(a-1)T. \tag{6.2}$$

Let $m_{l,i}$ be the number of codewords $\mathbf{x}$ such that $x_l = i$. Then,

$$\begin{aligned} \sum_{\substack{\mathbf{x},\mathbf{y}\in C \\ \mathbf{x}\neq\mathbf{y}}} N(\mathbf{x}, \mathbf{y}) &= \sum_{l=1}^{n} \sum_{0\leq i<j\leq q-1} m_{l,i}\, m_{l,j} \\ &= \sum_{l=1}^{n} f(m_{0,l}, m_{1,l}, \ldots, m_{q-1,l}) \\ &\leq n\lambda(a). \end{aligned}$$

Combining this with (6.2), the lemma follows. □

Next we find an explicit expression for $\lambda(a)$. We note that if the m_i were real numbers, then the maximum of $f(m_0, m_1, \ldots, m_{q-1})$ would be obtained for $m_i = a/(q-1)$ for all i. For non-negative integers m_i let the maximum be obtained for

$$(m_0, m_1, \ldots, m_{q-1}) = (\mu_0, \mu_1, \ldots, \mu_{q-1}).$$

Because of the symmetry we may assume that

$$\mu_0 \leq \mu_1 \leq \cdots \leq \mu_{q-1}.$$

Further, by assumption,

$$\sum_{i=0}^{q-1} \mu_i = a.$$

In particular, $\mu_{q-1} \geq 1$. Let

$$\begin{aligned} m_0 &= \mu_0 + 1, \\ m_{q-1} &= \mu_{q-1} - 1, \\ m_i &= \mu_i \text{ for } 1 \leq i \leq q-2. \end{aligned}$$

Then

$$\begin{aligned} 0 &\leq 2f(\mu_0, \mu_1, \ldots, \mu_{q-1}) - 2f(m_0, m_1, \ldots, m_{q-1}) \\ &= (a^2 - \mu_0^2 - \mu_1^2 - \cdots - \mu_{q-1}^2) - (a^2 - m_0^2 - m_1^2 - \cdots - m_{q-1}^2) \\ &= -\mu_0^2 - \mu_{q-1}^2 + (\mu_0 + 1)^2 + (\mu_{q-1} + 1)^2 \\ &= 2\mu_0 - 2\mu_{q-1} + 2. \end{aligned}$$

Hence, $\mu_{q-1} \leq \mu_0 + 1$. This implies that if $a = \alpha q + \beta$, where $0 \leq \beta \leq q-1$, then

$$\begin{array}{ll} \mu_i = \alpha & \text{for } 0 \leq i < q - \beta, \\ \mu_i = \alpha + 1 & \text{for } q - \beta \leq i < q - 1. \end{array}$$

Hence

$$\begin{aligned} \lambda(a) &= \frac{1}{2}\left\{a^2 - (q-\beta)\alpha^2 - \beta(\alpha+1)^2\right\} \\ &= \frac{a(a-\alpha) - (a - \alpha q)(1+\alpha)}{2}. \end{aligned}$$

Combining this with Lemma 6.1.1, we get the following bound.

Theorem 6.1.1 *For $a \geq 2$ and $T \geq 1$ we have*

$$n_q(a, T) \geq GBT_q(a, T),$$

where

$$GBT_q(a, T) = \left\lceil \frac{2a(a-1)T}{a(a-\alpha) - (a - \alpha q)(\alpha + 1)} \right\rceil$$

and $\alpha = \lfloor a/q \rfloor$.

For $q = 2$, this is exactly the bound (6.1).

Since

$$GBT_q(q\mu + (q-1), T) = GBT_q(q\mu + q, T),$$

an immediate corollary of the theorem is:

Corollary 6.1.1 *A q-ary $(T-1)$-EC-AUED code of length*

$$n < GBT_q(q\mu + q, T)$$

has size $a \leq q\mu + (q-2)$.

6.2 A method to determine or estimate $n_q(a, T)$

It appears that in many cases, the Böinck - van Tilborg bound and also its generalization is best possible, that is, we have equality in Theorem 6.1.1. In [16], we developed a method to prove this in the binary case and in [26] in the ternary case, using an efficient construction method. For a given a, the construction is recursive and requires a computer search for some small values of T to start the recursion. The validity of the recursion is based on two lemmas involving the generalized Böinck - van Tilborg bound. We state and prove them next.

Lemma 6.2.1 *For all $a > 0$, $T_1 \geq 0$, and $T_2 \geq 0$, we have*

$$n_q(a, T_1 + T_2) \leq n_q(a, T_1) + n_q(a, T_2).$$

Proof: We represent a code of size a and length n by an $(a \times n)$ matrix with the codewords as the rows. Let C_1 be a $(T_1 - 1)$-EC-AUED code of size a and length $n_q(a, T_1)$ and C_2 a $(T_2 - 1)$-EC-AUED code of size a and length $n_q(a, T_2)$. Let $C = C_1 | C_2$ (matrix concatenation). This is an asymmetric code of size a and length

$$n = n_q(a, T_1) + n_q(a, T_2).$$

Let $(\mathbf{x}|\mathbf{x}')$ and $(\mathbf{y}|\mathbf{y}')$ be distinct codewords of C, where $\mathbf{x}, \mathbf{x}' \in C_1$ and $\mathbf{y}, \mathbf{y}' \in C_2$. Then

$$N((\mathbf{x}|\mathbf{x}'), (\mathbf{y}|\mathbf{y}')) = N(\mathbf{x}, \mathbf{y}) + N(\mathbf{x}', \mathbf{y}') \geq T_1 + T_2.$$

Hence, C is an $(T_1 + T_2 - 1)$-EC-AUED code of length $n_q(a, T_1) + n_q(a, T_2)$. This proves the lemma. □

Lemma 6.2.2 *If*

$$n_q(a,T_1) = GBT_q(a,T_1),$$
$$n_q(a,T_2) = GBT_q(a,T_2),$$

and

$$GBT_q(a,T_1) + GBT_q(a,T_2) = GBT_q(a,T_1+T_2),$$

then

$$n_q(a,T_1+T_2) = GBT_q(a,T_1+T_2).$$

Proof: Let C_1, C_2 and C be defined as in the proof of the previous lemma. Then, by Theorem 6.1.1, Lemma 6.2.1, and the given conditions, we get

$$\begin{array}{rcl} GBT_q(a,T_1+T_2) & \leq & n_q(a,T_1+T_2) \\ & \leq & n_q(a,T_1) + n_q(a,T_2) \\ & = & GBT_q(a,T_1) + GBT_q(a,T_2) \\ & = & GBT_q(a,T_1+T_2). \end{array}$$

In particular, $n_q(a,T_1+T_2) = GBT_q(a,T_1+T_2)$. □

6.3 Optimal binary $(T-1)$-EC-AUED codes

To determine $n_2(a,T)$ we only need to consider a even (by Corollary 6.1.1).

In [6] optimal codes of size $a = 2\mu$ are constructed for $\mu = 1, 2, 3$ by a more direct, but less efficient, method. The size of the codes and the bounds on n are the following:

$$a = 2 \text{ for } 2T \leq n < 3T,$$
$$a = 4 \text{ for } 3T < n < \frac{10}{3}T,$$
$$a = 6 \text{ for } \frac{10}{3}T \leq n < \frac{7}{2}T.$$

We construct optimal codes for $\mu = 4, 5, 6, 7$ by a combination of a computer search and the use of Lemmas 6.2.1, 6.2.2, and Corollary 6.1.1 (for $q = 2$).

When we are considering binary codes we will use the notation

$$BT(2\mu, T) = \left\lceil \left(4 - \frac{2}{\mu}\right) T \right\rceil$$

for the Böinck-van Tilborg bound.

Theorem 6.3.1 *For* $T \equiv 2 \pmod 4$, *we have*

$$\left\lceil \frac{7}{2}T \right\rceil \leq n_2(8,T) \leq \left\lceil \frac{7}{2}T \right\rceil + 1$$

and

$$n_2(8,T) = \left\lceil \frac{7}{2}T \right\rceil,$$

otherwise.

Proof: For $a = 8$ we have $BT(8,T) = \lceil \frac{7}{2}T \rceil$, hence $n_2(8,T) \geq \left\lceil \frac{7}{2}T \right\rceil$.

For $T = 1$, $BT(8,1) = 4$. According to de Bruijn et al., [10], the size of an optimal 1-EC-AUED code of length 4 is $\binom{4}{2} = 6$. So there is no 1-EC-AUED code of length 4 and size 8. A computer search shows that $n_1(8,1) = 5$, which is 1 above the bound.

A computer search shows that for $T = 2$ there is no code meeting the bound $BT(8,2)$ and the length of the best code is 1 above the bound, so $n_2(8,2) = 8$.

Matrices showing this are:

$$C_1 = \begin{bmatrix} 00011 \\ 00101 \\ 00110 \\ 01001 \\ 01010 \\ 01100 \\ 10001 \\ 10010 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 00000111 \\ 00011001 \\ 00101010 \\ 01001100 \\ 01110000 \\ 10010010 \\ 10100100 \\ 11000001 \end{bmatrix}.$$

For $T = 3, 4, 5$, there are codes with length $n_2(8,T) = \lceil \frac{7}{2}T \rceil$. Matrices showing this are:

$$C_3 = \begin{bmatrix} 00000011111 \\ 00011100011 \\ 00101101100 \\ 01010110100 \\ 01101010001 \\ 01110001010 \\ 10011011000 \\ 10100110010 \end{bmatrix}, \quad C_4 = \begin{bmatrix} 00000001111111 \\ 00011110000111 \\ 00101110111000 \\ 01110011001001 \\ 10110101010010 \\ 11000111100100 \\ 11011000011100 \\ 11101000100011 \end{bmatrix},$$

$$C_5 = \begin{bmatrix} 000000000111111111 \\ 000011111000001111 \\ 000101111011110000 \\ 011010011100110001 \\ 011011100111000010 \\ 101100011101000110 \\ 101101100100011001 \\ 110110001010011010 \end{bmatrix}.$$

A computer search shows that there is no codes meeting the bound for $T = 6$. The best code is with length 1 above the bound. So $n_2(8,6) = 22$ and one of the possibilities to obtain C_6 is a concatenation of C_1 and C_5.

Using $BT(8,T) + BT(8,4) = BT(8,T+4)$ for all T and Lemma 6.2.2 it follows that the recursion which we will use to obtained codes for all T is $C_T = C_4|C_{T-4}$. For all $T \not\equiv 2 \pmod 4$ the length of the codes is exactly $BT(8,T)$ and for $T \equiv 2 \pmod 4$ the length is bounded by $BT(8,T)$ and $BT(8,T)+1$.

Note that the fact $n_2(8,T) = n_2(7,T)$ follows from Corollary 6.1.1. □

Theorem 6.3.2 *For $T \geq 2$ we have*

$$n_2(10,T) = \left\lceil \frac{18}{5}T \right\rceil.$$

Proof: For size 10 we use the same method. We have $BT(10,T) = \lceil \frac{18}{5}T \rceil$, hence $n_2(10,T) \geq \lceil \frac{18}{5}T \rceil$. We use that

$$BT(10,T) + BT(10,5) = BT(10,T+5).$$

The recursion which we will use to obtain codes for all T is $C_T = C_5|C_{T-5}$. We have to note that according to de Bruijn et al., [10], the size of the optimal 1-EC-AUED code of length 4, (since $BT(10,1) = 4$), is 6. So there is no code meeting the bound for $T = 1$. A computer search shows that the length of the best code is 1 above the bound, so $n_2(10,1) = 5$ and the code is presented with the following matrix :

$$C_1 = \begin{bmatrix} 00011 \\ 00101 \\ 00110 \\ 01001 \\ 01010 \\ 01100 \\ 10001 \\ 10010 \\ 10100 \\ 11000 \end{bmatrix}.$$

For all other $T \geq 2$ there are codes meeting the bound. The codes C_T for $T = 2, 3, 4, 5$, needed to start the recursion, are:

$$C_2 = \begin{bmatrix} 00001111 \\ 00110011 \\ 00111100 \\ 01010101 \\ 01011010 \\ 01100110 \\ 01101001 \\ 10010110 \\ 10011001 \\ 10100101 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 00000011111 \\ 00011100011 \\ 00101101100 \\ 01010110100 \\ 01101010001 \\ 01110001010 \\ 10011011000 \\ 10100110010 \\ 10110000101 \\ 11000101001 \end{bmatrix},$$

$$C_4 = \begin{bmatrix} 000000001111111 \\ 000011110000111 \\ 000101110111000 \\ 001110011001001 \\ 010110101010010 \\ 011000111100100 \\ 011011000011100 \\ 011101000100011 \\ 100111001100100 \\ 101001011010010 \end{bmatrix}, \quad C_5 = \begin{bmatrix} 000000000111111111 \\ 000011111000001111 \\ 000101111011110000 \\ 011010011100110001 \\ 011011100111000010 \\ 101100011101000110 \\ 101101100100011001 \\ 110110001010011010 \\ 110110100001100101 \\ 111001010010101100 \end{bmatrix}.$$

If we use the recursion to obtain $C_6 = C_1|C_5$ the code is with length 1 above the bound, since the length of C_1 is 1 above the bound. However there is code meeting exactly the bound, namely $C_6 = C_3|C_3$.

This proves the theorem. The fact $n_2(10,T) = n_2(9,T)$ follows from Corollary 6.1.1. □

Theorem 6.3.3 *For* $T \equiv 3 \pmod 6$*, we have*

$$\left\lceil \frac{11}{3}T \right\rceil \leq n_2(12,T) \leq \left\lceil \frac{11}{3}T \right\rceil + 1$$

and

$$n_2(12,T) = \left\lceil \frac{11}{3}T \right\rceil,$$

otherwise.

Proof: For size 12 we have $BT(12,T) = \lceil \frac{11}{3}T \rceil$, hence $n_2(12,T) \geq \lceil \frac{11}{3}T \rceil$.

For $T = 1$, $BT(12,1) = 4$. According to de Bruijn et al., [10], the optimal 1-EC-AUED code of length 4 has size 6. So there is no 1-EC-AUED codes of size 12 and length 4. A computer search shows that the best code is with length $n_2(12,1) = 6$. The matrix showing this is:

$$C_1 = \begin{bmatrix} 000011 \\ 000101 \\ 000110 \\ 001001 \\ 001010 \\ 001100 \\ 010001 \\ 010010 \\ 010100 \\ 011000 \\ 100001 \\ 100010 \end{bmatrix}.$$

Computations have shown that $n_2(12,2) = \lceil \frac{11}{3}T \rceil$ but for $T = 3$ there are no codes which meeting this bound. The length is 1 above the bound. For $T = 4,5,6,7,8$ we have codes meeting exactly the bound. Matrices showing this for $T = 2,3,4,5,6,7,8$ are:

$$C_2 = \begin{bmatrix} 00001111 \\ 00110011 \\ 00111100 \\ 01010101 \\ 01011010 \\ 01100110 \\ 01101001 \\ 10010110 \\ 10011001 \\ 10100101 \\ 10101010 \\ 11000011 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 000000011111 \\ 001111100000 \\ 010001100011 \\ 010010101100 \\ 010101010100 \\ 011010010001 \\ 011100001010 \\ 100011000101 \\ 100100100110 \\ 100110011000 \\ 101000101001 \\ 101001010010 \end{bmatrix},$$

$$C_4 = \begin{bmatrix} 000000001111111 \\ 000011110000111 \\ 000101110111000 \\ 001110011001001 \\ 010110101010010 \\ 011000111100100 \\ 011011000011100 \\ 011101000100011 \\ 100111001100100 \\ 101001011010010 \\ 101010100101010 \\ 101100100010101 \end{bmatrix}, \quad C_5 = \begin{bmatrix} 000000000111111111 \\ 000001111100001111 \\ 000010111011110000 \\ 001101001110011000 1 \\ 001101110011100001 0 \\ 010110001110100011 0 \\ 010110110010001100 1 \\ 011011000101001101 0 \\ 011011010000110010 1 \\ 011100101001010110 0 \\ 100111001001100100 1 \\ 100111100000011011 0 \end{bmatrix},$$

$$C_6 = \begin{bmatrix} 0000000000011111111111 \\ 0000011111100000011111 \\ 0000101111101111100000 \\ 0011110001110001100011 \\ 0011110110010110001100 \\ 0111001001111010010100 \\ 1011011010001001111000 \\ 1100110010111100010001 \\ 1101001100110100101010 \\ 1101100101000011011001 \\ 1110010101001101000110 \\ 1110101010000010100111 \end{bmatrix},$$

$$C_7 = \begin{bmatrix} 00000000000001111111111111 \\ 00000011111110000000111111 \\ 00000101111110111111000000 \\ 00111010001111000111000011 \\ 00111010110011111000001100 \\ 01011100011101001001110100 \\ 01011111000010010110111000 \\ 01101101100101110000010011 \\ 10101111001000101001101001 \\ 10110101110001000110100101 \\ 11010110100100100011001110 \\ 11101000111000001110011010 \end{bmatrix}, \quad C_8 = C_2|C_6.$$

Since $BT(12,T) + BT(12,6) = BT(12,T+6)$, the recursion which we will use to obtain codes for all T is $C_T = C_6|C_{T-6}$.

The best code for $T = 9$ is $C_9 = C_3|C_6$, which has length 1 above the bound. So for all $T \equiv 3 \pmod 6$ the length of the codes is bounded by $BT(12,T)$ and $BT(12,T)+1$. For the rest $T \not\equiv 3 \pmod 6$, the length of the codes is exactly $BT(12,T)$. This proves the theorem.

Again from Corollary 6.1.1 it follows that $BT(12,T) = BT(11,T)$. □

Theorem 6.3.4 *For $T \geq 2$ we have*

$$n_2(14,T) = \left\lceil \frac{26}{7} T \right\rceil .$$

Proof: For size 14 we have $BT(14,T) = \lceil \frac{26}{7} T \rceil$, hence $n_2(14,T) \geq \lceil \frac{26}{7} T \rceil$.

For $T = 1$, $BT(14,1) = 4$. According again to de Bruijn et al., [10], the size of the optimal 1-EC-AUED code of length 4 is 6, so there is no 1-EC-AUED code of length 4 and size 14. A computer search shows that the best code is with length $n_2(14,1) = 6$ and the matrix showing this is:

$$C_1 = \begin{bmatrix} 000011 \\ 000101 \\ 000110 \\ 001001 \\ 001010 \\ 001100 \\ 010001 \\ 010010 \\ 010100 \\ 011000 \\ 100001 \\ 100010 \\ 100100 \\ 101000 \end{bmatrix}.$$

Since $BT(14,T)+BT(14,7) = BT(14,T+7)$, we need codes for T from 2 to 8 to start the recursion, which is $C_T = C_7|C_{T-7}$. The codes for $T = 2,3,4,5,6,7,8$ are presented below:

$$C_2 = \begin{bmatrix} 00001111 \\ 00110011 \\ 00111100 \\ 01010101 \\ 01011010 \\ 01100110 \\ 01101001 \\ 10010110 \\ 10011001 \\ 10100101 \\ 10101010 \\ 11000011 \\ 11001100 \\ 11110000 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 000000111111 \\ 001111000011 \\ 010011001101 \\ 010011110010 \\ 011101010100 \\ 011110101000 \\ 100111011000 \\ 101011100100 \\ 101100001101 \\ 101100110010 \\ 110101100001 \\ 110110000110 \\ 111001001010 \\ 111010010001 \end{bmatrix},$$

$$C_4 = \begin{bmatrix} 000000001111111 \\ 000011110000111 \\ 000101110111000 \\ 001110011001001 \\ 010110101010010 \\ 011000111100100 \\ 011011000011100 \\ 011101000100011 \\ 100111001100100 \\ 101001011010010 \\ 101010100101010 \\ 101100100010101 \\ 110001101001001 \\ 110010010110001 \end{bmatrix}, \quad C_5 = \begin{bmatrix} 0000000000111111111 \\ 0000011111000001111 \\ 0000101111011110000 \\ 0011010011100110001 \\ 0011011100111000010 \\ 0101100011101000110 \\ 0101101100100011001 \\ 0110110001010011010 \\ 0110110100001100101 \\ 0111001010010101100 \\ 1001110010011001001 \\ 1001111000000110110 \\ 1010100110100101010 \\ 1010101001110000101 \end{bmatrix},$$

$$C_6 = \begin{bmatrix} 00000000000011111111111 \\ 00000011111100000011111 \\ 00000101111101111100000 \\ 00011110001110001100011 \\ 00011110110010110001100 \\ 00111001001111010010100 \\ 01011011010001001111000 \\ 01100110010111100010001 \\ 01101001100110100101010 \\ 01101100101000011011001 \\ 01110010101001101000110 \\ 01110101010000010100111 \\ 10011101100001100010011 \\ 10100111100010001110100 \end{bmatrix},$$

$$C_7 = \begin{bmatrix} 00000000000001111111111111 \\ 00000011111110000000111111 \\ 00000101111110111111000000 \\ 00111010001111000111000011 \\ 00111010110011111000001100 \\ 01011100011101001001110100 \\ 01011111000010010110111000 \\ 10011101100101110000010011 \\ 10101111001000101001101001 \\ 11001110110000001110000111 \\ 11100011100101010011100100 \\ 11100100010111100100101010 \\ 11110000111000010101011001 \\ 11110001001010101010010110 \end{bmatrix}, \quad C_8 = C_4|C_4.$$

This proves the theorem for all $T > 1$.

From Corollary 6.1.1 we have $BT(14, T) = BT(13, T)$. □

6.4 Optimal ternary $(T-1)$-EC-AUED codes

In this section we use the method described above for $q = 3$.

Theorem 6.4.1 *For $T \geq 1$ we have*

$$\begin{aligned} &n_3(3,T) = 2T, \\ &n_3(4,T) = \left\lceil \tfrac{12}{5}T \right\rceil, \\ &n_3(5,T) = n_3(6,T) = \left\lceil \tfrac{5}{2}T \right\rceil. \end{aligned}$$

Proof: We first give the proof and the construction of the codes for $a = 6$. We have $GBT_3(6,T) = \left\lceil \frac{5}{2}T \right\rceil$. Hence, $GBT_3(6,T) + GBT_3(6,2) = GBT_3(6,T+2)$. Codes showing the stated result for $T = 1$ and $T = 2$ are

$$C_1 = \begin{bmatrix} 200 \\ 020 \\ 002 \\ 100 \\ 010 \\ 001 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 12200 \\ 21020 \\ 20102 \\ 02012 \\ 00221 \\ 11111 \end{bmatrix}.$$

As shown in the proofs of Lemmas 6.2.1 and 6.2.2, codes C_T that prove the result for general T are obtained by the recursion $C_T = C_2|C_{T-2}$.

Note that the fact $n_3(5,T) = n_3(6,T)$ follows from Corollary 6.1.1.

For $a = 3$, we only need one matrix to start the recursion $C_T = C_1|C_{T-1}$:

$$C_1 = \begin{bmatrix} 02 \\ 11 \\ 20 \end{bmatrix}.$$

For $a = 4$, we need 5 matrices to start the recursion $C_T = C_5|C_{T-5}$:

$$C_1 = \begin{bmatrix} 001 \\ 020 \\ 110 \\ 200 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 00022 \\ 01101 \\ 10110 \\ 22000 \end{bmatrix},$$

$$C_3 = \begin{bmatrix} 00000122 \\ 00111011 \\ 02022000 \\ 11200001 \end{bmatrix}, \quad C_5 = \begin{bmatrix} 000000022222 \\ 001111101111 \\ 120012210001 \\ 212220000010 \end{bmatrix},$$

$$C_4 = C_2|C_2.$$

□

We finally consider $a = 7, 8, 9$.

Theorem 6.4.2 *We have*

$$\left\lceil \frac{21}{8} T \right\rceil \leq n_3(7,T) \leq \left\lceil \frac{8}{3} T \right\rceil$$

for all $T \geq 1$.

Proof: For $a = 7$, we have $GBT_q(7,T) = \lceil \frac{21}{8} T \rceil$. We do not know if this bound can be met in all cases. Computations have shown that it is met for $T \leq 7$. Codes proving this are the following.

$$C_1 = \begin{bmatrix} 012 \\ 021 \\ 102 \\ 111 \\ 120 \\ 201 \\ 210 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 00001222 \\ 00122011 \\ 02110102 \\ 11010211 \\ 12201001 \\ 20210020 \\ 21101110 \end{bmatrix}.$$

Further, let

$$C_2 = C_1|C_1, \;\; C_T = C_3|C_{T-3} \text{ for } T = 4, 5, 6, 7.$$

For $T = 8$ the best code we have found so far is $C_4|C_4$ whose length is one above the bound. If there exists a code C_8 meeting the bound for $T = 8$, the recursive construction $C_T = C_8|C_{T-8}$ gives codes meeting the bound for all T. However, if this is not the case, we still can use a recursive construction to get estimates. We note that $\lceil \frac{21}{8}T \rceil = \lceil \frac{8}{3}T \rceil$ for $1 \leq T \leq 7$, and the construction $C_T = C_3|C_{T-3}$ gives a code of length $\lceil \frac{8}{3}T \rceil$ for all T. □

Theorem 6.4.3 *We have*

$$\left\lceil \frac{8}{3}T \right\rceil \leq n_3(8, T) \leq n_3(9, T) \leq \left\lceil \frac{8}{3}T \right\rceil + 1$$

for $T \equiv 1 \pmod 3$*, and*

$$n_3(8, T) = n_3(9, T) = \left\lceil \frac{8}{3}T \right\rceil$$

otherwise.

Proof: For $a = 9$ the bound is $GBT_3(9, T) = \lceil \frac{8}{3}T \rceil$. There are no codes meeting the bound for $T = 1$, the bound is 3, but the shortest code has length 4:

$$C_1 = \begin{bmatrix} 0002 \\ 0011 \\ 0020 \\ 0101 \\ 0110 \\ 0200 \\ 1001 \\ 1010 \\ 1100 \end{bmatrix}.$$

For $T=2$ and $T=3$, there are codes meeting the bound:

$$C_2 = \begin{bmatrix} 000022 \\ 001111 \\ 020201 \\ 022010 \\ 110011 \\ 112200 \\ 200210 \\ 202001 \\ 221100 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 00002222 \\ 01111112 \\ 02220002 \\ 10120121 \\ 11202011 \\ 12011201 \\ 20210210 \\ 21022100 \\ 22101020 \end{bmatrix}.$$

Since $GBT_3(9,T)+GBT_3(9,3) = GBT_3(9,T+3)$, the recursive construction is $C_T = C_3|C_{T-3}$. A computer search shows that there is no code for $T=4$ which meet the bound. The best code is with length 1 above the bound. One of the possibilities to obtain C_4 is $C_2|C_2$.

So for $T \equiv 1(mod\ 3)$ the length of the codes is bounded by $GBT_3(9,T)$ and $GBT_3(9,T)+1$ and for the rest T, which are not equivalent to $1(mod\ 3)$ the length of the codes is exactly $GBT_3(9,T)$. □

Chapter 7

Bose-Lin Codes

Bose and Lin introduced a class of systematic codes for detection of binary asymmetric errors. In this chapter, we describe a generalization to q-ary asymmetric error detecting codes. For these codes, the possible undetectable errors are characterized and the undetectable errors of minimum weight are determined.

7.1 Notations and definitions

For $\mathbf{x} = (x_0, x_1, \ldots, x_{k-1}) \in F_q^k$, let

$$\begin{aligned} w(\mathbf{x}) &= \sum_{i=0}^{k-1} x_i, \text{ the } \textit{weight} \text{ of } \mathbf{x}, \\ u(\mathbf{x}) &= \sum_{i=0}^{k-1} (q-1-x_i), \text{ the } \textit{coweight} \text{ of } \mathbf{x}. \end{aligned}$$

Clearly, $w(\mathbf{x}) + u(\mathbf{x}) = k(q-1)$.

For non-negative integers a and s, where $a < q^s$, let

$$\langle a \rangle_s = (a_{s-1}, a_{s-2}, \ldots, a_0) \in F_q^s,$$

where

$$a = \sum_{i=0}^{s-1} a_i q^i, \; a_i \in F_q,$$

that is, $\langle a \rangle_s$ is essentially the q-ary expansion of a. Define

$$w(a) = w(\langle a \rangle_s) = \sum_{i=0}^{s-1} a_i.$$

For example, if $q = 5$ and $k = 6$, then

$$w((1,3,1,0,2,4)) = 11 \text{ and } u((1,3,1,0,2,4)) = 13.$$

Further, $\langle 33 \rangle_4 = (0,1,1,3)$ and $w(33) = 5$.

For two sequences $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{y} \subseteq \mathbf{x}$ denotes that $\mathbf{x}$ covers $\mathbf{y}$, that is, $y_i \leq x_i$ for all i.

For integers a and n, let $[a]_n$ denote the (least non-negative) residue of a modulo n.

For integers $m \geq 0$ and $n \geq 0$, $S(m,n)$ denote the number of sequences in F_q^m of weight n. A discussion of expressions for and computation of $S(m,n)$ was given in Chapter 5.

7.2 Generalized Bose-Lin codes

The Bose-Lin codes are systematic codes determined by four (integral) parameters, $q \geq 2$ (symbol alphabet size), k (number of information symbols), r (number of check symbols), and ω where $0 \leq \omega \leq r$. We use the notations $\rho = r - \omega$, $\sigma = S(\omega, \lfloor (q-1)\omega/2 \rfloor)$, $\theta = q^\rho$, and $\mu = \sigma\theta$. Finally, let

$$\{\mathbf{b}_{\omega,0}, \mathbf{b}_{\omega,1}, \ldots, \mathbf{b}_{\omega,\sigma-1}\},$$

be the set of sequences in F_q^ω of weight $\lfloor (q-1)\omega/2 \rfloor$.

A codeword is a sequence in F_q^{k+r}. It consists of an information sequence $\mathbf{x}$ with k information symbols concatenated by a check sequence $c(\mathbf{x})$ with r check symbols. Let $u = u(\mathbf{x})$, the coweight of $\mathbf{x}$. The check sequence, which depends only on u modulo μ, that is, the integer $[u]_\mu$, will be determined as follows. Express $[u]_\mu$ as

$$[u]_\mu = \alpha\theta + [u]_\theta,$$

where $0 \leq [u]_\theta < \theta$. Since $[u]_\mu < \mu = \sigma\theta$, we get $0 \leq \alpha < \sigma$. The *check sequence* is defined by $c(\mathbf{x}) = c_1(\mathbf{x})|c_2(\mathbf{x})$ where $|$ denotes concatenation,

$$c_1(\mathbf{x}) = \mathbf{b}_{\omega,\alpha}$$

(a sequence of length ω and weight $\lfloor (q-1)\omega/2 \rfloor$), and

$$c_2(\mathbf{x}) = \langle [u]_\theta \rangle_\rho$$

(the q-ary expansion of the residue of u modulo θ).

For $q = 2$ (and ω even) we get the Bose-Lin codes [7]. For $\omega = 0$ (and general q) we get the Bose-Pradhan codes [8]; actually, Bose and Pradhan considered the codes with the smallest possible value of r, namely

$$r = \lceil \log_q((q-1)k + 1) \rceil.$$

For $\omega = 2$ (and general q) we get the codes studied by Bose, Elmougy, and Tallini [5], [12].

Our main results are the following theorems.

Theorem 7.2.1 *Let C be the GBL-code with parameters q, k, r, ω. A codeword $\mathbf{x}|c(\mathbf{x}) \in C$ can be transformed to the codeword $\mathbf{y}|c(\mathbf{y}) \neq \mathbf{x}|c(\mathbf{x})$ by transmission over a complete q-ASC if and only if*

$$\mathbf{y} \subset \mathbf{x} \text{ and } \langle \lambda \rangle_\rho \subseteq \langle [u(\mathbf{x})]_\theta \rangle_\rho,$$

where $\lambda = [u(\mathbf{x}) - u(\mathbf{y})]_\mu$.

Proof: The sent codeword is $\mathbf{x}|c_1(\mathbf{x})|c_2(\mathbf{x})$ and the received codeword is $\mathbf{y}|c_1(\mathbf{y})|c_2(\mathbf{y})$, where $\mathbf{x} \neq \mathbf{y}$. Looking at each part, we see that this is possible if and only if

$$\mathbf{y} \subset \mathbf{x}, \tag{7.1}$$

$$c_1(\mathbf{y}) \subseteq c_1(\mathbf{x}), \tag{7.2}$$

and

$$c_2(\mathbf{y}) \subseteq c_2(\mathbf{x}). \tag{7.3}$$

Since $w(c_1(\mathbf{y})) = w(c_1(\mathbf{x})) = \lfloor (q-1)\omega/2 \rfloor$, (7.2) is possible if and only if

$$c_1(\mathbf{y}) = c_1(\mathbf{x}). \tag{7.4}$$

Further, by definition, $c_2(\mathbf{x})$ is the q-ary expansion of $u(\mathbf{x})$ modulo θ and $c_2(\mathbf{y})$ is the q-ary expansion of $u(\mathbf{y})$ modulo θ. Also, we note that if $\mathbf{y} \subset \mathbf{x}$, then $u(\mathbf{y}) > u(\mathbf{x})$. Hence, suppose that (7.1), (7.3), and (7.4) are satisfied, and let

$$u = u(\mathbf{x}) \tag{7.5}$$

and

$$j\mu - \lambda = u(\mathbf{y}) - u(\mathbf{x}), \tag{7.6}$$

where $j \geq 1$ and $0 \leq \lambda < \mu$. Note that $\lambda = [u(\mathbf{x}) - u(\mathbf{y})]_\mu$. Let

$$[u]_\mu = \alpha\theta + [u]_\theta \text{ and } [u + j\mu - \lambda]_\mu = \beta\theta + [u + j\mu - \lambda]_\theta.$$

Since $[u + (j\mu - \lambda)]_\mu = [u - \lambda]_\mu$ and $[u + (j\mu - \lambda)]_\theta = [u - \lambda]_\theta$, we get from the definition of the code that

$$\begin{array}{ll} c_1(\mathbf{x}) = \mathbf{b}_{\omega,\alpha} & c_2(\mathbf{x}) = \langle [u]_\theta \rangle_\rho, \\ c_1(\mathbf{y}) = \mathbf{b}_{\omega,\beta} & c_2(\mathbf{y}) = \langle [u - \lambda]_\theta \rangle_\rho. \end{array}$$

Hence, (7.4) is satisfied if and only if $\mathbf{b}_{\omega,\alpha} = \mathbf{b}_{\omega,\beta}$, that is, if and only if

$$\beta = \alpha, \tag{7.7}$$

and (7.3) is satisfied if and only if

$$\langle [u - \lambda]_\theta \rangle_\rho \subseteq \langle [u]_\theta \rangle_\rho. \tag{7.8}$$

We consider (7.8) in more details. First we note that since $\alpha = \beta$, we must have $\lambda < \theta$. Next, we see that (7.8) states that the q-ary expansion of $[u]_\theta$ covers the q-ary expansion of $[u - \lambda]_\theta$. This implies in particular that the q-ary expansion of λ must be covered by the q-ary expansion of $[u]_\theta$. In particular, we must have $\lambda \leq [u]_\theta$. On the other hand, if $\lambda \leq [u]_\theta$, then (7.7) is satisfied; further (7.8) is satisfied exactly when $\langle \lambda \rangle_\rho \subseteq \langle [u]_\theta \rangle_\rho$. We also note that $\langle \lambda \rangle_\rho \subseteq \langle [u]_\theta \rangle_\rho$ implies that $\lambda \leq [u]_\theta$. This completes the proof of the theorem. □

Remark 7.1 *For a non-complete q-ASC, the "only if" part of the theorem is still true, but the "if" part may not be true. A simple example to illustrate this is given by the code with parameters* $q = 3$, $k = 2$, $r = 1$, $\omega = 0$. *For this code, both* (222) *and* (201) *are codewords. For a complete channel the first can clearly be transformed into the second. However, for a channel where for example 2 can not be changed to 1, this is not the case.*

Theorem 7.2.2 *Let C be the GBL-code with parameters q, k, r, ω, and assume that the transmission is over a complete q-ASC. The code C detects all errors of weight up to*

$$(\sigma-1)\theta+(q-1)\rho,$$

but there are undetectable errors of weight

$$(\sigma-1)\theta+(q-1)\rho+1.$$

Undetectable errors of this minimal weight occur exactly for codewords of coweight $t\theta-\delta$ for $t\geq 1$ and $1\leq\delta\leq q$. For such codewords, an error is undetectable if the weight of the error to the information part is

$$\mu-\theta-\epsilon+\delta$$

where $1\leq\delta\leq\epsilon\leq q$ and the last ρ symbols of the check part, namely

$$(q-1,q-1,\ldots,q-1,q-\delta),$$

are changed to

$$(0,0,\ldots,0,q-\epsilon).$$

Proof: We use the notations introduced in the proof of Theorem 7.2.1. From the proof of Theorem 7.2.1, we see that the weight of the undetectable error considered is

$$\begin{aligned}
&w(\mathbf{x}|c(\mathbf{x}))-w(\mathbf{y}|c(\mathbf{y}))\\
&\quad= w(\mathbf{x})-w(\mathbf{y})+w(c_2(\mathbf{x}))-w(c_2(\mathbf{y}))\\
&\quad= w(\mathbf{x})-w(\mathbf{y})+w(\langle[u]_\theta\rangle_\rho)-w(\langle[u-\lambda)]_\theta\rangle_\rho)\\
&\quad= u(\mathbf{y})-u(\mathbf{x})+w(\langle[u]_\theta\rangle_\rho)-w(\langle[u-\lambda)]_\theta\rangle_\rho)\\
&\quad= j\mu-\lambda+w(\langle\lambda\rangle_\rho)\\
&\quad= j\mu-\lambda+w(\lambda).
\end{aligned}$$

Suppose that

$$\lambda=\sum_{i=0}^{\rho-1}a_iq^i,\ \text{and}\ \lambda'=\sum_{i=0}^{\rho-1}a'_iq^i,$$

where $a_i,a'_i\in F_q$ and $\langle\lambda\rangle_\rho\subseteq\langle\lambda'\rangle_\rho$, that is, $a_i\leq a'_i$ for all i. Then

$$(\lambda'-w(\lambda'))-(\lambda-w(\lambda))=\sum_{i=0}^{\rho-1}(a'_i-a_i)(q^i-1)\geq 0. \tag{7.9}$$

We note that $\langle \theta-1\rangle_\rho = (q-1, q-1, \ldots, q-1)$. Hence $w(\langle \theta-1\rangle_\rho) = (q-1)\rho$ and $\langle [u]_\theta\rangle_\rho \subseteq \langle \theta-1\rangle_\rho$. By (7.9), if also $\langle \lambda\rangle_\rho \subseteq \langle [u]_\theta\rangle_\rho$, then

$$\lambda - w(\lambda) \leq [u]_\theta - w([u]_\theta) \leq \theta - 1 - (q-1)\rho. \tag{7.10}$$

Therefore, the weight of an undetectable error is lower bounded as follows:

$$j\mu - (\lambda - w(\lambda)) \geq \mu - \theta + 1 + (q-1)\rho = (\sigma-1)\theta + (q-1)\rho + 1.$$

Consider the undetectable errors of minimal weight. We see that we get equality in (7.9) if and only if $a'_i = a_i$ for all $i \geq 1$. Hence we have equality in both places in (7.10) if and only if

$$j = 1,\ \lambda = \theta - \epsilon,\ \text{and}\ [u]_\theta = \theta - \delta,$$

where $1 \leq \delta \leq \epsilon \leq q$. This proves Theorem 7.2.2. □

A natural question is: given q and r, which value of ω maximizes

$$A(q, r, \omega) = (\sigma - 1)\theta + (q-1)\rho.$$

For $q = 2$ it was shown by a simple proof in [20] that the maximum is obtained for $\omega = 4$ when $r \geq 5$. For $q \geq 3$ it seems to be much more complicated to answer the question. Numerical computations indicate that for $q \geq 3$, the maximum is obtained for $\omega = 2$. The computations show that $A(q, r, 2) > A(q, r, \omega)$ for $3 \leq q \leq 7$ and $\omega \leq 100$. We give some computations in the table below for the first few value of ω and for $3 \leq q \leq 7$. The function $S(m, n)$ which we use was discussed in more details in Chapter 5.

Lemma 5.1.4 can be used to compute $S(\omega, \lfloor (q-1)\omega/2\rfloor)$ for $\omega = 1, 2, \ldots$. Note that $\lfloor (q-1)\omega/2\rfloor = ((q-1)\omega - 1)/2$ when ω is odd and q is even, and $\lfloor (q-1)\omega/2\rfloor = (q-1)\omega/2$ otherwise. The first few values are given by the following table.

ω	$S(\omega, (q-1)\omega/2)$	$S(\omega, ((q-1)\omega-1)/2)$
1	1	1
2	q	–
3	$(3q^2+1)/4$	$3q^2/4$
4	$(2q^3+q)/3$	–
5	$(115q^4+50q^2+27)/192$	$(115q^4+20q^2)/192$
6	$(11q^5+5q^3+4q)/20$	–

Using this table we compute the value of the function $S(\omega, \lfloor (q-1)\omega/2 \rfloor)$ for $\omega = 1, 2, ..., 6$. Then we compute

$$\begin{aligned} A(q, r, \omega) &= (\sigma - 1)\theta + (q-1)\rho \\ &= (S(\omega, \lfloor (q-1)\omega/2 \rfloor) - 1)\, q^{r-\omega} + (q-1)(r-\omega). \end{aligned}$$

In the table below are given the first few values of $A(q, r, \omega)$, $r \geq \omega$ for $3 \leq q \leq 7$ and $1 \leq \omega \leq 6$.

$q \setminus \omega$	1	2	3
3	$2(r-1)$	$2 \cdot 3^{r-2} + 2(r-2)$	$6 \cdot 3^{r-3} + 2(r-3)$
4	$3(r-1)$	$3 \cdot 4^{r-2} + 3(r-2)$	$11 \cdot 4^{r-3} + 3(r-3)$
5	$4(r-1)$	$4 \cdot 5^{r-2} + 4(r-2)$	$18 \cdot 5^{r-3} + 4(r-3)$
6	$5(r-1)$	$5 \cdot 6^{r-2} + 5(r-2)$	$26 \cdot 6^{r-3} + 5(r-3)$
7	$6(r-1)$	$6 \cdot 7^{r-2} + 6(r-2)$	$36 \cdot 7^{r-3} + 6(r-3)$

$q \setminus \omega$	4	5	6
3	$18 \cdot 3^{r-4} + 2(r-4)$	$50 \cdot 3^{r-5} + 2(r-5)$	$281 \cdot 3^{r-6} + 2(r-6)$
4	$43 \cdot 4^{r-4} + 3(r-4)$	$154 \cdot 4^{r-5} + 3(r-5)$	$579 \cdot 4^{r-6} + 3(r-6)$
5	$72 \cdot 5^{r-4} + 4(r-4)$	$380 \cdot 5^{r-5} + 4(r-5)$	$1750 \cdot 5^{r-6} + 4(r-6)$
6	$145 \cdot 6^{r-4} + 5(r-4)$	$779 \cdot 6^{r-5} + 5(r-5)$	$4331 \cdot 6^{r-6} + 5(r-6)$
7	$230 \cdot 7^{r-4} + 6(r-4)$	$1450 \cdot 7^{r-5} + 6(r-5)$	$9330 \cdot 7^{r-6} + 6(r-6)$

Theorem 7.2.3 *Let $q \geq 3$. We have $A(q, r, 2) > A(q, r, 1)$ for $r \geq 3$. For $3 \leq \omega \leq 6$ we have $A(q, r, \omega - 1) > A(q, r, \omega)$ for $r \geq \omega$.*

Proof: By definition,

$$\begin{aligned} A(q, r, \omega) &= (\sigma - 1)\theta + (q-1)\rho \\ &= (S(\omega, \lfloor (q-1)\omega/2 \rfloor) - 1) q^{r-\omega} + (q-1)(r-\omega). \end{aligned}$$

In particular

$$\begin{aligned} A(q, r, 1) &= (q-1)(r-1), \\ A(q, r, 2) &= (q-1)q^{r-2} + (q-1)(r-2), \\ A(q, r, 3) &\leq (3q^2 - 3)q^{r-3}/4 + (q-1)(r-3). \end{aligned}$$

Hence

$$A(q, r, 2) - A(q, r, 1) = (q-1)q^{r-2} - (q-1) \geq 0$$

for $r \geq 2$. Similarly,

$$\begin{aligned} &A(q,r,2) - A(q,r,3) \\ &\quad \geq \; ((q^2-q) - (3q^2-3)/4)q^{r-3} + (q-1) \\ &\quad = \; (q-1)(q-3)q^{r-3}/4 + (q-1) > 0 \end{aligned}$$

for $r \geq 3$,

$$\begin{aligned} &A(q,r,3) - A(q,r,4) \\ &\quad \geq \; (q(3q^2/4-1) - ((2q^3+q)/3-1))q^{r-4} + (q-1) \\ &\quad = \; (q^3-16q+12)q^{r-4}/12 + (q-1) > 0 \end{aligned}$$

for $r \geq 4$,

$$\begin{aligned} &A(q,r,4) - A(q,r,5) \\ &\quad \geq \; (q-1)(13q^3+13q^2+27q-165)q^{r-5}/192 + (q-1) > 0 \end{aligned}$$

for $r \geq 5$,

$$\begin{aligned} &A(q,r,5) - A(q,r,6) \\ &\quad \geq \; (47q^5-140q^3-1152q+960)q^{r-6}/960 + (q-1) > 0 \end{aligned}$$

for $r \geq 6$. □

This shows that for $\omega \leq 6$, the maximum is obtained for $\omega = 2$ and that for $\omega \geq 2$, the value of $A(q,r,\omega)$ decreases with ω. We conjecture that this is true also for $\omega > 6$.

7.3 The probability of undetected error

The probability of undetected error depends on the channel model as well as the code and the probability distribution of the sent arrays. For $q = 2$, the asymmetric channel was considered and the probability of undetected error was determined (assuming that errors are independent and the transition probability is the same for all symbols). For $q > 2$ there are more than one possible asymmetric channel and we consider a general model. We assume that the errors are independent. In a particular position, if the sent symbol is a, then the received symbol is b with probability $\pi(b|a)$. For any asymmetric model we have $\pi(b|a) = 0$ for $b > a$. Further, $\sum_{b=0}^{a} \pi(b|a) = 1$. Independent errors mean that if $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ is sent, then $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ is received with probability

$$\pi(\mathbf{b}|\mathbf{a}) = \prod_{i=1}^{n} \pi(b_i|a_i).$$

If $P(\mathbf{x})$ denotes the probability distribution of the information arrays, and $P(\mathbf{x}, \pi)$ is the probability that we get an undetected error when $\mathbf{x}$ is encoded and transmitted, then the probability of undetected error for the code is $P(C, \pi) = \sum_{\mathbf{x} \in F_q^k} P(\mathbf{x}) P(\mathbf{x}, \pi)$.

We note that if $\mathbf{x}$ and $\mathbf{x}'$ are equivalent (that is, $\mathbf{x}'$ is a permuted version of $\mathbf{x}$), then $c(\mathbf{x}') = c(\mathbf{x})$ and $P(\mathbf{x}', \pi) = P(\mathbf{x}, \pi)$. Hence, if we let

$$\mathbf{v}(\mathbf{x}) = (v_0, v_1, \ldots, v_{q-1}),$$

where $v_j = \#\{i \mid x_i = j\}$, then $P(\mathbf{x}, \pi)$ only depends on $\mathbf{v} = \mathbf{v}(\mathbf{x})$ and π; we denote it by $\Pi(\mathbf{v}, \pi)$. With this notation,

$$P(C, \pi) = \sum_{\mathbf{v}} \Pi(\mathbf{v}, \pi) \sum_{\substack{\mathbf{x} \in F_q^k \\ \mathbf{v}(\mathbf{x}) = \mathbf{v}}} P(\mathbf{x}).$$

From Theorem 7.2.1 we immediately get the following explicit expression for $\Pi(\mathbf{v}, \pi)$.

Theorem 7.3.1 *For any given* $\mathbf{v} = (v_0, v_1, \ldots, v_{q-1})$*, let*

$$\mathbf{x} = (\overbrace{0, 0, \ldots, 0}^{v_0}, \overbrace{1, 1, \ldots, 1}^{v_1}, \ldots, \overbrace{q-1, q-1, \ldots, q-1}^{v_{q-1}})$$

and $u = u(\mathbf{x}) = \sum_{j=0}^{q-1} (q-1-j) v_j$*. Then*

$$\Pi(\mathbf{v}, \pi) = \sum_{\substack{\lambda \\ \langle \lambda \rangle_\rho \subseteq \langle [u]_\theta \rangle_\rho}} \sum_{j \geq 1} \sum_{\substack{\mathbf{y} \subset \mathbf{x} \\ u(\mathbf{y}) = j\mu - \lambda + u(\mathbf{x})}} \pi(\mathbf{y}|\mathbf{x}).$$

Chapter 8

Summary and some open problems

This work is based on symmetric and non-symmetric channels and the capability of the codes, using over these channels, to correct and/or detect errors.

For symmetric codes we considered codes which can detect errors. The error detection is very important for the correct transmission of a message. If we use the definitions for a code to be good or proper for error detection, given in Chapter 2, the problem to determine if the code is good or proper by using the probability of undetected error, is equivalent to determine the weight or distance distribution of the code. But this is very hard problem to consider and the weight/distance distribution is known only for few codes. So we derived necessary conditions for codes to be good for error detection. The conditions depend on the minimum distance d_{min}, the size M of the code and a lower bound A of A_d. When we have these values, we can consider a function $h(p)$, which is a function in terms of the symbol probability. The condition says that if the length of the code is greater than this function $h(p)$, then the code is ugly for error detection. We think that this is easier way to determine if the code is good or not good for error detection than to consider the probability of undetected error $P_{ue}(C, p)$. We gave some examples to show how good our approximation is.

To determine if a code is proper for error detection is again a very hard computational problem because again it depends on the probability of undetected error $P_{ue}(C, p)$, which means on weight/distance distribution of the code, if we use the definition for proper code in Chapter 2. Large codes are proper and here we determined how large is large. We proved that any q-ary code of size at least

$q^n - \beta(q,n)$, where $\beta(q,n)$ is a function depending on the length of the code n and the alphabet size q, is proper. Again this method is much more easier to determine if the code is proper or not than to consider the weight/distance distribution and the probability of undetected error $P_{ue}(C,p)$.

We presented a function $S(m,n)$, which is the number of sequences of length m and weight n. Some very usefull properties of this function were derived. With the help of this properties a theorem for generalized Bose-Lin codes was proven.

The error correction capability of a code is also very important during the transmission. We constructed some binary and ternary codes which can correct up to t symmetric errors and detect all unidirectional errors. Our codes are optimal in terms of the the shortest length of a code to be t-EC-AUED code. A generalization $GBT_q(n,T)$ for arbitrary alphabet size q is made of the lower bound of the length of a $(T-1)$-EC-AUED code, which is used during the construction. In many cases it appears that $n = GBT_q(n,T)$ and in some other cases the length is 1 above the bound.

Bose and Lin introduced a class of symmetric binary codes which can detect binary asymmetric errors. In this work based on this class of codes we derived a class of generalized Bose-Lin codes for arbitrary alphabet size q. The class is described. We also gave a result which says when there are undetectable errors and the minimum weight of the undetected error is found. We considered the maximum weight of the detected errors and we concluded that the function which presents the maximum weight of the detected errors is getting its maximum when $\omega = 2$, where $0 \leq \omega \leq r$ and r is the number of check symbols. We showed this for $\omega \leq 6$ and we conjectured that this is true for $\omega > 6$.

The first open problem, which we leave for a future work, is to prove that the maximum weight of the detected errors, using generalized Bose-Lin codes, is maximal when the number of the check symbols are 2 or more. We have proven this for $\omega \leq 6$, where $0 \leq \omega \leq r$ and r is the number of check symbols. The problem is to prove it for $\omega > 6$.

Another open problem is to construct more $(T-1)$-EC-AUED codes with bigger parameters - alphabet size, length and size. The case for the ternary $(T-1)$-EC-AUED codes of size 7 is still open. It is not determined if there is or not a code such that $n_3(7,8) = 21$.

As it was mentioned earlier, we showed that all large codes are proper for error detection and we gave bounds on how large. The first problem is to find better bounds on the size of code to be proper. Interesting questions are what is happen with the codes which are not so large and when they are proper. Is it possible to find some other conditions for these codes which show if they are proper or not for error detection, excluding the condition using the weight/distance distribution and the probability of undetected error $P_{ue}(C, p)$?

Bibliography

[1] K. A. S. Adbel-Ghaffar. A simple derivation of the undetected error probabilities of complementary codes. *IEEE Trans. Inform. Theory*, 50:861–862, 2004.

[2] A. Ahlswede and H. Aydinian. Error control codes for parallel asymmetric channels. In *Proceedings of the IEEE Symposium on Information Theory*, pages 1768–1772, Seattle, WA, July 2006.

[3] M. Blaum. *Codes for Detecting and Correcting Unidirectional Errors.* IEEE Computer Society Press, 1993.

[4] F. J. H. Boinck and H. C. A. van Tilborg. Constructions and bounds for systematic tEC/AUED codes. *IEEE Trans. Inform. Theory*, 36:1381–1390, 1990.

[5] B. Bose, S. Elmougy, and L. G. Tallini. Systematic t-unidirectional error-detecting codes in Z_m. unpublished manuscript, 2005.

[6] B. Bose and T. Kløve. Sperner's theorem and unidirectional codes. In *P. Charpin and Ø. Ytrehus (eds.), Workshop on Coding and Cryptography 2005*, pages 169–175, Bergen, Norway, March 14-18 2005.

[7] B. Bose and D. J. Lin. Systematic unidirectional error-detecting codes. *IEEE Trans. Comp.*, 34:1026–1032, 1985.

[8] B. Bose and D. K. Pradhan. Optimal unidirectional error detecting/correcting codes. *IEEE Trans. Comp.*, 31:564–568, 1982.

[9] B. Bose and T. R. N. Rao. Theory of unidirectional error correcting/detecting codes. *IEEE Trans. Comp.*, 31:23–32, 1982.

[10] N. G. de Bruijn, C. van Ebbenhorst Tengbergen, and D. Kruyswijk. On the set of divisors of a number. *Nieuw Archief voor Wiskunde (2)*, 23:191–193, 1951.

[11] R. Dodunekova and S. Dodunekov. Sufficient conditions for good and proper error-detecting codes. *IEEE Trans. Inform. Theory*, 543:2023–2026, 1997.

[12] S. Elmougy. *Some Contributions to Asymmetric Control Codes.* PhD thesis, Oregon State University, Corvallis, 2005.

[13] F. W. Fu and T. Kløve. The complement of binary linear codes for error detection. In *Proceedings of the 2002 IEEE Information Theory Workshop*, page 187, Bangalore, India, October 20-25 2002.

[14] F. W. Fu, T. Kløve, and V. K. Wei. On the undetected error probability for binary codes. *IEEE Trans. Inform. Theory*, 49:382–390, 2003.

[15] I. Gancheva and T. Kløve. Codes for error detection, good or not good. In *Book of abstracts of the 2005 IEEE International Symposium on Information Theory*, page 63, Adelaide, South Australia, September 4-9 2005.

[16] I. Gancheva and T. Kløve. Construction of some optimal t-EC-AUED codes. In *Proceedings of the Fourth International Workshop on Optimal Codes and Related Topics*, pages 152–156, Pamporovo, Bulgaria, June 17-23 2005.

[17] I. Gancheva and T. Kløve. Generalized Bose-Lin codes, a class of codes detecting asymmetric errors. In *Proceedings of the Fourth International Workshop on Optimal Codes and Related Topics*, pages 157–162, Pamporovo, Bulgaria, June 17-23 2005.

[18] R. L. Graham and N. J. A. Sloane. Lower bounds for constant weight codes. *IEEE Trans. Inform. Theory*, 27:37–43, 1980.

[19] T. Kløve and V. I. Korzhik. *Error Detecting Codes, General Theory and Their Application in Feedback Communication Systems.* Kluwer Academic Publishers, Boston, 1995.

[20] T. Kløve, P. Oprisan, and B. Bose. The probability of undetected error for a class of asymmetric error detecting codes. *IEEE Trans. Inform. Theory*, 51:1202–1205, 2005.

[21] S. K. Leung-Yan-Cheong, E. R. Barnes, and D. U. Friedman. Some properties of undetected error probability of linear codes. *IEEE Trans. Inform. Theory*, 25:110–112, 1979.

[22] D. J. Lin and B. Bose. On the maximality of group theoretic SEC-AUED codes. In *R. Capocelli (ed.), Sequences: Combinatorics, Security and Transmission*, pages 506–529, Springer-Verlag, New York, 1990.

[23] S. Lin and D. J. Costello, Jr. *Error Control Coding.* Pearson Education, Inc., USA, 2004.

[24] P. A. MacMahon. *Combinatory Analysis*, volume 1. Chelsea, New York, 1960.

[25] I. Naydenova and T. Kløve. Necessary conditions for codes to be good for error detection. In *Book of abstracts of the Annual Workshop of Coding Theory and Applications*, page 24, Bankya, Bulgaria, December 15-18 2005.

[26] I. Naydenova and T. Kløve. A bound on q-ary t-EC-AUED codes and constructions of some optimal ternary t-EC-AUED codes. In *Book of abstracts of the 2006 International Symposium on Information Theory and its Applications*, page 3, Seoul, South Korea, October 29 - November 1 2006.

[27] I. Naydenova and T. Kløve. Large proper codes for error detection. In *Proceedings of the 2006 IEEE Information Theory Workshop*, pages 170–174, Chengdu, China, October 22-26 2006.

[28] I. Naydenova and T. Kløve. Generalized Bose-Lin codes, a class of codes detecting asymmetric errors. *IEEE Trans. Inform. Theory*, 53:1188–1193, 2007.

[29] J. H. Weber. *Bounds and Constructions for Binary Block Codes Correcting Asymmetric or Unidirectional Errors.* PhD thesis, Delft University of Technology, The Netherlands, 1985.

[30] Z. Zhang and X. Xia. LYM inequalities for t-antichains. *Science in China (series A)*, 39:1009–1024, 1996.