Monotone Error Structure and Local Weight Distribution of Linear Codes

January 2008

Kenji YASUNAGA

Monotone Error Structure and Local Weight Distribution of Linear Codes

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2008

Kenji YASUNAGA

Related Publications by the Author

1. Journal Papers

1. Kenji Yasunaga and Toru Fujiwara, "Determination of the local weight distribution of binary linear block codes," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4444–4454, October 2006.

2. Kenji Yasunaga, Toru Fujiwara, and Tadao Kasami, "Local weight distribution of the (256, 93) third-order binary Reed-Muller code," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 3, pp. 698–701, March 2007.

2. Papers in Refereed Conferences

1. Kenji Yasunaga and Toru Fujiwara, "An algorithm for computing the local weight distribution of binary linear codes closed under a group of permutations," in *Proceedings of the 2004 International Symposium on Information Theory and Its Applications (ISITA2004)*, pp. 846–851, October 2004.

2. Kenji Yasunaga and Toru Fujiwara, "Relations between the local weight distributions of a linear block code, its extended code, and its even weight subcode," in *Proceedings of the 2005 IEEE International Symposium on Information Theory (ISIT2005)*, pp. 382–386, September 2005.

3. Kenji Yasunaga and Toru Fujiwara, "Correctable errors of weight half the minimum distance plus one for the first-order Reed-Muller codes," in *Proceedings of the 17th Symposium on Applied Algebra, Algebraic Algorithms, and Error Correcting Codes (AAECC-17), Lecture Notes in Computer Science*, Springer-Verlag, vol. 4851, pp. 110–119, December 2007.

3. Other Publications

1. Kenji Yasunaga and Toru Fujiwara, "An algorithm for computing the local distance profile of binary linear codes closed under a group of permutations," *IEICE Technical Report*, IT2003-47, pp. 37–41, September 2003.

2. Kenji Yasunaga and Toru Fujiwara, "The local weight distributions of the (128,50) extended binary primitive BCH code and the (128,64) Reed-Muller code," *IEICE Technical Report*, IT2004-19, pp. 7–12, July 2004.

3. Kenji Yasunaga and Toru Fujiwara, "Relations among the local weight distributions of a linear block code, its extended code and its even weight subcode," in *Proceedings of the 27th Symposium on Information Theory and Its Applications (SITA2004)*, pp. 559–562, December 2004.

4. Kenji Yasunaga and Toru Fujiwara, "The local weight distributions of transitive invariant codes and their punctured codes," in *Proceedings of the 2005 Hawaii, IEICE and SITA Joint Conference on Information Theory (HISC2005)*, May 2005, *IEICE Technical Report*, IT2005–14, pp. 79–84, May 2005.

5. Takahiro Yasuda, Kenji Yasunaga, and Toru Fujiwara, "Improvement of the Seguin lower bound using the local weight distribution," in *Proceedings of the 28th Symposium on Information Theory and Its Applications (SITA2005)*, pp. 435–438, November 2005.

6. Kenji Yasunaga and Toru Fujiwara, "Local weight distribution of the (256, 93) third-order binary Reed-Muller code," in *Proceedings of the 2006 Hawaii, IEICE and SITA Joint Conference on Information Theory (HISC2006)*, May 2006, *IEICE Technical Report*, IT2006–6, pp. 31–36, June 2006.

7. Kenji Yasunaga and Toru Fujiwara, "Correctable errors of weight half the minimum distance for the first-order Reed-Muller codes," in *Proceedings of the 29th Symposium on Information Theory and Its Applications (SITA2006)*, pp. 5–8, November 2006.

8. Kenji Yasunaga and Toru Fujiwara, "On trial set and uncorrectable errors for the first-order Reed-Muller codes," in *Proceedings of the 2007 Hawaii and SITA Joint Conference on Information Theory (HISC2007)*, pp. 67–72, May 2007.

9. Kenji Yasunaga and Toru Fujiwara, "Minimum weight codewords in trial sets," in *Proceedings of the 30th Symposium on Information Theory and Its Applications (SITA2007)*, pp. 562–564, November 2007.

Abstract

Error correcting codes are essential tools for reliable communication on noisy channels. In this dissertation, the error correction capability of codes is studied. The number of error bits guaranteed to correct is less than half the minimum distance of the code. However, many errors can be corrected even when the number of error bits is beyond half the minimum distance. Analyzing the error correction capability for this case is significant to recognize the limitations of codes. The performance of codes on the probabilistic channel models, such as a binary symmetric channel and an additive white Gaussian noise channel (AWGNC), is analyzed by the error probability. The error probabilities on these channel are usually given by upper and lower bounds that use the weight distribution of the code.

The first part of this dissertation investigates the error correction capability beyond half the minimum distance. The *monotone error structure* is mainly used for the analysis. This structure is known for long, but there were only a little studies on it. Helleseth, Kløve, and Levenshtein used this structure for the analysis of the error correction capability beyond half the minimum distance and introduced useful concepts: *larger halves* and *trial sets*. In this work, for the first-order Reed-Muller codes, the explicit expressions are derived for the number of correctable errors of weight half the minimum distance and half the minimum distance plus one. For general linear codes that satisfy some condition, a lower bound on the number of uncorrectable errors is derived. The condition is satisfied by some primitive BCH codes, some extended primitive BCH codes, long Reed-Muller codes, and random linear codes. The monotone structure, larger halves, and trial sets play a significant role to derive the results.

The second part of the dissertation studies methods of determining the *local weight distribution* of linear codes. It is known that the local weight distribution gives tighter upper bounds on the error probability over AWGNC than the bounds obtained by using the weight distribution. In this work, two approaches are considered: theoretical one and computational one. As a theoretical approach, the relations between the local weight distributions of a code, its extended code, and its even weight subcode are investigated. It is shown that, for some Reed-Muller codes and extended primitive BCH codes, the local weight distributions of the corresponding punctured codes are straightforwardly determined from those of them. As a computational approach, an algorithm for computing the local weight distributions is proposed. This algorithm is effective for codes whose automorphism group is large. Reed-Muller codes and extended primitive BCH codes have large automorphism groups. Using the algorithm and the relation, the local weight distributions are determined for some Reed-Muller codes, punctured Reed-Muller codes, extended primitive BCH codes, primitive BCH codes, and even weight subcodes of punctured Reed-Muller codes and primitive BCH codes.

Chapters of the dissertation are organized as follows. Chapter 1 introduces the problems studied in the dissertation and summerizes the results. Chapter 2 describes the definitions and properties of error correcting codes. Chapter 3 investigates the monotone error structure of the first-order Reed-Muller codes. Chapter 4 uses trial sets for the analysis of the error correction capability. For determination of the local weight distributions, Chapter 5 takes a theoretical approach and Chapter 6 takes a computational approach. Chapter 7 concludes the dissertation.

Contents

Acknowledgments

First I am deeply grateful to Professor Toru Fujiwara, my supervisor, for his insightful guidance, helpful advice, and continuous support throughout the course of my graduate studies at Osaka University. I would like to thank the late Professor Emeritus Tadao Kasami, who collaborated with me on the research described in Section 6.5.3, for his sharp insight and humble attitude to the research.

I would also like to thank the committee members of my thesis: Professors Norihisa Komoda, Shinji Shimojo, Shojiro Nishio, Fumio Kishino and Associate Professor Yasunori Ishihara. Their constructive comments and suggestions improved the quality of the thesis.

I wish to thank to Associate Professor Yasunori Ishihara, Assistant Professor Maki Yoshida, and all members in Information Security Engineering Laboratory for their insightful advice, encouragement, and inspiring conversation. I would also like to thank to the following people with whom I had useful discussion and cheerful conversations: Drs. Jun Asatani, Masanori Hirotomo, Masami Mohri, and Hideki Yagi.

Finally I would like to express my great appreciation to my wife, my parents, my grandparents, and my brothers. I could not complete this thesis without their support and encouragement.

Chapter 1

Introduction

1.1 Error Correcting Codes

In a scenario of error correcting codes, the sender wishes to send a message to the receiver, but the channel they can use may cause errors in the message. One strategy to tolerate errors is to add redundancy to the message. Then if the message is corrupted in transmission, the corrupted message may still retain the information about the original message. The receiver can recover the original message *only* from the corrupted message. A description of how to add redundancy is an error correcting code.

Today an error correcting code is a basic technology to achieve reliable communication and storage. We can use it where the system considered allows place for the devices and time for adding redundancy (coding) and recovering from errors (decoding). Since many coding and decoding methods have been developed so far, system designers are required to consider which coding and decoding method is proper for their system.

Although there are many codes with good error performance, such as BCH codes, Reed-Solomon codes, LDPC codes, together with efficient decoding algorithms, analyzing their error performance for typical channel models with an optimal decoding algorithm involves some difficulties. Such an analysis reveals the limitation of the code and is important for ones developing sub-optimal (and efficient) decoding algorithms for them.

One of the basic criteria for the error performance is the *minimum distance* d of the code. The minimum distance of the code is the minimum Hamming distance between two distinct coded messages (codewords) in the code. If the number of corrupted positions in the received message is less than $d/2$, the receiver can *always* correct errors. Therefore, code with larger minimum distance lead to better error performance.

The $d/2$ bound is the worst-case error correctability. Namely, there exists at least one uncorrectable error pattern that happens at $d/2$ positions and the receiver cannot correctly recover. We expect to correct many errors for the case the number of corrupted positions is beyond this bound. To characterize the error correction capability for this case is significant to recognize the limitations of codes. We know empirically that many of them can be corrected. However, analysis for beyond $d/2$ is known to be a difficult task and there is little work and analysis for specific practical codes, such as BCH codes, and Reed-Muller codes.

Although the minimum distance is one of the important criteria for error performance of codes, the error performance is determined by the error probability for probabilistic channels. The most fundamental and practically important probabilistic channel models are binary symmetric channels (BSC) and additive white Gaussian noise channels (AWGNC). The error probability is defined as the probability the receiver fails to correct errors, and is determined by a channel, a code, and a decoder. In our work we consider maximum likelihood (ML) decoders, which mean optimal decoders for the channel. The analysis of the error performance with ML decoder is significant for recognizing the limitation of the code and useful for sub-optimal decoding designers. For symmetric channels, such as BSC and AWGNC, a minimum distance decoder is an ML decoder.

The exact error probability for BSC can be obtained if we know the exact numbers of correctable errors for each number of corrupted positions, which is equivalent to the weight distribution of the coset leaders of the codes. The naive algorithm for computing the weight distribution of the coset leaders requires $2^{O(n)}$ time, where n is the length of codewords (the code length). Therefore, it is difficult to compute it by the naive algorithm even for moderate code length, say $n \geq 128$.

Since deriving the exact error probability is intractable, upper and lower bounds on the probability are used as alternatives. There are two ways for deriving the bounds on the error probability: the bounds on the number of correctable errors and the bounds on the probability itself. To analyze and bound the number of correctable errors, the *monotone error structure* is a useful concept. Also, for the bounds on the error probability over AWGNC, the *local weight distribution (LWD)* of the code is a good alternative for the (global) weight distribution of the code, which is usually and mostly used for giving the bounds.

In this dissertation, the monotone error structure and the local weight distributions are investigated. We will describe them below.

1.2 Monotone Error Structure

In minimum distance decoding, the decoder finds the nearest codeword to the received vector. If there are two or more codewords nearest to the received vector, the decoder can choose any of them. That is, the error probability does not change depending on the choice of correctable errors. If we decided to correct the lexicographically smallest error, then correctable errors and uncorrectable errors have the *monotone structure*. The monotone structure is the following property: if $\boldsymbol{x}$ is a correctable error then the vector covered by $\boldsymbol{x}$ is also a correctable error, and if $\boldsymbol{x}$ is an uncorrectable error then the vector that covers $\boldsymbol{x}$ is also an uncorrectable error, where we say the vector $\boldsymbol{x}$ covers the vector $\boldsymbol{y}$ if $x_i \geq y_i$ for every coordinate i. Although this structure has been known for long (for example, it is seen in a classical textbook [31, Theorem 3.11]), there was little work using it.

Zémor [50] showed that, using the monotone structure, the error probability of binary linear codes over BSC after ML decoding has a threshold behavior.

Helleseth, Kløve, and Levenshtein [19] gave an asymptotic analysis of the error performance beyond $d/2$. They mainly used the monotone structure of errors in their analysis, and thus showed that the monotone structure is useful for error performance analysis beyond $d/2$. The key ingredients of their analysis is *larger half* and *trial set*.

If the correctable and uncorrectable errors have the monotone structure, they are characterized by the maximum correctable and minimal uncorrectable errors. If we know the set of minimal uncorrectable errors, the entire uncorrectable errors are determined uniquely. Larger half is introduced for characterizing the minimal uncorrectable errors; Larger halves of all codewords except all-zero codewords contains the minimal uncorrectable errors. Helleseth et al. clarified the structure of larger halves of codewords.

A trial set for the code is defined as the set of codewords whose larger halves contains the minimal uncorrectable errors. There are two applications of a trial set: giving an upper bound on the number of uncorrectable errors and a minimum distance decoding. For both applications, a smaller trial set is desirable. Therefore investigation of *minimum* trial sets is significant.

1.3 Local Weight Distribution

To derive the exact value of error probability is difficult for most practical codes. Therefore, we estimate the probability by upper and lower bounds. For linear codes, many bounds are proposed so far and many of them use the *weight distribution* (also referred to as weight spectrum or distance profile) of the code. For details of various bounds, see a survey paper [35]. To determine the weight distribution of linear codes is a difficult task and is one of the central problems in coding theory. Recently it has been reported that the *local weight distribution* of codes can be used to give more accurate bounds on the error probability over AWGNC [1, 17].

The local weight distribution is defined as the weight distribution of *minimal codewords*. The brute-force algorithm for computing the local weight distribution requires $O(n^2k2^k)$ time, where n is the code length and k is the length of original messages (the dimension). On the contrary, that of the weight distribution requires $O(n2^k)$ time.

In [3] the local weight distributions of the Hamming codes, the extended Hamming codes, the second-order Reed-Muller codes, and long random linear codes are derived. In [28] an algorithm for computing the local weight distributions of cyclic codes are proposed and the distributions are determined for the BCH codes of length 63. An algorithm in [28] uses an invariance property of minimality under cyclic permutations. Since the size of cyclic permutations is $O(n)$, the algorithm reduced the time complexity to $1/n$ of that of the brute-force algorithm.

1.4 Contributions and Dissertation Structure

In Chapter 2, before presenting the results, we give a brief description of linear codes including code modification techniques and typical linear codes we will use.

In Chapters 3 and 4, the error correction capabilities of linear codes beyond half the minimum distance $d/2$ are investigated. The *monotone error structure* is a main ingredient for the analysis. Analysis for the first-order Reed-Muller codes is done in Chapter 3. Analysis using trial sets for general linear codes is done in Chapter 4. The results in Sections 3.4, 3.5,and 3.6 have appeared in [44], [47], and [45], respectively. The results in Chapter 4 have been presented in [46].

For the first-order Reed-Muller codes, we determine the numbers of correctable errors of weights $d/2$ and $d/2 + 1$. We also determine the weight distribution of the minimal uncorrectable errors. The first-order Reed-Muller code is a very old and simple code. However, determining the exact number of correctable errors beyond $d/2$ is a difficult task because the rate of the code is low and thus there are many correctable errors.

For general linear codes, we give bounds on the size of minimum trial sets. The weight distribution of trial sets leads to an upper bound on the number of uncorrectable errors, and this bound is tight if a given trial set is small. Therefore, bounding the size of minimum trial sets yields good bounds on the number of correctable/uncorrectable errors. We also give a condition under which all minimum weight codewords are in trial sets. If the condition holds, we can derive a lower bound on the number of uncorrectable errors of weight $d/2$. We show that long Reed-Muller codes and long random linear codes meet the condition. In particular, for Reed-Muller codes with fixed order, the corresponding upper and lower bounds come close.

Next we consider determining the local weight distributions for linear codes. Two approaches are studied: a theoretical approach in Chapter 5 and a computational approach in Chapter 6. The results in Section 6.5.3 have appeared in [48]. The rest of the results in Chapters 5 and 6 have been presented in [43].

As a theoretical approach, we study relations between the local weight distributions of a code, its extended code, and its even weight subcode. We derive the way to determine the local weight distributions of the extended code and the even weight subcode from that of the original code. We also show the way to determine the local weight distribution from its extended code in the case the extended code is a transitive invariant code. To determine the local weight distributions using above three ways, we are required to know the number of *only-odd-decomposable* codewords. However, we give a simple sufficient condition under which there is no only-odd decomposable codeword in the code: the code has only codewords of weight multiples of four. This condition holds for the Reed-Muller code of length greater than or equal to 128 and the extended primitive BCH codes of length 128 and the dimension less than or equal to 57.

As a computational approach, we propose an algorithm for computing the local weight distribution of a given code using the automorphism group of the code. This algorithm is effective for a code with large automorphism group. The size of automorphism group of Reed-Muller codes is $2^{O(n \log n)}$, that of extended primitive BCH codes is $O(n^2)$, that of cyclic codes is $O(n)$. Therefore, our algorithm is effective for Reed-Muller codes and extended primitive BCH codes.

Using the proposed algorithm, we determine the local weight distributions of the extended primitive BCH codes of length 128, dimensions $50, 43, 36$, and the third-order Reed-Muller codes of length 128 and 256. From the relations derived in Chapter 5 we also determine the local weight distributions of corresponding punctured Reed-Muller codes and primitive BCH codes. The list of codes whose local weight distributions are determined is shown in Table 1.1.

Chapter 7 concludes the dissertation with a summary of the work and directions of future work.

Table 1.1: The list of codes whose local weight distributions are determined.

Code	Code length	Dimension	Reference
Hamming code	all	all	[3]
Extended Hamming code	all	all	[3]
Second-order Reed-Muller code	all	all	[3]
Random linear code	∞	all	[3]
Primitive BCH code	63	18, 24, 30, 36, 39, 45	[28]
Primitive BCH code	63	51, 57	[29]
Primitive BCH code	127	36, 43, 50	This work
Extended primitive BCH code	128	36, 43, 50	This work
Even weight subcode of primitive BCH code	127	35, 42, 49	This work
Third-order Reed-Muller code	128	64	This work
Third-order Reed-Muller code	256	93	This work
Punctured Reed-Muller code	127	64	This work
Punctured Reed-Muller code	255	93	This work
Even weight subcode of punctured Reed-Muller code	127	63	This work
Even weight subcode of punctured Reed-Muller code	255	92	This work

Chapter 2

Preliminaries

This chapter provides definitions and properties of linear codes. The basics of linear codes including the monotone structure and the local weight distribution are presented in Section 2.1. Several code modification techniques and code families used in our work are shown in Sections 2.2 and 2.3, respectively. Since codes we consider in our work are binary codes, we define codes over binary alphabet.

2.1 Basic Definitions and Properties

2.1.1 Linear Codes

Let $\mathbb{F} = \{0, 1\}$ be a finite field of size two and $\mathbb{F}^n = \{0, 1\}^n$ be a binary vector space of dimension n. An error correcting code C is a subset of $\mathbb{F}^n$. An element $\boldsymbol{c} \in C$ is called a *codeword* of C. In transmission with an error correcting code C, the sender chooses a codeword in C and sends it to the receiver. Then n is the *code length* of C and $\log_2 |C|$ is the *dimension* of C. The *minimum distance* d of C is defined as the minimum Hamming distance between distinct codewords in C. That is,

$$d = \min_{\substack{\boldsymbol{c}_1, \boldsymbol{c}_2 \in C \\ \boldsymbol{c}_1 \neq \boldsymbol{c}_2}} d_H(\boldsymbol{c}_1, \boldsymbol{c}_2),$$

where $d_H(\boldsymbol{x}, \boldsymbol{y})$ is the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$.

If C is a linear subspace of $\mathbb{F}^n$, C is called a *linear code*. For a linear code C, the minimum distance d is equal to the minimum Hamming weight of codewords in C. That is,

$$d = \min_{\substack{\boldsymbol{c}_1, \boldsymbol{c}_2 \in C \\ \boldsymbol{c}_1 \neq \boldsymbol{c}_2}} d_H(\boldsymbol{c}_1, \boldsymbol{c}_2) = \min_{\boldsymbol{c} \in C \setminus \{\boldsymbol{0}\}} w(\boldsymbol{c}),$$

where $w(\boldsymbol{x})$ is the Hamming weight of $\boldsymbol{x}$, which is equal to $d_H(\boldsymbol{x}, \mathbf{0})$. A linear code C of code length n, dimension k, and minimum distance d is referred to as (n, k, d) code C or simply (n, k) code C. Unless otherwise stated, all codes we use in this dissertation are linear codes.

Let G be a $k \times n$ matrix over $\mathbb{F}$. Then G is called a *generator matrix* of C if a linear span of G equals C. That is, if the k rows of G are $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k \in \mathbb{F}^n$, then every $\boldsymbol{c} \in C$ can be written as

$$\boldsymbol{c} = a_1\boldsymbol{g}_1 + a_2\boldsymbol{g}_2 + \cdots + a_k\boldsymbol{g}_k,$$

where $a_i \in \mathbb{F}$ for $1 \leq i \leq k$. The number of linearly independent rows of G corresponds to the dimension of C.

Let H be an $(n - k) \times n$ matrix over $\mathbb{F}$. Then H is called a *parity check matrix* of C if its kernel equals C. That is, $\boldsymbol{c} \in C$ if and only if

$$H\boldsymbol{c}^T = \mathbf{0}.$$

2.1.2 Channel, Error Probability, and Decoding

We introduce two channel models: binary symmetric channel (BSC) and additive white Gaussian noise channel (AWGNC). Both channels are binary input memoryless symmetric channels.

In BSC, each transmitted bit is independently flipped with a fixed probability p, where $0 \leq p < 1/2$. When $\boldsymbol{c} \in C$ is transmitted over BSC, the received vector $\boldsymbol{y} \in \mathbb{F}^n$ is represented as

$$\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e},$$

where $\boldsymbol{e} = (e_1, e_2, \ldots, e_n) \in \mathbb{F}^n$ and $e_i = 1$ with probability p. The vector $\boldsymbol{e}$ is called an *error vector*. The weight of $\boldsymbol{e}$ indicates the number of corrupted bits in $\boldsymbol{y}$. The probability p is called a cross over probability of BSC.

In AWGNC, a white Gaussian noise is added to each transmitted bit. We assume transmitted sequences take real values. Therefore, we need to map a binary vector to a real-valued sequence. We usually use the following mapping function $s : \mathbb{F} \to \mathbb{R}$ such that $s(0) = 1$ and $s(1) = -1$. Thus a codeword $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in C$ is transmitted as $(s(c_1), s(c_2), \ldots, s(c_n))$. Then when $\boldsymbol{c} \in C$ is transmitted over AWGNC, the received sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$ is represented as, for every i with $1 \leq i \leq n$,

$$y_i = s(c_i) + z_i,$$

where z_i is a Gaussian random variable with zero mean and variance σ^2. The probability density function of z_i is

$$f_{z_i}(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z^2}{2\sigma^2}}.$$

The error probability P_e is defined as the probability that the receiver fails to decode correctly.

$$P_e = \Pr\left[\bigcup_{\substack{\boldsymbol{c}_1,\boldsymbol{c}_2 \in C \\ \boldsymbol{c}_1 \neq \boldsymbol{c}_2}} \mathcal{E}_{\boldsymbol{c}_1,\boldsymbol{c}_2}\right] = \Pr\left[\bigcup_{\boldsymbol{c} \in C\setminus\{\boldsymbol{0}\}} \mathcal{E}_{\boldsymbol{0},\boldsymbol{c}}\right],$$

where $\mathcal{E}_{\boldsymbol{c}_1,\boldsymbol{c}_2}$ denotes an event that a codeword $\boldsymbol{c}_1 \in C$ is transmitted and the decoding result of the receiver is a codeword $\boldsymbol{c}_2 \in C$. The second equality holds if C is a linear code. Note that P_e depends on a code, a decoding, and a channel model.

A decoding function $D : \mathbb{K} \to C$ is a function that maps a received vector to a codeword in C. A field $\mathbb{K}$ depends on the channel model; $\mathbb{K} = \mathbb{F}$ for BSC and $\mathbb{K} = \mathbb{R}$ for AWGNC. For memoryless channels such as BSC and AWGNC, a decoding that minimize P_e is called *maximum likelihood (ML) decoding*. For symmetric channels such as BSC and AWGNC, a *minimum distance decoding* is a ML decoding. A minimum distance decoding function is

$$D(\boldsymbol{y}) = \arg\min_{\boldsymbol{c} \in C} d(\boldsymbol{y}, \boldsymbol{c}),$$

where $\boldsymbol{y}$ is a received vector, a codeword $\boldsymbol{c}$ is mapped by the function s if needed, and the distance function $d(\cdot,\cdot)$ depends on a channel model; For BSC the Hamming distance is employed and for AWGNC the Euclidean distance is employed.

2.1.3 Coset, Syndrome Decoding, and Correctable Errors

Let $C \subseteq \mathbb{F}^n$ be an (n, k, d) linear code. Then $\mathbb{F}^n$ is partitioned into 2^{n-k} *cosets* of C, denoted by $C_1, C_2, \ldots, C_{2^{n-k}}$;

$$\mathbb{F}^n = \bigcup_{i=1}^{2^{n-k}} C_i \quad \text{and} \quad C_i \cap C_j = \emptyset \quad \text{for} \quad i \neq j,$$

where each $C_i = \{\boldsymbol{v}_i + \boldsymbol{c} : \boldsymbol{c} \in C\}$ with $\boldsymbol{v}_i \in \mathbb{F}^n$. The vector $\boldsymbol{v}_i$ is called the *coset leader* of the coset C_i. Every vector in C_i can be taken as $\boldsymbol{v}_i$.

Let H be a parity check matrix of C. The *syndrome* of a vector $\boldsymbol{v} \in \mathbb{F}^n$ is defined as

$$\boldsymbol{v}H^T.$$

All vectors having the same syndrome are in the same coset. Syndrome decoding associates an error vector to each syndrome[1]. The syndrome decoder presumes that the error vector added to the received vector $\boldsymbol{y}$ is the coset leader of the coset which contains $\boldsymbol{y}$. The syndrome decoding function $D : \mathbb{F}^n \to C$ is defined as

$$D(\boldsymbol{y}) = \boldsymbol{y} + \boldsymbol{v}_i \quad \text{if} \ \ \boldsymbol{y} \in C_i.$$

If each $\boldsymbol{v}_i$ has the minimum weight in the coset C_i, the syndrome decoder performs as a minimum distance decoder. Therefore, in what follows, we assume that a minimum weight vector is taken as the coset leader for every coset.

Let $E^0(C)$ be the set of all coset leaders of C. Then $E^0(C)$ is the set of the correctable errors and $E^1(C) = \mathbb{F}^n \setminus E^0(C)$ is the set of uncorrectable errors. Since there are 2^{n-k} cosets,

$$|E^0(C)| = 2^{n-k} \quad \text{and} \quad |E^1(C)| = 2^n - 2^{n-k}.$$

Define

$$E_i^0(C) = \{\boldsymbol{x} \in E^0(C) : w(\boldsymbol{x}) = i\},$$
$$E_i^1(C) = \{\boldsymbol{x} \in E^1(C) : w(\boldsymbol{x}) = i\}.$$

If the weight of the error vector is less than $\lceil d/2 \rceil$, the codeword nearest to the received vector is the transmitted codeword. We can always correct errors of weight less than $\lceil d/2 \rceil$. Thus

$$|E_i^0(C)| = \binom{n}{i} \quad \text{for} \quad 0 \le i \le \left\lfloor \frac{d-1}{2} \right\rfloor.$$

The error probability of C over BSC after ML decoding is given as

$$\sum_{i=1}^{n} p^i (1-p)^{n-i} |E_i^1(C)|,$$

where p is the cross over probability of BSC.

2.1.4 Monotone Error Structure

It is known that if we take a lexicographically smallest minimum weight vector as the coset leader for every coset, then the correctable and uncorrectable errors (in the syndrome

[1] Here we consider only discrete channels, such as BSC.

decoding) have the monotone structure. Namely, the coset leader is the smallest vector in each coset with respect to the following total ordering $\preceq$:

$$\boldsymbol{x} \preceq \boldsymbol{y} \quad \text{if and only if} \quad \begin{cases} w(\boldsymbol{x}) < w(\boldsymbol{y}), & \text{or} \\ w(\boldsymbol{x}) = w(\boldsymbol{y}) & \text{and} \quad v(\boldsymbol{x}) \leq v(\boldsymbol{y}), \end{cases}$$

where $v(\boldsymbol{x})$ denotes the numerical value of $\boldsymbol{x}$:

$$v(\boldsymbol{x}) = \sum_{i=1}^{n} x_i 2^{n-i}.$$

The relation $v(\boldsymbol{x}) < v(\boldsymbol{y})$ means $\boldsymbol{x}$ is lexicographically smaller than $\boldsymbol{y}$. We write $\boldsymbol{x} \prec \boldsymbol{y}$ if $\boldsymbol{x} \preceq \boldsymbol{y}$ and $\boldsymbol{x} \neq \boldsymbol{y}$.

To describe the monotone structure, we introduce a partial ordering $\subseteq$ called "covering" such that

$$\boldsymbol{x} \subseteq \boldsymbol{y} \quad \text{if and only if} \quad S(\boldsymbol{x}) \subseteq S(\boldsymbol{y}),$$

where

$$S(\boldsymbol{v}) = \{i : v_i \neq 0\}$$

is the support of $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$. We write $\boldsymbol{x} \subset \boldsymbol{y}$ if $\boldsymbol{x} \subseteq \boldsymbol{y}$ and $\boldsymbol{x} \neq \boldsymbol{y}$.

The monotone error structure is the following property. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}^n$ with $\boldsymbol{x} \subseteq \boldsymbol{y}$. If $\boldsymbol{y}$ is a correctable error, then $\boldsymbol{x}$ is also correctable, and if $\boldsymbol{x}$ is uncorrectable, then $\boldsymbol{y}$ is also uncorrectable.

For two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}^n$ we write $\boldsymbol{x} \cap \boldsymbol{y}$ as the vector whose support is $\mathrm{S}(\boldsymbol{x}) \cap \mathrm{S}(\boldsymbol{y})$. Define the left most coordinate of $\boldsymbol{x}$ as $l(\boldsymbol{x}) = \min S(\boldsymbol{x})$.

2.1.5 Local Weight Distribution

For an integer i with $0 \leq i \leq n$ and $U \subseteq \mathbb{F}^n$ define

$$A_i(U) = \{\boldsymbol{v} \in U : w(\boldsymbol{v}) = i\}.$$

Then the (global) weight distribution of C is the $(n+1)$-tuple

$$(|A_0(C)|, |A_1(C)|, \ldots, |A_n(C)|).$$

The *local weight distribution* of C is defined as the weight distribution of *minimal codewords* in C. A codeword $\boldsymbol{c} \in C$ is called minimal if $\boldsymbol{c}$ is minimal with respect to covering $\subseteq$. Namely, $\boldsymbol{c} \in C$ is minimal if $\boldsymbol{c}' \subset \boldsymbol{c}$ for $\boldsymbol{c}' \in C$ implies $\boldsymbol{c}' = \boldsymbol{0}$. Let $L_i(C)$

be the set of minimal codewords of weight i in C. The local weight distribution is the $(n+1)$-tuple

$$(|L_0(C)|, |L_1(C)|, \ldots, |L_n(C)|).$$

We denote by C^* the set of minimal codewords in C.

2.1.6 Automorphism Group of Codes

A permutation of vector coordinate is a rearrangement of coordinates in the vector. Let $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be a set of coordinate of codewords in C. A coordinate permutation function $\pi : \{\alpha_1, \ldots, \alpha_n\} \to \{\alpha_1, \ldots, \alpha_n\}$ is a function such that $\bigcup_{\alpha \in \{\alpha_1, \ldots, \alpha_n\}} \pi(\alpha) = \{\alpha_1, \ldots, \alpha_n\}$. We abuse π as a vector permutation function such that $\pi((v_{\alpha_1}, v_{\alpha_2}, \ldots, v_{\alpha_n})) = (v_{\pi^{-1}(\alpha_1)}, v_{\pi^{-1}(\alpha_2)}, \ldots, v_{\pi^{-1}(\alpha_n)})$. We also abuse π as a vector permutation functions for a set U of vectors such that $\pi U = \bigcup_{\boldsymbol{v} \in U} \pi(\boldsymbol{v})$.

For example, let a permutation $\pi : \{1, 2, 3, 4\} \to \{1, 2, 3, 4\}$ such that $\pi(1) = 2, \pi(2) = 3, \pi(3) = 1, \pi(4) = 4$. For a vector $\boldsymbol{v} = (v_1, v_2, v_3, v_4)$, $\pi\boldsymbol{v} = (v_2, v_3, v_1, v_4)$. For a set $U = \{0011, 0111, 1010\}$, $\pi U = \{0101, 1101, 0110\}$.

An automorphism group of a code C is a set of permutations that permute C into C itself. Formally,

$$\mathrm{Aut}(C) = \{\pi : \pi C = C\}.$$

One can verify that $\mathrm{Aut}(C)$ forms a group.

Automorphism group of codes is a key property for our algorithms presented in Chapter 6.

2.2 Code Modifications

We can change the parameters of a code by simply modifying the code. Let C be an (n, k, d) linear code, G be a generator matrix of C, and H be a parity check matrix of C.

2.2.1 Extended Code

By adding parity check bits for all codewords, we can construct $(n+1, k)$ code C_{ex}. A codeword $(c_1, c_2, \ldots, c_n) \in C$ correspond to the codeword $(c_1, c_2, \ldots, c_n, c_{n+1}) \in C_{\text{ex}}$ where $c_{n+1} = c_1 + c_2 + \cdots + c_n$. The code C_{ex} is called an extended code of C.

The extension is effective for a code C with odd d. Then C_{ex} is an $(n+1, k, d+1)$ code. That is, the minimum distance increases by one.

A generator matrix G_{ex} of C_{ex} is obtained from G by adding parity check bits to odd weight rows in G. A parity check matrix H_{ex} of C_{ex} is obtained from H by lengthen every row in H by adding 0 in the $(n+1)$-th coordinate and adding the all-one row of length $n+1$.

2.2.2 Punctured Code

We can construct an $(n-1, k)$ code C_{punc} by deleting one coordinate for all codewords. The code C_{punc} is called a punctured code of C.

A generator matrix G_{punc} (and a parity check matrix H_{punc}) of C_{punc} is obtained from G (and H) by deleting one coordinate from G (and H).

2.2.3 Even Weight Subcode

We consider a code C having odd-weight codewords. Note that every linear code either has both odd and even weight codewords or has only even-weight codewords. We can construct an $(n, k-1)$ code C_{even} by removing odd-weight codewords from C. The code C_{even} is called an even weight subcode of C.

To obtain a generator matrix G_{even}, first we reduce G by elementary row operations to G' that has only one odd-weight row in the matrix. Then we obtain G_{even} by deleting the odd-weight row from G'. A parity check matrix H_{even} is obtained by adding all-one row to H.

2.3 Code Families

We show codes families we will use in this dissertation[2].

2.3.1 Reed-Muller Codes

Reed-Muller codes are *polynomial evaluation codes*. A message is some polynomial f or the coefficients of f. The codeword from a message f is an n-tuple of the evaluations of f at n points. Namely, the message f is encoded as

$$(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)),$$

[2] Since we consider binary codes in our work, we here define code families over binary field. We can also define these codes over alphabet size q with $q \geq 3$.

where α_i for $1 \leq i \leq n$ are the evaluation points. We sometimes refer to a polynomial f as a codeword (not only as a message).

A "message" polynomial f in Reed-Muller codes is an m-variate polynomial and the evaluation points are all distinct elements in $\mathbb{F}^m$. The set of m-variate polynomials with degree at most r is the r-th order Reed-Muller code, denoted by $\mathrm{RM}_{m,r}$. The code length of $\mathrm{RM}_{m,r}$ is 2^m, the dimension is $1 + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{r}$, and the minimum distance is 2^{m-r}. Since each variable in $f \in \mathrm{RM}_{m,r}$ takes a binary value, $\mathrm{RM}_{m,r}$ corresponds to the set of Boolean functions of m-variables with degree at most r.

The automorphism group of $\mathrm{RM}_{m,r}$ contains the *general affine group*. A permutation $\pi_{A,\boldsymbol{b}} : \mathbb{F}^m \to \mathbb{F}^m$ in the general affine group is

$$\pi_{A,\boldsymbol{b}} : \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \longmapsto A \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} + \boldsymbol{b},$$

where A is an invertible $m \times m$ matrix over $\mathbb{F}$ and $\boldsymbol{b}$ is an m-tuple column vector over $\mathbb{F}$. The size of the general affine group is

$$2^m(2^m - 1)(2^m - 2) \cdots (2^m - 2^{m-1}) \in 2^{O(n \log n)}.$$

Binary Reed-Muller codes correspond to the Boolean functions. Since $\overline{f}$ for $f \in \mathrm{RM}_{m,r}$ represents the negation of f, a codeword $\boldsymbol{1} + \boldsymbol{c}$ for $\boldsymbol{c} \in \mathrm{RM}_{m,r}$ is denoted by $\overline{\boldsymbol{c}}$.

2.3.2 BCH Codes

Since a BCH code over $\mathbb{F}$ is a subfield subcode of a Reed-Solomon code over $\mathbb{F}^m$, we first define Reed-Solomon codes. BCH codes and Reed-Solomon codes are also polynomial evaluation codes.

A message polynomial f in Reed-Solomon codes over $\mathbb{F}^m$ is a univariate polynomial over $\mathbb{F}^m$ and the evaluation points is distinct n points in $\mathbb{F}^m$. It is required that $n \leq 2^m$. The $(n, n-(d-1), d)$ Reed-Solomon code is the set of univariate polynomials of degree at most $n - d$ over $\mathbb{F}^m$. Then the binary subfield subcode of the Reed-Solomon code is a BCH code. That is, for the $(n, n-(d-1), d)$ Reed-Solomon code C, a BCH code is $C \cap \mathbb{F}^n$. A BCH code is called *primitive* if $n = 2^m - 1$.

It is known that the BCH code from the $(n, n-(d-1), d)$ Reed-Solomon code is an (n, k, d) code with $k \geq n - 1 - m(\lceil d/2 \rceil - 1)$. In particular, for even d, we have an (n, k, d) BCH code with $k \geq n - 1 - m(t - 1)$. A simple and complete proof of this fact appears in [37].

Since an extended primitive BCH code has code length $n = 2^m$, the evaluation points for extended primitive BCH codes are the all distinct points in $\mathbb{F}^m$. The automorphism group of the extended primitive BCH codes contains the *affine group*. A permutation $\pi_{a,b} : \mathbb{F}^m \to \mathbb{F}^m$ in the affine group is such that

$$\pi_{a,b} : x \longmapsto ax + b,$$

where $a, b \in \mathbb{F}^m$. The size of the affine group is $2^m(2^m - 1)$.

2.4 Remarks

We provide basic definitions and properties of linear codes in this chapter. For more information on error correcting codes, see [26, 34, 39].

Reed-Muller codes were firstly treated by Muller [30] and Reed [33]. In our work, the first-order Reed-Muller codes are the target codes of the analysis in Chapter 3 and Reed-Muller codes appear in Sections 4.3, 4.5, 6.5.3, and 6.6. BCH codes were introduced by Bose and Chaudhuri [10, 11], and independently by Hocquenghem [20]. In our work, BCH codes appear in Sections 4.3 and 6.6. The relations of the local weight distributions between the original code and its modified codes, including the extended code and the even weight subcode, are treated in Chapter 5.

Chapter 3

Monotone Error Structure in First-Order Reed-Muller Codes

3.1 Introduction

If the uncorrectable (and correctable) errors have the monotone structure, they are characterized by the *minimal uncorrectable* (and *maximal correctable*) errors. Helleseth, Kløve, and Levenshtein [19] introduced *larger halves* of codewords to describe the minimal uncorrectable errors.

In this chapter, the monotone structure of the the first-order Reed-Muller codes is investigated. Let RM_m denote the first-order Reed-Muller code of length 2^m. RM_m is a $(2^m, m+1, 2^{m-1})$ code. First, the numbers of uncorrectable errors (and thus correctable errors) of weights half the minimum distance and half the minimum distance plus one are determined. Namely, $|E^1_{2^{m-2}}(\mathrm{RM}_m)|$ and $|E^1_{2^{m-2}+1}(\mathrm{RM}_m)|$ are determined. The result for weight 2^{m-2} was already given in [40]. The approach here does not reveal the structure of cosets containing weight-2^{m-2} vectors completely, and is more direct and thus simpler. After that, the weight distribution of the minimal uncorrectable errors for RM_m is derived.

Section 3.2 states the properties of larger halves. Section 3.3 provides the definition of the first-order Reed-Muller codes and their properties related to larger halves. The results for the numbers of uncorrectable errors of weight half the minimum distance and half the minimum distance plus one are presented in Sections 3.4 and 3.5, respectively. The weight distribution of the minimal uncorrectable errors is derived in Section 3.6.

3.2 Minimal Uncorrectable Errors and Larger Halves

When the set of uncorrectable errors $E^1(C)$ has a monotone structure, $E^1(C)$ can be characterized by minimal uncorrectable errors in $E^1(C)$. An uncorrectable error $\boldsymbol{y} \in E^1(C)$ is minimal if there exists no $\boldsymbol{x}$ such that $\boldsymbol{x} \subset \boldsymbol{y}$ in $E^1(C)$. Let $M^1(C)$ denote the set of all minimal uncorrectable errors in C. Larger halves of a codeword $\boldsymbol{c} \in C$ are defined as minimal vectors $\boldsymbol{v}$ with respect to covering such that $\boldsymbol{v} + \boldsymbol{c} \prec \boldsymbol{v}$. Note that every larger half is an uncorrectable error. The following condition is a necessary and sufficient condition that $\boldsymbol{v} \in \mathbb{F}^n$ is a larger half of $\boldsymbol{c} \in C$:

$$\boldsymbol{v} \subseteq \boldsymbol{c}, \tag{3.1}$$

$$w(\boldsymbol{c}) \leq 2w(\boldsymbol{v}) \leq w(\boldsymbol{c}) + 2, \tag{3.2}$$

$$l(\boldsymbol{v}) \begin{cases} = l(\boldsymbol{c}) & \text{if } \ 2w(\boldsymbol{v}) = w(\boldsymbol{c}), \\ > l(\boldsymbol{c}) & \text{if } \ 2w(\boldsymbol{v}) = w(\boldsymbol{c}) + 2. \end{cases} \tag{3.3}$$

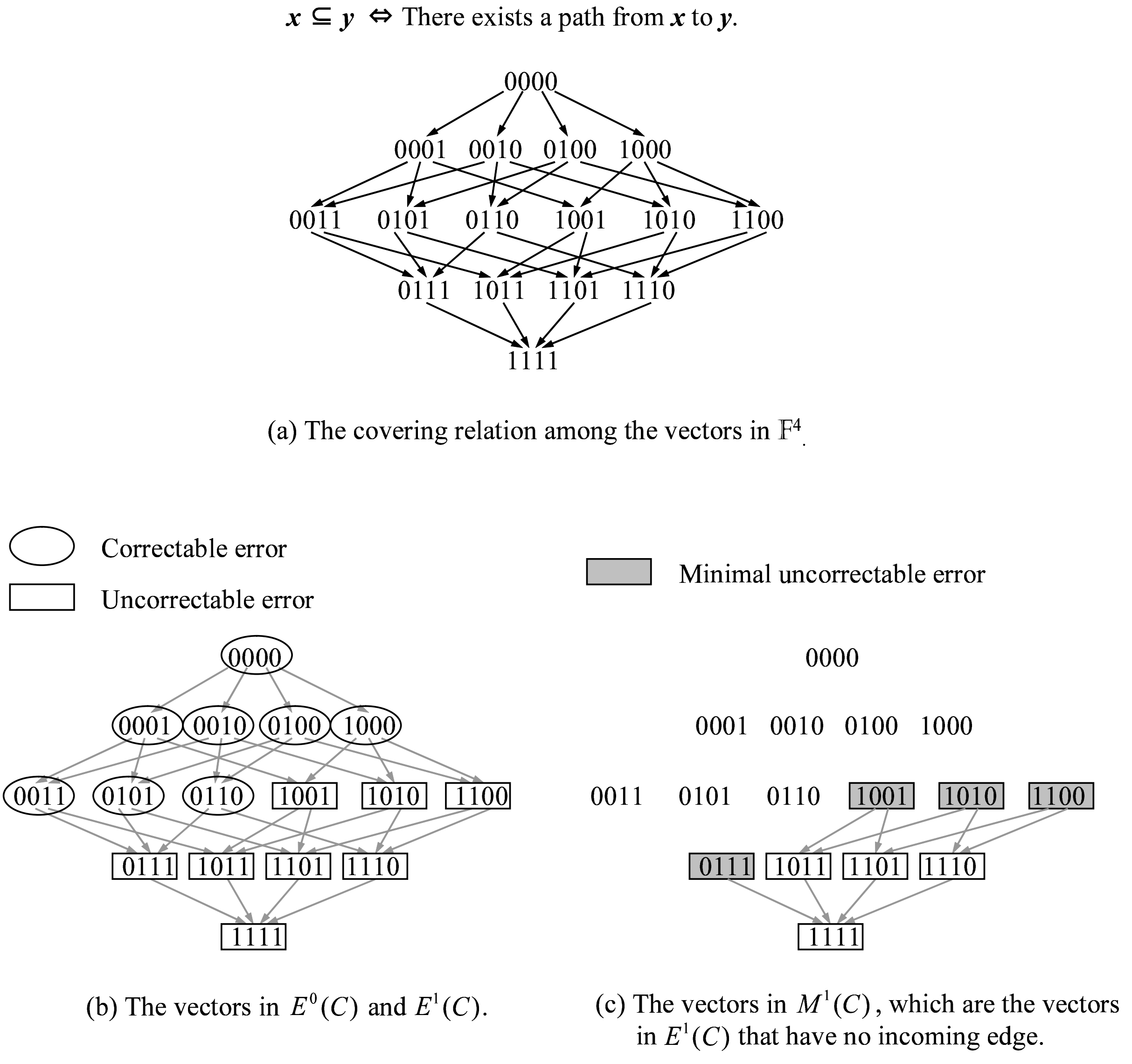
The condition (3.3) is not applied for the case that $w(\boldsymbol{c})$ is odd. The proof of equivalence between the definition and the above condition is found in the proof of Theorem 1 of [19]. Let $LH(\boldsymbol{c})$ be the set of all larger halves of $\boldsymbol{c} \in C$. For a subset U of $C \setminus \{\boldsymbol{0}\}$, let

$$LH(U) = \bigcup_{\boldsymbol{c} \in U} LH(\boldsymbol{c}).$$

For an even-weight codeword $\boldsymbol{c}$, the weights of larger halves of $\boldsymbol{c}$ are $w(\boldsymbol{c})/2$ and $w(\boldsymbol{c})/2 + 1$ from the condition (3.2). Let $LH^-(\boldsymbol{c})$ and $LH^+(\boldsymbol{c})$ denote the sets of larger halves of $\boldsymbol{c}$ of weight $w(\boldsymbol{c})/2$ and $w(\boldsymbol{c})/2 + 1$, respectively. Then $LH(\boldsymbol{c}) = LH^-(\boldsymbol{c}) \cup LH^+(\boldsymbol{c})$. Also let $LH^-(U) = \bigcup_{\boldsymbol{c} \in U} LH^-(\boldsymbol{c})$ and $LH^+(U) = \bigcup_{\boldsymbol{c} \in U} LH^+(\boldsymbol{c})$ for $U \subseteq C_{\text{even}}$. Define the set of minimal uncorrectable errors $M^1(C)$ as the set of uncorrectable errors $\boldsymbol{v} \in E^1(C)$ such that $\boldsymbol{u} \subseteq \boldsymbol{v}$ for $\boldsymbol{u} \in E^1(C)$ implies $\boldsymbol{u} = \boldsymbol{v}$. The set $M^1(C)$ consists of minimal (with respect to covering) vectors in $E^1(C)$, and $LH(\boldsymbol{c})$ for $\boldsymbol{c} \in C$ consist of minimal vectors in $\{\boldsymbol{v} : \boldsymbol{v} + \boldsymbol{c} \prec \boldsymbol{v}\}$, which is a subset of $E^1(C)$. Therefore, the following holds;

$$M^1(C) \subseteq LH(C \setminus \{\boldsymbol{0}\}). \tag{3.4}$$

Example. Let $C = \{0000, 1111\}$. The code C is a $(4, 1, 4)$ code. There are $2^{4-1} = 8$ coset leaders. The set of them is $E^0(C) = \{0000, 0001, 0010, 0100, 1000, 0011, 0101\}$ and the set of uncorrectable errors is $E^1(C) = \{1001, 1010, 1100, 0011, 1011, 1101, 1110, 1111\}$. Then the set of minimal uncorrectable errors is $M^1(C) = \{0111, 1001, 1010, 1100\}$. The



(a) The covering relation among the vectors in $\mathbb{F}^4$.

(b) The vectors in $E^0(C)$ and $E^1(C)$.

(c) The vectors in $M^1(C)$, which are the vectors in $E^1(C)$ that have no incoming edge.

Figure 3.1: The covering relation among the vectors in $E^0(C)$, $E^1(C)$, and $M^1(C)$ for $C = \{0000, 1111\}$.

covering relation among the vectors in $\mathbb{F}^4$ are depicted by a directed graph in Figure 3.1-(a). We can see $E^0(C), E^1(C)$ in Figure 3.1-(b) and $M^1(C)$ in Figure 3.1-(c).

3.3 First-Order Reed-Muller Code RM_m

3.3.1 Nonlinearity of Boolean Functions and RM_m

The binary r-th order Reed-Muller code of length 2^m corresponds to the Boolean functions of m variables with degree at most r. Hence RM_m corresponds to the set of affine functions of m variables. The *nonlinearity* of a Boolean function f is defined as the minimum distance between f and affine functions, and is equal to the weight of the coset leader in the coset f belongs to. Thus the weight distribution of coset leaders of RM_m represents the distribution of nonlinearity of Boolean functions. If the number of coset leaders of weight i is N, the number of Boolean functions with nonlinearity i is $N \cdot |\mathrm{RM}_m| = N2^{m+1}$. Nonlinearity is an important criterion for cryptographic system, block ciphers and stream ciphers. There has been much study of nonlinearity of Boolean functions in cryptography (see [12, 13] and references therein). The weight distributions of the cosets of RM_5 are completely determined in [7]. In general, however, it is infeasible to compute the weight distributions of the cosets (even only the coset leaders) of RM_m.

3.3.2 Larger Halves in RM_m

For an integer $m \geq 1$, RM_m is defined recursively as

$$\mathrm{RM}_m = \begin{cases} \mathbb{F}^2 & \text{for } m = 1, \\ \displaystyle\bigcup_{\boldsymbol{c} \in \mathrm{RM}_{m-1}} \{\boldsymbol{c} \circ \boldsymbol{c}, \boldsymbol{c} \circ \overline{\boldsymbol{c}}\} & \text{for } m \geq 2, \end{cases}$$

where $\boldsymbol{u} \circ \boldsymbol{v}$ denotes the concatenation of $\boldsymbol{u}$ and $\boldsymbol{v}$. Since all codewords in RM_m except the all-zero and the all-one codewords are minimum weight codewords, $\mathrm{RM}_m^* = \mathrm{RM}_m \setminus \{\mathbf{0}, \mathbf{1}\}$.

From the conditions (3.1)–(3.3) we have

$$|LH^-(\boldsymbol{c})| = \binom{2^{m-1}-1}{2^{m-2}-1} = \frac{1}{2}\binom{2^{m-1}}{2^{m-2}}, \tag{3.5}$$

$$|LH^+(\boldsymbol{c})| = \binom{2^{m-1}-1}{2^{m-2}+1} \tag{3.6}$$

for every $\boldsymbol{c} \in \mathrm{RM}_m^*$.

Define

$$S_m = \{l(\boldsymbol{c}) : \boldsymbol{c} \in \mathrm{RM}_m\}.$$

Then S_m forms information bits for RM_m, and $|S_m| = m+1$. For notational simplicity, we write $S_m = \{s_1, s_2, \ldots, s_{m+1}\}$ with $s_1 < s_2 < \cdots < s_{m+1}$. We define the set $C_m(s_i) \subseteq \mathrm{RM}_m^*$ for $1 \le i \le m+1$ as follows:

$$C_m(s_i) = \{\boldsymbol{c} \in \mathrm{RM}_m^* : l(\boldsymbol{c}) = s_i\}.$$

Then $\mathrm{RM}_m^* = \bigcup_{i=1}^{m+1} C_m(s_i)$. We have

$$|C_m(s_i)| = \begin{cases} 2^m - 1 & \text{for } i = 1, \\ 2^{m+1-i} & \text{for } 2 \le i \le m+1. \end{cases} \tag{3.7}$$

Let $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_l$ be codewords in RM_m^*. We say $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_l$, and $\mathbf{1}$ are linearly independent if $a_1\boldsymbol{c}_1 + a_2\boldsymbol{c}_2 + \cdots + a_l\boldsymbol{c}_l + a_{l+1}\mathbf{1} = \mathbf{0}$ for $a_i \in \{0,1\}$, $1 \le i \le l+1$ implies $a_1 = a_2 = \cdots = a_{l+1} = 0$. That is, if $l+1$ codewords $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_l$, and $\mathbf{1}$ are linearly independent, then every $\boldsymbol{c}_i$ with $1 \le i \le l$ cannot be represented as a sum of other l codewords.

Lemma 1. *For $2 \le l \le m$, let $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_l$ be codewords in RM_m^* such that $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_l$, and $\mathbf{1}$ are linearly independent. Then $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \cdots \cap \boldsymbol{c}_l) = 2^{m-l}$.*

Proof. We prove the statement by induction on l. For the case $l = 2$, the statement follows from the fact that $w(\boldsymbol{c}_1+\boldsymbol{c}_2) = w(\boldsymbol{c}_1)+w(\boldsymbol{c}_2)-2w(\boldsymbol{c}_1\cap\boldsymbol{c}_2)$ and that $w(\boldsymbol{c}_1+\boldsymbol{c}_2) = w(\boldsymbol{c}_1) = w(\boldsymbol{c}_2) = 2^{m-1}$. For the induction step, assume that if l codewords in RM_m^* and $\mathbf{1}$ are linearly independent, then the weight of their intersection vector is 2^{m-l}. Let $\boldsymbol{c}_i \in \mathrm{RM}_m^*$ with $1 \le i \le l+1$ and $\mathbf{1}$ be linearly independent codewords. Let $\boldsymbol{x} = \boldsymbol{c}_1\cap\boldsymbol{c}_2\cap\cdots\cap\boldsymbol{c}_{l-1}\cap\boldsymbol{c}_l$ and $\boldsymbol{y} = \boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \cdots \cap \boldsymbol{c}_{l-1} \cap \boldsymbol{c}_{l+1}$. From the assumption, $w(\boldsymbol{x}) = w(\boldsymbol{y}) = 2^{m-l}$, and $w(\boldsymbol{x}+\boldsymbol{y}) = \boldsymbol{c}_1\cap\boldsymbol{c}_2\cap\cdots\cap\boldsymbol{c}_{l-1}\cap(\boldsymbol{c}_l+\boldsymbol{c}_{l+1}) = 2^{m-l}$ because $\boldsymbol{c}_i$ with $1 \le i \le l-1$, $\boldsymbol{c}_l+\boldsymbol{c}_{l+1}$, and $\mathbf{1}$ are linearly independent. From the relation $w(\boldsymbol{x}\cap\boldsymbol{y}) = (w(\boldsymbol{x})+w(\boldsymbol{y})-w(\boldsymbol{x}+\boldsymbol{y}))/2$ we have $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \cdots \cap \boldsymbol{c}_l \cap \boldsymbol{c}_{l+1}) = w(\boldsymbol{x} \cap \boldsymbol{y}) = (2^{m-l} + 2^{m-l} - 2^{m-l})/2 = 2^{m-(l+1)}$. $\square$

Lemma 2. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$ be distinct codewords in RM_m^*. For $m \ge 3$,*

$$w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) = \begin{cases} 2^{m-2} & \textit{if } \boldsymbol{c}_1 + \boldsymbol{c}_2 + \boldsymbol{c}_3 = \mathbf{1}, \\ 0 & \textit{if } \boldsymbol{c}_i + \boldsymbol{c}_j = \mathbf{1} \textit{ for different } i, j \in \{1,2,3\}, \\ 2^{m-3} & \textit{otherwise.} \end{cases}$$

Proof. The statement follows from the fact that $w(\boldsymbol{c}_1+\boldsymbol{c}_2+\boldsymbol{c}_3) = w(\boldsymbol{c}_1)+w(\boldsymbol{c}_2)+w(\boldsymbol{c}_3) - 2(w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) + w(\boldsymbol{c}_2 \cap \boldsymbol{c}_3) + w(\boldsymbol{c}_1 \cap \boldsymbol{c}_3)) + 4w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3)$ and Lemma 1. □

Lemma 3. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2$ be distinct codewords in $C_m(s_i)$ with $1 \leq i \leq m$. For $m \geq 2$,*

$$w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) = w(\boldsymbol{c}_1 \cap \overline{\boldsymbol{c}_2}) = 2^{m-2}.$$

Proof. Since $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C_m(s_i)$, $\boldsymbol{c}_1 \neq \overline{\boldsymbol{c}_2}$. Hence $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, and $\mathbf{1}$ are linearly independent. Thus from Lemma 1 we have the statement. □

3.4 Uncorrectable Errors of Weight 2^{m-2} for RM_m

In this section, we determine the number of uncorrectable errors of weight half the minimum distance for RM_m. The proof was already given in [40], but here we give a slightly simpler proof.

In the proof of [40], the cosets that have uncorrectable errors of weight 2^{m-2} are partitioned into three types. Then the number of cosets for each type is determined, and the structure of cosets containing the vectors of weight 2^{m-2} is revealed. On the other hand, in our proof, first we observe that uncorrectable errors of weight 2^{m-2} are equivalent to the set of larger halves of weight 2^{m-2} of codewords except the all-zero and the all-one codewords. Then counting the number of larger halves that are common among two or more codewords leads to the result. Our approach does not make clear the structure of cosets containing the vectors of weight 2^{m-2}. Therefore, our proof leads directly to the result and is thus simpler than that of [40].

We have $E^1_{2^{m-2}}(\mathrm{RM}_m) = LH^-(\mathrm{RM}^*_m)$ because the set of larger halves of weight 2^{m-2} contains the set of minimal uncorrectable errors of weight 2^{m-2}, and every uncorrectable error of weight 2^{m-2} is minimal. There can be some $\boldsymbol{v} \in E^1_{2^{m-2}}(\mathrm{RM}_m)$ that is a larger half of two or more codewords in RM^*_m. Let $i \geq 1$ be an integer. Define

$$D^i_m = \{\boldsymbol{v} \in E^1_{2^{m-2}}(\mathrm{RM}_m) : |\{\boldsymbol{c} \in \mathrm{RM}^*_m : \boldsymbol{v} \in LH^-(\boldsymbol{c})\}| = i\}.$$

That is, D^i_m is the set of all uncorrectable errors $\boldsymbol{v}$ of weight 2^{m-2} such that $\boldsymbol{v}$ is a common larger half among i codewords in RM^*_m. Then

$$|E^1_{2^{m-2}}(\mathrm{RM}_m)| = \sum_{i \geq 1} |D^i_m|. \tag{3.8}$$

The following lemma says that four or more codewords in RM^*_m cannot have a common larger half of weight 2^{m-2}.

Lemma 4. *$D_m^i = \emptyset$ for $m \geq 2$ and $i \geq 4$.*

Proof. For $\boldsymbol{v} \in E^1_{2^{m-2}}(\mathrm{RM}_m)$, assume that there are four codewords $\boldsymbol{c}_i \in \mathrm{RM}_m^*$ with $1 \leq i \leq 4$ such that $\boldsymbol{v} \in LH^-(\boldsymbol{c}_i)$. Then, from the condition (3.1), $\boldsymbol{v} \subseteq \boldsymbol{c}_i$ for $1 \leq i \leq 4$. Thus $|S(\boldsymbol{c}_i) \setminus S(\boldsymbol{v})| = 2^{m-2}$ for $1 \leq i \leq 4$. Since $S(\boldsymbol{c}_i) \setminus S(\boldsymbol{v}) \subset \{1, 2, \ldots, n\} \setminus S(\boldsymbol{v})$ for $1 \leq i \leq 4$ and $|\{1, 2, \ldots, n\} \setminus S(\boldsymbol{v})| = 3 \cdot 2^{m-2}$, there are two codewords, say $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, such that $(S(\boldsymbol{c}_1) \setminus S(\boldsymbol{v})) \cap (S(\boldsymbol{c}_2) \setminus S(\boldsymbol{v})) \neq \emptyset$. Then $|S(\boldsymbol{c}_1) \cap S(\boldsymbol{c}_2)| = |S(\boldsymbol{v})| + |(S(\boldsymbol{c}_1) \setminus S(\boldsymbol{v})) \cap (S(\boldsymbol{c}_2) \setminus S(\boldsymbol{v}))| > 2^{m-2}$. Hence, for the codeword $\boldsymbol{c}_1 + \boldsymbol{c}_2$, $w(\boldsymbol{c}_1 + \boldsymbol{c}_2) = w(\boldsymbol{c}_1) + w(\boldsymbol{c}_2) - 2w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) < 2^{m-1}$. This contradicts the fact that 2^{m-1} is the minimum weight. □

Corollary 1. *For $m \geq 2$,*

$$|E^1_{2^{m-2}}(\mathrm{RM}_m)| = |D_m^1| + |D_m^2| + |D_m^3|, \tag{3.9}$$

$$(2^m - 1)\binom{2^{m-1}}{2^{m-2}} = |D_m^1| + 2|D_m^2| + 3|D_m^3|. \tag{3.10}$$

Proof. (3.9) is from (3.8) and Lemma 4. The left-hand side of (3.10) is the product of $|\mathrm{RM}_m^*| = 2^{m+1} - 2$ and $|LH^-(\boldsymbol{c})|$ for $\boldsymbol{c} \in \mathrm{RM}_m^*$. This value is equal to the right-hand side from Lemma 4. □

Next, we will determine $|D_m^2|$ and $|D_m^3|$. $|D_m^1|$ and $|E^1_{2^{m-2}}(\mathrm{RM}_m)|$ will thereby be determined from Corollary 1.

Lemma 5. *For $m \geq 2$,*

$$D_m^2 = \bigcup_{s_i \in S_m \setminus \{s_1, s_{m+1}\}} \{\boldsymbol{c}_1 \cap \boldsymbol{c}_2 : \boldsymbol{c}_1, \boldsymbol{c}_2 \in C_m(s_i), \boldsymbol{c}_1 \neq \boldsymbol{c}_2\}, \tag{3.11}$$

$$D_m^3 = \{\boldsymbol{c}_1 \cap \boldsymbol{c}_2 : \boldsymbol{c}_1, \boldsymbol{c}_2 \in C_m(s_1), \boldsymbol{c}_1 \neq \boldsymbol{c}_2\}. \tag{3.12}$$

Proof. For different $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C_m(s_i)$ with $1 \leq i \leq m$, $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) = 2^{m-2}$ from Lemma 3, and $S(\boldsymbol{c}_1 \cap \boldsymbol{c}_2)$ contains s_i. Therefore, the vector $\boldsymbol{v} = \boldsymbol{c}_1 \cap \boldsymbol{c}_2$ is a larger half of both $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$. For $i = 1$, $\boldsymbol{v}$ is also a larger half of the codeword of $\overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$ since $\overline{\boldsymbol{c}_1 + \boldsymbol{c}_2} \in C_m(s_1)$ and $S(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \subset S(\overline{\boldsymbol{c}_1 + \boldsymbol{c}_2})$. For $2 \leq i \leq m$, there is no other codeword $\boldsymbol{c} \in C_m(s_i) \setminus \{\boldsymbol{c}_1, \boldsymbol{c}_2\}$ such that $\boldsymbol{v} \in LH^-(\boldsymbol{c})$. This can be shown by a similar argument of the proof of Lemma 4. For $i = m + 1$, there is only one codeword in $C_m(s_i)$. □

Corollary 2. *For $m \geq 2$,*

$$|D_m^2| = |D_m^3| = \frac{1}{3}\binom{2^m - 1}{2}.$$

Proof. From Lemma 5, for each codeword in $C_m(1)$ there are two other codewords such that those three have the common larger half. For each codeword in $C_m(s_i)$ for $2 \le i \le m$, there is another codeword such that those two have the common larger half. Therefore, we have

$$\begin{aligned}|D_m^3| &= \frac{|C_m(s_1)|(|C_m(s_1)|-1)}{6} \\ &= \frac{1}{3}\binom{2^m-1}{2}\end{aligned}$$

and

$$\begin{aligned}|D_m^2| &= \sum_{i=2}^{m} \frac{|C_m(s_i)|(|C_m(s_i)|-1)}{2} \\ &= \frac{1}{3}\binom{2^m-1}{2}\end{aligned}$$

from (3.7). □

The number of uncorrectable errors of weight half the minimum distance is determined by Corollaries 1 and 2.

Theorem 1 ([40]). *For* $m \ge 2$,

$$|E^1_{2^{m-2}}(\mathrm{RM}_m)| = (2^m-1)\binom{2^{m-1}}{2^{m-2}} - \binom{2^m-1}{2}.$$

The number of correctable errors are obtained by the equation $|E^0_{2^{m-2}}(\mathrm{RM}_m)| + |E^1_{2^{m-2}}(\mathrm{RM}_m)| = \binom{2^m}{2^{m-2}}$. These expressions can be approximated by Stirling's approximation, $n! \approx \sqrt{2\pi n}(n/e)^n$, and thus we have

$$\begin{aligned}|E^0_{2^{m-2}}(\mathrm{RM}_m)| &\approx \sqrt{\frac{16}{3\pi 2^m}}\left(\frac{16}{3\sqrt{3}}\right)^{2^{m-1}}, \\ |E^1_{2^{m-2}}(\mathrm{RM}_m)| &\approx \frac{2^{2^m+\frac{m+1}{2}}}{\sqrt{\pi}}.\end{aligned}$$

3.5 Uncorrectable Errors of Weight $2^{m-2}+1$ for RM_m

In this section, we determine the number of uncorrectable errors of weight half the minimum distance plus one for the first-order Reed-Muller codes.

The set $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$ contains $LH^+(\mathrm{RM}^*_m)$, and $LH^+(\mathrm{RM}^*_m)$ contains all minimal uncorrectable errors of weight $2^{m-2}+1$ from (4.1). Therefore, the remaining uncorrectable errors in $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$ are non-minimal ones.

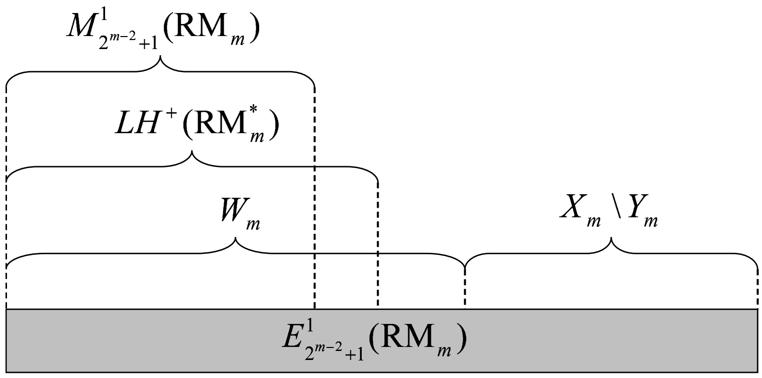


Figure 3.2: The structure of $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$. $M^1_{2^{m-2}+1}(\mathrm{RM}_m)$ is the set of minimal uncorrectable errors of weight $2^{m-2}+1$.

We will determine the size of $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$ by partitioning the set into two subsets. The first one is the set of vectors of weight $2^{m-2}+1$ that are covered by codewords in RM^*_m. More precisely, it is

$$W_m = \{\boldsymbol{v} \in \mathbb{F}^n_{2^{m-2}+1} : \boldsymbol{v} \subseteq \boldsymbol{c} \text{ for some } \boldsymbol{c} \in \mathrm{RM}^*_m\}, \tag{3.13}$$

where

$$\mathbb{F}^n_i = \{\boldsymbol{v} \in \mathbb{F}^n : w(\boldsymbol{v}) = i\} \quad \text{for } 1 \le i \le n.$$

Note that every $\boldsymbol{v} \in W_m$ is uncorrectable because the coset containing $\boldsymbol{v}$ contains the smaller weight vector $\boldsymbol{c}+\boldsymbol{v}$.

The second subset is the set of the remaining vectors, $E^1_{2^{m-2}+1}(\mathrm{RM}_m) \setminus W_m$. Here note that W_m contains $LH^+(\mathrm{RM}^*_m)$ and $LH^+(\mathrm{RM}^*_m)$ contains all minimal uncorrectable errors. Hence a vector in the second set is a non-minimal vector. Such a vector covers a minimal uncorrectable error of weight 2^{m-2}. Since the set of minimal uncorrectable errors of weight 2^{m-2} is $LH^-(\mathrm{RM}^*_m)$, we consider the set of vectors obtained by adding a weight-one vector to vectors in $LH^-(\mathrm{RM}^*_m)$ that are not covered by codewords in RM^*_m. Define

$$\mathbb{F}^n_1(\boldsymbol{c}) = \{\boldsymbol{e} \in \mathbb{F}^n_1 : \boldsymbol{e} \cap \boldsymbol{c} = \boldsymbol{0}\}$$

for $\boldsymbol{c} \in \mathrm{RM}^*_m$. Then, the second subset can be represented as $\mathrm{X}_m \setminus \mathrm{Y}_m$, where

$$\mathrm{X}_m = \bigcup_{\boldsymbol{c} \in \mathrm{RM}^*_m} \{\boldsymbol{v}+\boldsymbol{e} : \boldsymbol{v} \in LH^-(\boldsymbol{c}), \boldsymbol{e} \in \mathbb{F}^n_1(\boldsymbol{c})\},$$

$$\mathrm{Y}_m = \{\boldsymbol{u} \in \mathrm{X}_m : \boldsymbol{u} \subseteq \boldsymbol{c} \text{ for some } \boldsymbol{c} \in \mathrm{RM}^*_m\},$$

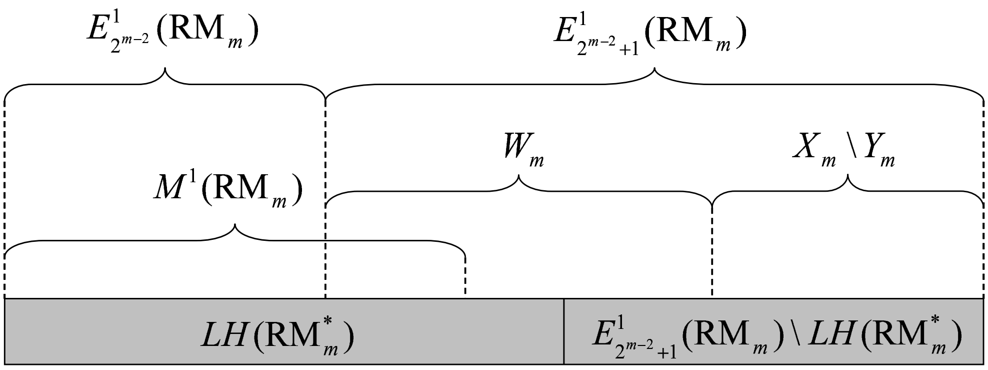


Figure 3.3: The relations between $LH(\mathrm{RM}^*_m)$, $E^1_{2^{m-2}}(\mathrm{RM}_m)$, and $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$.

and thus we have

$$|E^1_{2^{m-2}+1}(\mathrm{RM}_m)| = |W_m| + |X_m \setminus Y_m|. \tag{3.14}$$

The relations between $M^1(\mathrm{RM}_m)$, $LH^+(\mathrm{RM}^*_m)$, W_m, and $X_m \setminus Y_m$ in $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$ are shown in Figure 3.2. Figure 3.3 presents another view of the relations together with $E^1_{2^{m-2}}(\mathrm{RM}_m)$.

The set W_m contains $\binom{2^{m-1}}{2^{m-2}+1}$ vectors for each codeword in RM^*_m, and all $|\mathrm{RM}^*_m| \cdot \binom{2^{m-1}}{2^{m-2}+1}$ such vectors are distinct because of the following lemma.

Lemma 6. *Let $\boldsymbol{c}$ be a codeword in RM^*_m and $\boldsymbol{v}$ be a vector of weight $2^{m-2}+1$ such that $\boldsymbol{v} \subseteq \boldsymbol{c}$. Then there is no other codeword $\boldsymbol{c}'$ in RM^*_m such that $\boldsymbol{v} \subseteq \boldsymbol{c}'$.*

Proof. If $\boldsymbol{v} \subseteq \boldsymbol{c}'$, then $\boldsymbol{c}' \neq \overline{\boldsymbol{c}}$ and $w(\boldsymbol{c} \cap \boldsymbol{c}') \geq w(\boldsymbol{v}) = 2^{m-2}+1$. These contradict Lemma 1. □

Now we have

$$|W_m| = 2(2^m - 1)\binom{2^{m-1}}{2^{m-2}+1}.$$

Next, we will determine the size of $X_m \setminus Y_m$. For X_m and Y_m, we define the corresponding multisets $\tilde{\mathrm{X}}_m$ and $\tilde{\mathrm{Y}}_m$. That is, $\tilde{\mathrm{X}}_m$ is a multiset obtained by taking the union of the sets of vector obtained by adding vectors $\boldsymbol{e} \in \mathbb{F}^n_1(\boldsymbol{c})$ to larger halves $\boldsymbol{v} \in LH^-(\boldsymbol{c})$ for each $\boldsymbol{c} \in \mathrm{RM}^*_m$. The set $\tilde{\mathrm{Y}}_m$ is a multiset of vectors in $\tilde{\mathrm{X}}_m$ that are covered by some codeword in RM^*_m. Then we have

$$\begin{aligned} |\tilde{\mathrm{X}}_m| &= |\mathrm{RM}^*_m| \cdot \binom{2^{m-1}-1}{2^{m-2}-1} \cdot 2^{m-1} \\ &= 2^{m-1}(2^m - 1)\binom{2^{m-1}}{2^{m-2}} \end{aligned} \tag{3.15}$$

since the number of larger halves of each codeword is $\binom{2^{m-1}-1}{2^{m-2}-1}$ from (3.1)–(3.3), and there are 2^{m-1} choices for $\boldsymbol{e} \in \mathbb{F}_1^n(\boldsymbol{c})$. We determine $|\mathrm{X}_m \setminus \mathrm{Y}_m|$ by using $\tilde{\mathrm{X}}_m$ and $\tilde{\mathrm{Y}}_m$. First we show that the multiplicity of vectors in $\tilde{\mathrm{X}}_m \setminus \tilde{\mathrm{Y}}_m$ is not greater than 2.

Lemma 7. *The multiplicity of a vector in $\tilde{\mathrm{X}}_m \setminus \tilde{\mathrm{Y}}_m$ is less than or equal to 2 for $m \geq 5$.*

Proof. Let $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$ be distinct codewords in RM_m^*. For $1 \leq i \leq 3$, suppose there exist $\boldsymbol{v}_i, \boldsymbol{e}_i, \boldsymbol{u}$ such that $\boldsymbol{v}_i \in LH^-(\boldsymbol{c}_i)$, $\boldsymbol{e}_i \in \mathbb{F}_1^n(\boldsymbol{c}_i)$, $\boldsymbol{u} = \boldsymbol{v}_i + \boldsymbol{e}_i$, and there exists no $\boldsymbol{c}_4 \in \mathrm{RM}_m^*$ satisfying $\boldsymbol{u} \subseteq \boldsymbol{c}_4$.

First we show $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$, and $\mathbf{1}$ are linearly independent. Since $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) \neq 0$ for $m \geq 4$ from the assumption, we have $\boldsymbol{c}_2 \neq \overline{\boldsymbol{c}_1}$ and $\boldsymbol{c}_3 \neq \overline{\boldsymbol{c}_1}$. If $\boldsymbol{c}_3 = \boldsymbol{c}_1 + \boldsymbol{c}_2$ then $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) = 0$, leading to the contradiction. Suppose $\boldsymbol{c}_3 = \overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$. If $S(\boldsymbol{e}_1) \in S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)$ then $\boldsymbol{e}_1 = \boldsymbol{e}_3$ because $\{S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)\} \cap S(\boldsymbol{c}_3) = \emptyset$. In this case we cannot choose $\boldsymbol{e}_2$ such that $\boldsymbol{e}_2 \in \mathbb{F}_1^n(\boldsymbol{c}_2)$ and $\boldsymbol{e}_2 \subseteq \boldsymbol{c}_1 \cap \boldsymbol{c}_3$. The same thing occurs if $S(\boldsymbol{e}_1) \in S(\boldsymbol{c}_3) \setminus S(\boldsymbol{c}_1)$. Thus the contradiction arises, and $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$, and $\mathbf{1}$ are linearly independent.

If $\boldsymbol{v}_1 = \boldsymbol{v}_2$, then $\boldsymbol{v}_1 \in LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)$ and thus $\boldsymbol{v}_1 = \boldsymbol{c}_1 \cap \boldsymbol{c}_2$ from Lemma 3. Since $\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \subseteq \overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$ and $\boldsymbol{e}_1 \subseteq \overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$, we have $\boldsymbol{v}_1 + \boldsymbol{e}_1 \subseteq \overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$, leading to the contradiction. Therefore $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$ are distinct, and so are $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$. Then $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2 \cap \boldsymbol{v}_3) = 2^{m-2}-2$, and thus $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) \geq w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2 \cap \boldsymbol{v}_3) = 2^{m-2}-2$. On the other hand, $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) = 2^{m-3}$ from Lemma 1. Thus we have $2^{m-3} \geq 2^{m-2} - 2$. The contradiction arises for the case $m \geq 5$. □

Thus the size of $\mathrm{X}_m \setminus \mathrm{Y}_m$ is represented as follows.

$$|\mathrm{X}_m \setminus \mathrm{Y}_m| = |\tilde{\mathrm{X}}_m| - |\tilde{\mathrm{Y}}_m| - \frac{|\tilde{\mathrm{Z}}_m|}{2}, \tag{3.16}$$

where $\tilde{\mathrm{Z}}_m$ is the multiset defined as

$$\tilde{\mathrm{Z}}_m = \{\boldsymbol{v} \in \tilde{\mathrm{X}}_m : \boldsymbol{v} \not\subseteq \boldsymbol{c} \text{ for every } \boldsymbol{c} \in \mathrm{RM}_m^*, \text{ the multiplicity of } \boldsymbol{v} \text{ is } 2\}.$$

We will determine $|\tilde{\mathrm{Y}}_m|$ and $|\tilde{\mathrm{Z}}_m|$. The next lemma is useful to determine $|\tilde{\mathrm{Y}}_m|$.

Lemma 8. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathrm{RM}_m^*$. Then*

1. *there exist $\boldsymbol{v} \in LH^-(\boldsymbol{c}_1)$, $\boldsymbol{e} \in \mathbb{F}_1^n(\boldsymbol{c}_1)$ satisfying $\boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2$ if and only if*
$$\boldsymbol{c}_1 \neq \boldsymbol{c}_2 \text{ and } l(\boldsymbol{c}_1) \in S(\boldsymbol{c}_2); \tag{3.17}$$
2. *if (3.17) holds, then*
$$\begin{aligned}\{(\boldsymbol{v}, \boldsymbol{e}) : \boldsymbol{v} \in LH^-(\boldsymbol{c}_1), \boldsymbol{e} \in \mathbb{F}_1^n(\boldsymbol{c}_1), \boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2\}& \\ = \{(\boldsymbol{c}_1 \cap \boldsymbol{c}_2, \boldsymbol{e}) : \boldsymbol{e} \in \mathbb{F}_1^n, S(\boldsymbol{e}) \subseteq S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)\}&.\end{aligned} \tag{3.18}$$

Proof. (First part) The only if part is obvious. We prove the if part. Let $\boldsymbol{v} = \boldsymbol{c}_1 \cap \boldsymbol{c}_2$. Since $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$ and $\boldsymbol{c}_1 + \boldsymbol{c}_2 \neq \mathbf{1}$ from (3.17), we have $w(\boldsymbol{v}) = 2^{m-2}$ from Lemma 1. We have $l(\boldsymbol{v}) = l(\boldsymbol{c}_1)$ from $l(\boldsymbol{c}_1) \in S(\boldsymbol{c}_2)$. Thus $\boldsymbol{v} \in LH^-(\boldsymbol{c}_1)$. If we take $\boldsymbol{e} \in \mathbb{F}_1^n(\boldsymbol{c}_1)$ such that $S(\boldsymbol{e}) \subseteq S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)$, then $\boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2$.

(Second part) The $\supseteq$ part is obvious, so we show the $\subseteq$ part. Since $\boldsymbol{v} \subseteq \boldsymbol{c}_1$ and $\boldsymbol{v} \subseteq \boldsymbol{c}_2$, it holds that $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \geq w(\boldsymbol{v}) = 2^{m-2}$. We also have $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) = 2^{m-2}$. Therefore we have $\boldsymbol{v} = \boldsymbol{c}_1 \cap \boldsymbol{c}_2$. It immediately follows that $S(\boldsymbol{e}) \subseteq S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)$ from $\boldsymbol{c}_1 \cap \boldsymbol{e} = \mathbf{0}$ and $\boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2$. □

From Lemma 8, $\boldsymbol{v} + \boldsymbol{e} \in \tilde{\mathrm{X}}_m$ is covered by every $\boldsymbol{c}_2 \in \mathrm{RM}_m^*$ satisfying (3.17). The number of codewords $\boldsymbol{c}_2$ satisfying (3.17) is $|\mathrm{RM}_m|/2 - 2 = 2^m - 2$. There are $|S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)| = 2^{m-2}$ choices of $\boldsymbol{e}$ from (3.18). Thus we have

$$\begin{aligned} |\tilde{\mathrm{Y}}_m| &= |\mathrm{RM}_m^*| \cdot (2^m - 2) \cdot 2^{m-2} \\ &= 2^m(2^m - 1)(2^{m-1} - 1). \end{aligned} \tag{3.19}$$

The following lemma is useful to derive $|\tilde{\mathrm{Z}}_m|$.

Lemma 9. *Let $\boldsymbol{u} \in \tilde{\mathrm{X}}_m$ of multiplicity 2. That is, $\boldsymbol{u}$ is represented as $\boldsymbol{u} = \boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$ where $\boldsymbol{v}_i \in LH^-(\boldsymbol{c}_i)$, $\boldsymbol{c}_i \in \mathrm{RM}_m^*$, $\boldsymbol{e}_i \in \mathbb{F}_1^n(\boldsymbol{c}_i)$ for $i = 1, 2$, and $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$. Then, for $m \geq 5$, there exists $\boldsymbol{c}_3 \in \mathrm{RM}_m^*$ such that $\boldsymbol{u} \subseteq \boldsymbol{c}_3$ if and only if $\boldsymbol{e}_1 = \boldsymbol{e}_2$.*

Proof. First note that $\boldsymbol{c}_1 + \boldsymbol{c}_2 \neq \mathbf{1}$ since $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$ cannot hold for $m \geq 3$ if $\boldsymbol{c}_1 + \boldsymbol{c}_2 = \mathbf{1}$. (Only if part) We have $\boldsymbol{c}_1 \neq \boldsymbol{c}_3$ from $\boldsymbol{v}_1 + \boldsymbol{e}_1 \not\subseteq \boldsymbol{c}_1$ and $\boldsymbol{v}_1 + \boldsymbol{e}_1 \subseteq \boldsymbol{c}_3$. Since $\boldsymbol{v}_1 \subseteq \boldsymbol{c}_1$, and $\boldsymbol{v}_1 \subseteq \boldsymbol{c}_3$, we have $\boldsymbol{v}_1 = \boldsymbol{c}_1 \cap \boldsymbol{c}_3$. Equivalently, $\boldsymbol{v}_2 = \boldsymbol{c}_2 \cap \boldsymbol{c}_3$. Then $\boldsymbol{v}_1 \cap \boldsymbol{v}_2 = \boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3$, and hence $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2) = w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3)$. Since $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$ are distinct, $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3)$ is either 2^{m-2}, 2^{m-3}, or 0 from Lemma 2. We have $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2)$ is 2^{m-2} if $\boldsymbol{v}_1 = \boldsymbol{v}_2$, and is $2^{m-2} - 2$ otherwise because $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. Therefore $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2) = 2^{m-2}$ for $m \geq 5$ from the fact $2^{m-3} \neq 2^{m-2} - 2$. Hence $\boldsymbol{v}_1 = \boldsymbol{v}_2$, and thus $\boldsymbol{e}_1 = \boldsymbol{e}_2$.

(If part) Since $\boldsymbol{e}_1 = \boldsymbol{e}_2$ and $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$, we have $\boldsymbol{v}_1 = \boldsymbol{v}_2 = \boldsymbol{c}_1 \cap \boldsymbol{c}_2 \subseteq \overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$. Since $\boldsymbol{e}_1 \cap \boldsymbol{c}_1 = \boldsymbol{e}_2 \cap \boldsymbol{c}_2 = \boldsymbol{e}_1 \cap \boldsymbol{c}_2 = \mathbf{0}$, we have $\boldsymbol{e}_1 \subseteq \overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$. By taking $\boldsymbol{c}_3 = \overline{\boldsymbol{c}_1 + \boldsymbol{c}_2}$ we have $\boldsymbol{u} = \boldsymbol{v}_1 + \boldsymbol{e}_1 \subseteq \boldsymbol{c}_3$. □

From Lemma 9, for each $\boldsymbol{c}_1 \in \mathrm{RM}_m^*$, $|\tilde{\mathrm{Z}}_m|$ is obtained by counting all patterns in $\{\boldsymbol{v}_1 + \boldsymbol{e}_1 : \boldsymbol{v}_1 \in LH^-(\boldsymbol{c}_1), \boldsymbol{e}_1 \in \mathbb{F}_1^n(\boldsymbol{c}_1)\}$ such that $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$ for some $\boldsymbol{v}_2, \boldsymbol{e}_2$ with $\boldsymbol{v}_2 \in LH^-(\boldsymbol{c}_2), \boldsymbol{c}_2 \in \mathrm{RM}_m^* \setminus \{\boldsymbol{c}_1\}, \boldsymbol{e}_2 \in \mathbb{F}_1^n(\boldsymbol{c}_2)$ and $\boldsymbol{e}_1 \neq \boldsymbol{e}_2$. We will count such $\boldsymbol{v}_1 + \boldsymbol{e}_1$ for each $\boldsymbol{c}_1 \in \mathrm{RM}_m^*$.

There are three cases to be considered:

1. In the case that $l(\boldsymbol{c}_1) = l(\boldsymbol{c}_2)$; we choose $\boldsymbol{w}$ such that $\boldsymbol{w} \subseteq \boldsymbol{c}_1 \cap \boldsymbol{c}_2, w(\boldsymbol{w}) = 2^{m-2}-1$, and $l(\boldsymbol{w}) = l(\boldsymbol{c}_1 \cap \boldsymbol{c}_2)$. We choose $\boldsymbol{e}_2$ so that $S(\boldsymbol{e}_2) \subseteq S(\boldsymbol{c}_1) \setminus S(\boldsymbol{c}_2)$, and choose $\boldsymbol{e}_1$ so that $S(\boldsymbol{e}_1) \subseteq S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)$. Then letting $\boldsymbol{v}_1 = \boldsymbol{w} + \boldsymbol{e}_2$ and $\boldsymbol{v}_2 = \boldsymbol{w} + \boldsymbol{e}_1$ gives vectors as $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. There are $(2^{m-2}-1) \cdot 2^{m-2} \cdot 2^{m-2}$ such $\boldsymbol{v}_1 + \boldsymbol{e}_1$.

   For each codeword $\boldsymbol{c}_1$ in $C_m(s_i)$ there are $|C_m(s_i)|-1$ codewords $\boldsymbol{c}_2$ in RM_m^* satisfying $l(\boldsymbol{c}_1) = l(\boldsymbol{c}_2)$.

2. In the case that $l(\boldsymbol{c}_1) > l(\boldsymbol{c}_2)$; since $\boldsymbol{v}_1 \in LH^-(\boldsymbol{c}_1)$ and $\boldsymbol{v}_2 \in LH^-(\boldsymbol{c}_2)$, the $l(\boldsymbol{c}_2)$-th bit of $\boldsymbol{e}_1$ is one.

   (a) If the $l(\boldsymbol{c}_1)$-th bit of $\boldsymbol{c}_2$ is one; we choose $\boldsymbol{w}$ such that $\boldsymbol{w} \subseteq \boldsymbol{c}_1 \cap \boldsymbol{c}_2, w(\boldsymbol{w}) = 2^{m-2}-1$, and $l(\boldsymbol{w}) = l(\boldsymbol{c}_1 \cap \boldsymbol{c}_2)$. We choose $\boldsymbol{e}_2$ so that $S(\boldsymbol{e}_2) \subseteq S(\boldsymbol{c}_1) \setminus S(\boldsymbol{c}_2)$. Then letting $\boldsymbol{v}_1 = \boldsymbol{w} + \boldsymbol{e}_2$ and $\boldsymbol{v}_2 = \boldsymbol{w} + \boldsymbol{e}_1$ gives vectors as $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. There are $(2^{m-2}-1) \cdot 2^{m-2}$ such $\boldsymbol{v}_1 + \boldsymbol{e}_1$.
   For each codeword $\boldsymbol{c}_1$ in $C_m(s_i)$ with $i \geq 2$, there are $\left(\left(\sum_{j<i} |C_m(s_j)| + 1\right)/2 - 1\right)$ codewords $\boldsymbol{c}_2$ in RM_m^* satisfying $l(\boldsymbol{c}_1) \in S(\boldsymbol{c}_2)$.

   (b) If the $l(\boldsymbol{c}_1)$-th bit of $\boldsymbol{c}_2$ is zero; then $\boldsymbol{e}_2$ must be the vector having one in the $l(\boldsymbol{c}_1)$-th bit. We choose $\boldsymbol{w}$ such that $\boldsymbol{w} \subseteq \boldsymbol{c}_1 \cap \boldsymbol{c}_2$ and $w(\boldsymbol{w}) = 2^{m-2}-1$. Then letting $\boldsymbol{v}_1 = \boldsymbol{w} + \boldsymbol{e}_2$ and $\boldsymbol{v}_2 = \boldsymbol{w} + \boldsymbol{e}_1$ gives vectors as $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. There are 2^{m-2} such $\boldsymbol{v}_1 + \boldsymbol{e}_1$.
   For each codeword $\boldsymbol{c}_1$ in $C_m(s_i)$ with $i \geq 2$, there are $\left(\left(\sum_{j<i} |C_m(s_j)| + 1\right)/2 - 1\right)$ codewords $\boldsymbol{c}_2$ in RM_m^* satisfying $l(\boldsymbol{c}_1) \notin S(\boldsymbol{c}_2)$ and $\boldsymbol{c}_1 + \boldsymbol{c}_2 \neq \mathbf{1}$.

3. In the case that $l(\boldsymbol{c}_1) < l(\boldsymbol{c}_2)$; the number of vectors we should count is equal to that for the second case.

From the above analysis we have

$$\begin{aligned}
|\tilde{\mathrm{Z}}_m| &= \sum_{i=1}^{m+1} |C_m(s_i)|(|C_m(s_i)|-1)(2^{m-2}-1)(2^{m-2})^2 \\
&\quad + 2\sum_{i=2}^{m+1} |C_m(s_i)| \left(\frac{1}{2}\left(\sum_{j=1}^{i-1} |C_m(s_j)| + 1\right) - 1\right)(2^{m-2}-1)2^{m-2} \\
&\quad + 2\sum_{i=2}^{m+1} |C_m(s_i)| \left(\frac{1}{2}\left(\sum_{j=1}^{i-1} |C_m(s_j)| + 1\right) - 1\right)2^{m-2} \\
&= (2^{m-2}-1)(2^{m-2})^2\left((2^m-1)(2^m-2) + \sum_{i=2}^{m+1} 4^{m+1-i} - 2^{m+1-i}\right)
\end{aligned}$$

$$
\begin{aligned}
&\quad + 2(2^{m-2})^2 \left(\sum_{i=2}^{m+1} 2^{m+1-i}(2^m - 2^{m+1-i} - 1) \right) \\
&= (2^{m-2}-1)(2^{m-2})^2 \left((2^m-1)(2^m-2) + \frac{1}{3}(2^{2m}-1) - (2^m-1) \right) \\
&\quad + 2(2^{m-2})^2 \left((2^m-1)^2 - \frac{1}{3}(2^{2m}-1) \right) \\
&= (2^{m-2}-1)2^{m-1}\frac{2^m(2^m-1)(2^m-2)}{6} + 2^{m-1}\frac{2^m(2^m-1)(2^m-2)}{6} \\
&= 2^{2m-3}\binom{2^m}{3}. \qquad (3.20)
\end{aligned}
$$

From (3.14), (3.15), (3.16), (3.19), and (3.20), we can determine the number of uncorrectable errors of weight $2^{m-2}+1$ for RM_m.

Theorem 2. *For $m \geq 5$,*

$$|E^1_{2^{m-2}+1}(\mathrm{RM}_m)| = 4(2^m-1)(2^{m-3}+1)\binom{2^{m-1}}{2^{m-2}+1} - (4^{m-2}+3)\binom{2^m}{3}.$$

The number of correctable errors of weight $2^{m-2}+1$ is obtained from the equation,

$$|E^0_{2^{m-2}+1}(\mathrm{RM}_m)| + |E^1_{2^{m-2}+1}(\mathrm{RM}_m)| = \binom{2^m}{2^{m-2}+1}.$$

The number of Boolean functions of m variables with nonlinearity $2^{m-2}+1$ is immediately given.

Corollary 3. *The number of Boolean functions of m variables with nonlinearity $2^{m-2}+1$ is $2^{m+1}|E^0_{2^{m-2}+1}(\mathrm{RM}_m)|$, for the case $m \geq 5$, which is equal to*

$$2^{m+1}\left(\binom{2^m}{2^{m-2}+1} - 4(2^m-1)(2^{m-3}+1)\binom{2^{m-1}}{2^{m-2}+1} + (4^{m-2}+3)\binom{2^m}{3} \right).$$

The expressions for $|E^0_{2^{m-2}+1}(\mathrm{RM}_m)|$ and $|E^1_{2^{m-2}+1}(\mathrm{RM}_m)|$ are approximated as

$$
\begin{aligned}
|E^0_{2^{m-2}+1}(\mathrm{RM}_m)| &\approx \sqrt{\frac{3}{2^{m-3}\pi}} \left(\frac{16}{3\sqrt{3}} \right)^{2^{m-1}}, \\
|E^1_{2^{m-2}+1}(\mathrm{RM}_m)| &\approx \frac{2^{2^{m+1}+\frac{3}{2}m}}{\sqrt{\pi}}.
\end{aligned}
$$

3.6 Minimal Uncorrectable Errors for RM_m

In this section, we determine the weight distribution of minimal uncorrectable errors in the first-order Reed-Muller codes.

For an integer i with $0 \leq i \leq n$, define

$$M_i^1(C) = \{\boldsymbol{v} \in M^1(C) : w(\boldsymbol{v}) = i\}.$$

The weight distribution of minimal uncorrectable errors for RM_m is defined as $(|M_0^1(\mathrm{RM}_m)|, |M_1^1(\mathrm{RM}_m)|, \ldots, |M_n^1(\mathrm{RM}_m)|)$.

First we observe that $M^1(\mathrm{RM}_m) \subseteq LH(\mathrm{RM}_m^*)$ and $LH(\mathrm{RM}_m^*)$ contains vectors of weights 2^{m-2} and $2^{m-2}+1$. Since 2^{m-2} is the smallest weight in $M^1(\mathrm{RM}_m)$, $LH^-(\mathrm{RM}_m^*)(= E_{2^{m-2}}^1(\mathrm{RM}_m))$ is the set of minimal uncorrectable errors of weight 2^{m-2}. Thus we have

$$|M_i^1(\mathrm{RM}_m)| = \begin{cases} 0 & \text{for } \ 0 \leq i \leq 2^{m-2}-1, 2^{m-2}+2 \leq i \leq n, \\ |E_{2^{m-2}}^1(\mathrm{RM}_m)| & \text{for } \ i = 2^{m-2}. \end{cases} \tag{3.21}$$

The size of $E_{2^{m-2}}^1(\mathrm{RM}_m)$ is given in Theorem 1.

For the weight $2^{m-2}+1$ we have

$$|M_{2^{m-2}+1}^1(\mathrm{RM}_m)| = |LH^+(\mathrm{RM}_m^*)| - |LH^+(\mathrm{RM}_m^*) \setminus M^1(\mathrm{RM}_m)|. \tag{3.22}$$

We will determine $|LH^+(\mathrm{RM}_m^*)|$ and $|LH^+(\mathrm{RM}_m^*) \setminus M^1(\mathrm{RM}_m)|$ in the rest of this section.

The size of $LH^+(\mathrm{RM}_m^*)$ is immediately determined. From Lemma 6 there is no common larger half of weight $2^{m-2}+1$ of two or more codewords in RM_m^*. Therefore

$$\begin{aligned} |LH^+(\mathrm{RM}_m^*)| &= \binom{2^{m-1}-1}{2^{m-2}+1} \cdot |\mathrm{RM}_m^*| \\ &= 2(2^m-1)\binom{2^{m-1}-1}{2^{m-2}+1}. \end{aligned} \tag{3.23}$$

Next we will determine $|LH^+(\mathrm{RM}_m^*) \setminus M^1(\mathrm{RM}_m)|$. For $\boldsymbol{v} \in LH^+(\mathrm{RM}_m^*)$, $\boldsymbol{v} \notin M^1(\mathrm{RM}_m)$ if and only if $\boldsymbol{v} \supseteq \boldsymbol{v}'$ for some $\boldsymbol{v}' \in LH^-(\mathrm{RM}_m^*)$. Then the following lemma holds.

Lemma 10. *Let $\boldsymbol{c}, \boldsymbol{c}'$ be codewords in RM_m^*. Then*

1. *there exist $\boldsymbol{v} \in LH^+(\boldsymbol{c}), \boldsymbol{v}' \in LH^-(\boldsymbol{c}')$ satisfying $\boldsymbol{v} \supseteq \boldsymbol{v}'$ if and only if*

$$l(\boldsymbol{c}) < l(\boldsymbol{c}') \text{ and } l(\boldsymbol{c}') \in S(\boldsymbol{c}); \tag{3.24}$$

2. *if (3.24) holds, then*

$$\begin{aligned}&\{(\boldsymbol{v},\boldsymbol{v}') : \boldsymbol{v} \in LH^+(\boldsymbol{c}), \boldsymbol{v}' \in LH^-(\boldsymbol{c}'), \boldsymbol{v}' \subseteq \boldsymbol{v}\}\\ &\qquad = \{(\boldsymbol{c}\cap\boldsymbol{c}'+\boldsymbol{e}, \boldsymbol{c}\cap\boldsymbol{c}') : \boldsymbol{e} \in \mathbb{F}_1^n, S(\boldsymbol{e}) \subseteq S(\boldsymbol{c}) \setminus \{S(\boldsymbol{c}') \cup \{l(\boldsymbol{c})\}\}. \quad (3.25)\end{aligned}$$

Proof. (First part) We first show the if part. From (3.24), we have $\boldsymbol{c}+\boldsymbol{c}' \neq \boldsymbol{0}, \boldsymbol{1}$ and thus $w(\boldsymbol{c}\cap\boldsymbol{c}') = 2^{m-2}$ from Lemma 1. If we take $\boldsymbol{v}' = \boldsymbol{c}\cap\boldsymbol{c}'$ then $\boldsymbol{v}' \in LH^-(\boldsymbol{c}')$. Since $\boldsymbol{v}' \subseteq \boldsymbol{c}$ and $l(\boldsymbol{v}') = l(\boldsymbol{c}') > l(\boldsymbol{c})$, there exists $\boldsymbol{v} \in LH^+(\boldsymbol{c})$ satisfying $\boldsymbol{v}' \subseteq \boldsymbol{v}$. Next we show the only if part. The inequality $l(\boldsymbol{c}) < l(\boldsymbol{c}')$ comes from $l(\boldsymbol{c}) < l(\boldsymbol{v}) \leq l(\boldsymbol{v}') = l(\boldsymbol{c}')$, and $l(\boldsymbol{c}') \in S(\boldsymbol{c})$ comes from $l(\boldsymbol{c}') = l(\boldsymbol{v}') \in S(\boldsymbol{v}') \subseteq S(\boldsymbol{v}) \subseteq S(\boldsymbol{c})$.

(Second part) From the discussion on the first part of the proof, $\boldsymbol{v}' = \boldsymbol{c}\cap\boldsymbol{c}'$. Then $\boldsymbol{v} \in LH^+(\boldsymbol{c})$ if and only if $\boldsymbol{v} = \boldsymbol{v}+\boldsymbol{e}, \boldsymbol{e} \in \mathbb{F}_1^n, S(\boldsymbol{c}) \setminus \{S(\boldsymbol{c}') \cup \{l(\boldsymbol{c})\}\}$. □

Next we consider the number of $\boldsymbol{v}' \in LH^-(\mathrm{RM}_m^*)$ covered by $\boldsymbol{v} \in LH^+(\mathrm{RM}_m^*)$.

Lemma 11. *For* $\boldsymbol{v} \in LH^+(\mathrm{RM}_m^*)$*, there is at most one* $\boldsymbol{v}' \in LH^-(\mathrm{RM}_m^*)$ *such that* $\boldsymbol{v}' \subseteq \boldsymbol{v}$ *for* $m \geq 4$.

Proof. Suppose there are two distinct vectors $\boldsymbol{v}' \in LH^-(\boldsymbol{c}')$ and $\boldsymbol{v}'' \in LH^-(\boldsymbol{c}'')$ such that $\boldsymbol{v}' \subseteq \boldsymbol{v}$ and $\boldsymbol{v}'' \subseteq \boldsymbol{v}$ for some $\boldsymbol{c}', \boldsymbol{c}'' \in \mathrm{RM}_m^*$. Then we have $\boldsymbol{v}' = \boldsymbol{c}\cap\boldsymbol{c}'$ and $\boldsymbol{v}'' = \boldsymbol{c}\cap\boldsymbol{c}''$ from Lemma 10. The vector $\boldsymbol{v}$ is represented as $\boldsymbol{v}'+\boldsymbol{e}_1$ and $\boldsymbol{v}''+\boldsymbol{e}_2$ for vectors $\boldsymbol{e}_1, \boldsymbol{e}_2 \in \mathbb{F}_1^n$. Then $d(\boldsymbol{v}', \boldsymbol{v}'') = d(\boldsymbol{v}+\boldsymbol{e}_1, \boldsymbol{v}+\boldsymbol{e}_2) = 2$, where $d(\boldsymbol{x}, \boldsymbol{y})$ is the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. However, $d(\boldsymbol{v}', \boldsymbol{v}'') = d(\boldsymbol{c}\cap\boldsymbol{c}', \boldsymbol{c}\cap\boldsymbol{c}'') \geq 2^{m-2}$ because $\boldsymbol{v}'$ and $\boldsymbol{v}''$ are distinct codewords in the second-order Reed-Muller code, the minimum distance of which is 2^{m-2}. Therefore a contradiction arises if $m \geq 4$. □

If $\boldsymbol{v} \in LH^+(\boldsymbol{c})$ covers $\boldsymbol{v}' \in LH^-(\boldsymbol{c}')$ for $\boldsymbol{c}' \in \mathrm{RM}_m^*$, then $\boldsymbol{v}'$ is unique for $\boldsymbol{v}$ from Lemma 11. Then the number of $\boldsymbol{v}$ in $LH^+(\boldsymbol{c})$ that covers $\boldsymbol{v}'$ is the size of $S(\boldsymbol{c}) \setminus \{S(\boldsymbol{c}') \cup \{l(\boldsymbol{c})\}\}$ from (3.25), which is equal to $2^{m-2}-1$. If we know the number of codewords whose larger halves cover $\boldsymbol{v}'$ for each $\boldsymbol{v}' \in LH^-(\mathrm{RM}_m^*)$, then the product of it and $2^{m-2}-1$ yields the number of vectors in $LH^+(\mathrm{RM}_m^*)$ that cover some larger half in $LH^-(\mathrm{RM}_m^*)$, which is $|LH^+(\mathrm{RM}_m^*) \setminus M^1(\mathrm{RM}_m)|$.

We determine the number of $\boldsymbol{v}' \in LH^-(\mathrm{RM}_m^*)$ such that $\boldsymbol{v}' \subseteq \boldsymbol{v}$ for some $\boldsymbol{v} \in LH^+(\mathrm{RM}_m^*)$. Suppose $\boldsymbol{v}' \in LH^-(\boldsymbol{c}')$ and $\boldsymbol{c}' \in C_m(s_i)$. Note from (3.24) that $i \neq 1$ because if $i = 1$ there is no $\boldsymbol{c}$ such that $l(\boldsymbol{c}) < s_i$. For $\boldsymbol{c}' \in C_m(s_i)$ with $i \leq 2$, the number of $\boldsymbol{c} \in \mathrm{RM}_m^*$ satisfying (3.24) is

$$\frac{|C_m(s_1)|+1}{2} - 1 + \sum_{j=2}^{i-1} \frac{|C_m(s_j)|}{2} = 2^m - 1 + 2^{m-i+1}.$$

From (3.25) we have $\boldsymbol{v}' = \boldsymbol{c} \cap \boldsymbol{c}'$. Then there may be other codeword $\boldsymbol{c}'' \in \mathrm{RM}_m^*$ such that $\boldsymbol{v}' = \boldsymbol{c} \cap \boldsymbol{c}''$. That is, $\boldsymbol{v}'$ is a common larger half of $\boldsymbol{c}'$ and $\boldsymbol{c}''$. Fortunately, the number of such larger halves is obtained in Section 3.4 and is $|D_m^2|$. In the case we consider here, there is no common larger half of three codewords, which is a larger half of a codeword in D_m^3. This is because, as in the proof of Lemma 5, D_m^3 consists of larger halves of codewords in $C_m(s_1)$, but the larger halves we consider here are those in $C_m(s_i)$ for $i \geq 2$. Therefore the number of $\boldsymbol{v}' \in LH^-(\mathrm{RM}_m^*)$ such that $\boldsymbol{v}' \subseteq \boldsymbol{v}$ for some $\boldsymbol{v} \in LH^+(\mathrm{RM}_m^*)$ is

$$\begin{aligned}
&\sum_{i=2}^{m+1} |C_m(s_i)|(2^m - 1 + 2^{m-i+1}) - |D_m^2| \\
&= \sum_{i=2}^{m+1} 2^{m-i+1}(2^m - 1 + 2^{m-i+1}) - \frac{1}{3}\binom{2^m - 1}{2} \\
&= (2^m - 1)\sum_{i=2}^{m+1} 2^{m-i+1} - \sum_{i=2}^{m+1} 4^{m-i+1} - \frac{(2^m-1)(2^m-2)}{6} \\
&= (2^m - 1)\sum_{i=0}^{m-1} 2^{i} - \sum_{i=0}^{m-1} 4^{i} - \frac{(2^m-1)(2^m-2)}{6} \\
&= (2^m - 1)^2 - \frac{4^m - 1}{3} - \frac{(2^m-1)(2^m-2)}{6} \\
&= \binom{2^m - 1}{2}.
\end{aligned}$$

Thus the product of $\binom{2^m-1}{2}$ and $2^{m-2} - 1$ gives the size of $|LH^+(\mathrm{RM}_m^*) \setminus M^1(\mathrm{RM}_m)|$.

Lemma 12. *For* $m \geq 4$*,*

$$|LH^+(\mathrm{RM}_m^*) \setminus M^1(\mathrm{RM}_m)| = (2^{m-2} - 1)\binom{2^m - 1}{2}.$$

Now the weight distribution of the minimal uncorrectable errors for RM_m is determined.

Theorem 3. *For* $m \geq 4$ *and* $0 \leq i \leq n$*,*

$$|M_i^1(\mathrm{RM}_m)| = \begin{cases} (2^m - 1)\binom{2^{m-1}}{2^{m-2}} - \binom{2^m-1}{2} & \text{for } i = 2^{m-2}, \\ 2(2^m - 1)\binom{2^{m-1}-1}{2^{m-2}+1} - (2^{m-2} - 1)\binom{2^m-1}{2} & \text{for } i = 2^{m-2} + 1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. The statement follows from Theorem 1, (3.21), (3.22), (3.23), and Lemma 12. □

By Stirling's approximation, we have $|M^1_{2^{m-2}+1}(\mathrm{RM}_m)| \approx |LH^+(\mathrm{RM}_m^*)| \approx \sqrt{\frac{2^m}{\pi}}\, 2^{2^{m-1}+1}$.

3.7 Concluding Remarks

In [40], the number of uncorrectable errors of weight 2^{m-2} for RM_m was derived. We have derived the same result with quite different and simple way. The main ingredients of our approach are larger halves and the monotone structure. Because of the property of larger halves, the number of uncorrectable errors of weight $2^{m-2}+1$ is also derived. The structure of $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$ is revealed as in Figure 3.2, but our approach does not reveal what type of vectors (functions) are the coset leaders of weight $2^{m-2}+1$.

One possible future work is applying our approach to the number of uncorrectable errors of weight greater than $2^{m-2}+1$, say $2^{m-2}+2$. If we take a similar approach to the case of weight $2^{m-2}+1$ to derive the number of uncorrectable errors of weight $2^{m-2}+2$, we will consider constructing the vectors obtained by adding weight one or two vectors to the minimal uncorrectable errors. Then if all the multiplicities of vectors obtained by the above construction are explicitly described, we can obtain the result, but it seems to be much more complicated than the case of weight $2^{m-2}+1$.

Chapter 4

Monotone Error Structure and Trial Sets

4.1 Introduction

Helleseth, Kløve, and Levenshtein [19] introduced *trial sets* for a code. Trial sets can be used for a minimum distance decoding and for giving an upper bound on the number of uncorrectable errors. The set of all codewords except the all-zero codeword and the set of minimal codewords [3] in the code are examples of trial sets. It is, however, desirable to obtain the smaller trial sets for their applications.

In this chapter, first some upper and lower bounds on the size of minimum trial sets are derived. Experimental results show our bound is tighter than known bounds, and the size of minimum trial sets are determined for several codes since upper and lower bounds coincides for them. Next we investigate whether trial sets always contain all the minimum weight codewords. We derive sufficient conditions under which all the minimum weight codewords are in every trial set and show that the condition is satisfied for Reed-Muller codes and random linear codes. For codes with odd minimum distance d, we give a sufficient condition under which all the codewords of weights d and $d + 1$ are in every trial set. For the code that satisfies the above conditions, we derive the upper and lower bounds on the number of uncorrectable errors of weight half the minimum distance. The lower bound asymptotically coincide with the upper bound for Reed-Muller codes and random linear codes.

The next section provides the definition and applications of trial sets. The bounds on the size of minimum trial sets are presented in Section 4.3. In Section 4.4, the conditions under which any trial set contains the minimum weight codewords are considered. In

Section 4.5, the lower bound on the number of uncorrectable errors of weight half the minimum distance is given for the codes satisfying the conditions presented in the previous section.

4.2 Definition and Applications of Trial Sets

A trial set T for the code C is defined as the set of codewords in $C \setminus \{\mathbf{0}\}$ that has the following property:

$$\boldsymbol{e} \in E^0(C) \text{ if and only if } \boldsymbol{e} \prec \boldsymbol{e} + \boldsymbol{c} \text{ for all } \boldsymbol{c} \in T.$$

Equivalently,

$$\boldsymbol{e} \in E^1(C) \text{ if and only if } \boldsymbol{e} + \boldsymbol{c} \prec \boldsymbol{e} \text{ for some } \boldsymbol{c} \in T.$$

The minimum distance decoding using a trial set T is the following.

[*Trial set decoding*]
Let $\boldsymbol{y} \in \mathbb{F}^n$ be a received vector.

1. Set $\boldsymbol{e} \leftarrow \boldsymbol{y}$.
2. Find a codeword $\boldsymbol{c} \in T$ such that $\boldsymbol{e} + \boldsymbol{c} \prec \boldsymbol{e}$.
   Set $\boldsymbol{e} \leftarrow \boldsymbol{e} + \boldsymbol{c}$.
3. Repeat Step 2 until no such $\boldsymbol{c}$ exists (Then $\boldsymbol{e}$ becomes the coset leader).
4. Output $\boldsymbol{e} + \boldsymbol{y}$.

From the definition of trial sets one can see that the decoder finds the coset leader of the coset containing the received vector, and thus performs as a minimum distance decoder. The trial set decoding is a type of gradient-like decoding [4]. Although there is no nontrivial upper bound on the time-complexity of the trial set decoding, the complexity seems to depend on the size of the trial set used in the algorithm.

The weight distribution of a trial set gives an upper bound on the number of uncorrectable errors. Let T be a trial set for an (n, k, d) linear code C. Then, for an integer i with $\lfloor (d-1)/2 \rfloor < i \leq n$, we have [19, Corollary 7]

$$|E_i^1(C)| \leq \sum_{j=d}^{2i} |A_j(T)| \sum_{l=\lceil j/2 \rceil}^{\min\{i,j\}} \binom{j}{l}\binom{n-j}{i-l} - \sum_{l=\lceil d/2 \rceil}^{i} |A_{2l}(T)| \binom{2l-1}{l}\binom{n-2l}{i-l}$$

$$= \sum_{j=\lceil \frac{d}{2} \rceil}^{2i} \left(|A_{2j}(T)| \left(\binom{2j-1}{j-1}\binom{n-2j}{i-j} + \sum_{l=j+1}^{\min\{i,2j\}} \binom{2j}{l}\binom{n-2j}{i-l} \right) \right.$$
$$\left. + |A_{2j-1}(T)| \sum_{l=j}^{\min\{i,2j-1\}} \binom{2j-1}{l}\binom{n-2j+1}{i-l} \right).$$

For two trial sets T and T' with $T' \subset T$, the bound using T' is tighter than that using T.

In both applications, smaller trial sets are desirable. Therefore, we consider the smallest trial set. Define a *minimum trial set* for C as the smallest trial set for C, denoted by $T_{\min}$. Note that $T_{\min}$ itself may not be unique.

A necessary and sufficient condition for a set to be a trial set is stated as follows [19, Corollary 3]:

$$T \subseteq C \setminus \{\mathbf{0}\} \text{ is a trial set for } C \text{ if and only if } M^1(C) \subseteq LH(T). \tag{4.1}$$

That is, a trial set is a set of codewords whose larger halves contain minimal uncorrectable errors.

The following proposition says that a trial set can consist of only minimal codewords.

Proposition 1 ([19, Corollary 5])**.** *Let T be a trial set for a linear code C of $d \geq 2$. Then $T \cap C^*$ is also a trial set for C.*

4.3 Size of Minimum Trial Sets

We give some upper and lower bounds on the size of minimum trial sets in this section. It is clear from Proposition 1 that $|T_{\min}| \leq |C^*|$. Let us define T_{nec} as the set of minimal codewords $\boldsymbol{c} \in C^*$ such that, for some $\boldsymbol{v} \in M^1(C)$, $\boldsymbol{v} \in LH(\boldsymbol{c})$ and $\boldsymbol{v} \notin LH(\boldsymbol{c}')$ for all $\boldsymbol{c}' \in C^* \setminus \{\boldsymbol{c}\}$. That is, for $\boldsymbol{c} \in C^*$,

$$\boldsymbol{c} \in T_{\text{nec}} \text{ if and only if } (M^1(C) \cap LH(\boldsymbol{c})) \setminus LH(C^* \setminus \{\boldsymbol{c}\}) \neq \emptyset.$$

Then codewords in T_{nec} are necessary to compose a trial set. We have the following bounds on the size of minimum trial sets.

Theorem 4. *Let $T_{\min}$ be a minimum trial set for an (n, k, d) linear code C with $d \geq 2$. Then*

$$\max\{k, |T_{\text{nec}}|\} \leq |T_{\min}| \leq |T_{\text{nec}}| + |M^1(C) \setminus LH(T_{\text{nec}})|.$$

Table 4.1: Bounds of the size of minimum trial sets for some BCH, extended BCH, and Reed-Muller codes.

(n,k) code C	Lower bounds		$\lvert T_{\min}\rvert$	Upper bounds	
	New			[19]	New
	k	$\lvert T_{\text{nec}}\rvert$		$\lvert C^*\rvert$	$\lvert T_{\text{nec}}\rvert + \lvert M^1(C) \setminus LH(T_{\text{nec}})\rvert$
(15,11) BCH	11*	11*	11∼83	308	83*
(15,7) BCH	7	44*	44∼87	108	87*
(15,5) BCH	5	30*	30	30*	30*
(16,11) exBCH	11	16*	16∼79	588	79*
(16,7) exBCH	7	45*	45∼86	126	86*
(16,5) exBCH	5	30*	30	30*	30*
(16,11) RM	11	15*	15∼79	588	79*
(16,5) RM	5	30*	30	30*	30*

* means the maximum/minimum value for the lower/upper bounds.

Proof. If a codeword $\boldsymbol{c} \in C$ is an input to a trial set decoder, then the decoder finds the coset leader $\mathbf{0}$ and thus outputs $\boldsymbol{c}$. The coset leader found by the decoder is a sum of codewords in $T_{\min}$ and the input. Therefore, the linear span of a trial set forms the code C. This leads to $k \leq |T_{\min}|$. $|T_{\text{nec}}| \leq |T_{\min}|$ is obvious. From the definition of T_{nec}, $T_{\min}$ contains T_{nec}. We show that the number of remaining codewords that should be in $T_{\min}$, that is $|T_{\min} \setminus T_{\text{nec}}|$, is upper bounded by $|M^1(C) \setminus LH(T_{\text{nec}})|$. Since the larger halves of $T_{\min}$ contain $M^1(C)$ from (4.1), the larger halves of the set $T_{\min} \setminus T_{\text{nec}}$ should contain the set $M^1(C) \setminus LH(T_{\text{nec}})$. Therefore, $|T_{\min} \setminus T_{\text{nec}}| \leq |M^1(C) \setminus LH(T_{\text{nec}})|$, and thus $|T_{\min}| \leq |T_{\text{nec}}| + |M^1(C) \setminus LH(T_{\text{nec}})|$. □

While a naive algorithm for computing $|T_{\min}|$ requires $2^{2^{O(n)}}$ time, the time complexity for computing $|T_{\text{nec}}|$ and $|M^1(C) \setminus LH(T_{\text{nec}})|$ is $2^{O(n)}$. Therefore, above bounds are useful to estimate $|T_{\min}|$.

We compute the bounds in Theorem 4 and the upper bound $|C^*|$ for some codes. The results are shown in Table 4.1. The new upper bound is tight for all codes compared to the known bound. The upper and lower bounds coincide for 3 codes, the $(15,5)$ BCH code, the $(16,5)$ extended BCH code, and the $(16,5)$ Reed-Muller code.

4.4 Minimum Weight Codewords in Trial Sets

In this section, we consider conditions under which any trial set contains all minimum weight codewords. Let d be a minimum distance (weight) of C. We consider sufficient conditions under which $A_d(C) \subseteq T_{\text{nec}}$ holds.

4.4.1 Odd Minimum Weight Case

When d is odd, the weight of the vectors in $LH(\boldsymbol{c})$ for $\boldsymbol{c} \in A_d(C)$ is $(d+1)/2$. Since the weight $(d+1)/2$ is the minimum weight of the uncorrectable errors, the uncorrectable errors of weight $(d+1)/2$ are minimal uncorrectable errors. This means that, for every $\boldsymbol{c} \in A_d(C)$, $LH(\boldsymbol{c})$ are minimal uncorrectable errors. Therefore we have

$$\boldsymbol{c} \in T_{\text{nec}} \text{ if and only if } LH(\boldsymbol{c}) \setminus LH(C^* \setminus \{\boldsymbol{c}\}) \neq \emptyset \tag{4.2}$$

for $\boldsymbol{c} \in A_d(C)$ with odd minimum distance d. We have the following facts.

Lemma 13. *Let C be a linear code with odd minimum distance d and $\boldsymbol{c}$ be a codeword in $A_d(C)$. Then $LH(\boldsymbol{c}) \cap LH(\boldsymbol{c}') = \emptyset$ for any $\boldsymbol{c}' \in A_d(C) \setminus \{\boldsymbol{c}\}$.*

Proof. Suppose for contradiction that there exists $\boldsymbol{v} \in LH(\boldsymbol{c}) \cap LH(\boldsymbol{c}')$ for some $\boldsymbol{c}' \in A_d(C) \setminus \{\boldsymbol{c}\}$, then we have $\boldsymbol{v} \subseteq \boldsymbol{c}$, $\boldsymbol{v} \subseteq \boldsymbol{c}'$, and thus $w(\boldsymbol{c} \cap \boldsymbol{c}') \geq w(\boldsymbol{v}) = (d+1)/2$. Then $w(\boldsymbol{c} + \boldsymbol{c}') = w(\boldsymbol{c}) + w(\boldsymbol{c}') - 2w(\boldsymbol{c} \cap \boldsymbol{c}') \leq d - 1$, contradicting the minimum weight of d. Therefore the statement follows. □

Lemma 14. *Let C be a linear code with odd minimum distance d and $\boldsymbol{c}$ be a codeword in $A_d(C)$. Then $LH(\boldsymbol{c}) \cap LH^-(\boldsymbol{c}') \neq \emptyset$ for $\boldsymbol{c}' \in A_{d+1}(C)$ if and only if $l(\boldsymbol{c}') \in S(\boldsymbol{c})$ and $w(\boldsymbol{c} \cap \boldsymbol{c}') = (d+1)/2$.*

Proof. (Only If part) If there is $\boldsymbol{v} \in LH(\boldsymbol{c}) \cap LH^-(\boldsymbol{c}')$, then $\boldsymbol{v} \subseteq \boldsymbol{c}, \boldsymbol{v} \subseteq \boldsymbol{c}', w(\boldsymbol{v}) = (d+1)/2$. Thus $w(\boldsymbol{c} \cap \boldsymbol{c}') \geq (d+1)/2$. On the other hand, it holds that $w(\boldsymbol{c} + \boldsymbol{c}') = w(\boldsymbol{c}) + w(\boldsymbol{c}') - 2w(\boldsymbol{c} \cap \boldsymbol{c}') \geq d$. The last inequality follows from the fact that $\boldsymbol{c} + \boldsymbol{c}'$ is a codeword in C. Since $w(\boldsymbol{c}) = d$ and $w(\boldsymbol{c}') = d+1$, we have $w(\boldsymbol{c} \cap \boldsymbol{c}') \leq (d+1)/2$. Therefore $w(\boldsymbol{c} \cap \boldsymbol{c}') = (d+1)/2$ holds. The condition $l(\boldsymbol{c}') \in S(\boldsymbol{c})$ must hold because all vectors $\boldsymbol{v}$ in $LH^-(\boldsymbol{c}')$ meets $l(\boldsymbol{v}) = l(\boldsymbol{c})$ from the definition.

(If part) If we have $l(\boldsymbol{c}') \in S(\boldsymbol{c})$ and $w(\boldsymbol{c} \cap \boldsymbol{c}') = (d+1)/2$, then $\boldsymbol{c} \cap \boldsymbol{c}'$ is a common larger half of $\boldsymbol{c}$ and $\boldsymbol{c}'$. □

From the above lemmas, we have the following sufficient condition under which all minimum weight codewords are in any trial set for odd minimum distance codes.

Theorem 5. *Let C be a linear code with odd minimum distance d. Then $A_d(C) \subseteq T_{\mathrm{nec}}$ holds if*

$$\binom{d}{\frac{d+1}{2}} > |A_{d+1}(C)|. \tag{4.3}$$

Proof. From Lemmas 13 and 14, we observe that two codewords in $A_d(C)$ does not have common larger halves and that a codeword in $A_d(C)$ and that in $A_{d+1}(C)$ can have a common larger half. First we show that the number of common larger half among $\boldsymbol{c} \in A_d(C)$ and $\boldsymbol{c}' \in A_{d+1}(C)$ is at most one. For contradiction, suppose there exist two distinct vectors in $LH(\boldsymbol{c}) \cap LH^-(\boldsymbol{c}')$. Then it must hold that $w(\boldsymbol{c} \cap \boldsymbol{c}') \geq (d+1)/2 + 1$, but this leads to the contradiction that $w(\boldsymbol{c} + \boldsymbol{c}') = w(\boldsymbol{c}) + w(\boldsymbol{c}') - 2w(\boldsymbol{c} \cap \boldsymbol{c}') \leq d - 2$.

For $\boldsymbol{c} \in A_d(C)$, if $|LH(\boldsymbol{c})| > |A_{d+1}(C)|$ then there exist at least one larger half $\boldsymbol{v}$ in $LH(\boldsymbol{c}) \setminus LH^-(A_{d+1}(C)) = LH(\boldsymbol{c}) \setminus LH(C \setminus \{\boldsymbol{0}, \boldsymbol{c}\})$. Thus $\boldsymbol{c} \in T_{\mathrm{nec}}$ from (4.2). Since $|LH(\boldsymbol{c})| = \binom{d}{(d+1)/2}$ for every $\boldsymbol{c} \in A_d(C)$, if $\binom{d}{(d+1)/2} > |A_{d+1}(C)|$ then $A_d(C) \subseteq T_{\mathrm{nec}}$. □

For a codeword $\boldsymbol{c} \in A_{d+1}(C)$, we have

$$\boldsymbol{c} \in T_{\mathrm{nec}} \text{ if } LH^-(\boldsymbol{c}) \setminus LH(C^* \setminus \{\boldsymbol{c}\}) \neq \emptyset. \tag{4.4}$$

A sufficient condition under which all the codewords of weights d and $d+1$ are in any trial set is given as follows.

Theorem 6. *Let C be a linear code with odd minimum distance d. Then $A_d(C) \cup A_{d+1}(C) \subseteq T_{\mathrm{nec}}$ holds if*

$$\binom{d}{\frac{d+1}{2}} > |A_d(C)| + |A_{d+1}(C)| - 1. \tag{4.5}$$

Proof. From the proof of Theorem 5 we have $|LH(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)| \leq 1$ for $\boldsymbol{c}_1 \in A_d(C)$ and $\boldsymbol{c}_2 \in A_{d+1}(C)$. Here we show $|LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)| \leq 1$ for $\boldsymbol{c}_1, \boldsymbol{c}_2 \in A_{d+1}(C)$. For contradiction, suppose there exist two distinct vectors in $LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)$. Then it must hold that $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \geq (d+1)/2 + 1$, but this leads to the contradiction that $w(\boldsymbol{c}_1 + \boldsymbol{c}_2) = w(\boldsymbol{c}_1) + w(\boldsymbol{c}_2) - 2w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \leq d - 1$.

For $\boldsymbol{c} \in A_{d+1}(C)$, if $|LH^-(\boldsymbol{c})| > |A_d(C)| + |A_{d+1}(C) \setminus \{\boldsymbol{c}\}|$ then there exist at least one larger half $\boldsymbol{v}$ in $LH^-(\boldsymbol{c}) \setminus \{LH(A_d(C)) \cup LH^-(A_{d+1}(C)\} = LH^-(\boldsymbol{c}) \setminus LH(C \setminus \{\boldsymbol{0}, \boldsymbol{c}\})$. Thus $\boldsymbol{c} \in T_{\mathrm{nec}}$ from (4.4). Since $|LH^-(\boldsymbol{c})| = \binom{d}{(d+1)/2}$ for every $\boldsymbol{c} \in A_{d+1}(C)$, if $\binom{d}{(d+1)/2} > |A_d(C)| + |A_{d+1}(C)| - 1$ then $A_d(C) \cup A_{d+1}(C) \subseteq T_{\mathrm{nec}}$. □

4.4.2 Even Minimum Weight Case

Define the set of leftmost coordinates of codewords in C;

$$S(C) = \{l(\boldsymbol{c}) : \boldsymbol{c} \in C\}.$$

For $i \in S(C)$, let

$$C(i) = \{\boldsymbol{c} \in C : l(\boldsymbol{c}) = i\}.$$

When d is even, the weight of the vectors in $LH^-(\boldsymbol{c})$ for $\boldsymbol{c} \in A_d(C)$ is $d/2$, and the weight $d/2$ is the minimum weight of the uncorrectable errors. Hence, the vectors in $LH^-(\boldsymbol{c})$ are minimal uncorrectable errors. Thus we have

$$\boldsymbol{c} \in T_{\text{nec}} \text{ if } LH^-(\boldsymbol{c}) \setminus LH(C^* \setminus \{\boldsymbol{c}\}) \neq \emptyset \tag{4.6}$$

for $\boldsymbol{c} \in A_d(C)$ with even minimum distance d.

For a code with even minimum distance d, a sufficient condition under which all the codewords of d are in any trial set is given.

Theorem 7. *Let C be a linear code with even minimum distance d. Then $A_d(C) \subseteq T_{\text{nec}}$ holds if*

$$\frac{1}{2}\binom{d}{\frac{d}{2}} > \max_{i \in S(C)} \left\lceil \frac{|A_d(C) \cap C(i)| - 1}{2} \right\rceil \tag{4.7}$$

or

$$\frac{1}{2}\binom{d}{\frac{d}{2}} > \left\lceil \frac{|A_d(C)| - 1}{2} \right\rceil. \tag{4.8}$$

Proof. First we note that, for two codewords $\boldsymbol{c}_1, \boldsymbol{c}_2 \in A_d(C)$, the number of common larger half of weight $d/2$ they have is at most one. That is, $|LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)| \leq 1$. For contradiction, suppose there exist two distinct vectors in $LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)$. Then it must hold that $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \geq d/2 + 1$, but this leads to the contradiction that $w(\boldsymbol{c}_1 + \boldsymbol{c}_2) = w(\boldsymbol{c}_1) + w(\boldsymbol{c}_2) - 2w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \leq d - 1$. Furthermore, when $|LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)| = 1$, the vector $\boldsymbol{v} \in LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)$ is represented as $\boldsymbol{v} = \boldsymbol{c}_1 \cap \boldsymbol{c}_2$ and it holds that $l(\boldsymbol{c}_1) = l(\boldsymbol{c}_2)$.

Let $\boldsymbol{c} \in A_d(C)$ and $l(\boldsymbol{c}) = i$. If $|LH^-(\boldsymbol{c})|$ is greater than the number of codewords $\boldsymbol{c}' \in A_d(C) \setminus \{\boldsymbol{c}\}$ of $l(\boldsymbol{c}') = i$, that is, $|LH^-(\boldsymbol{c})| > |A_d(C) \cap C(i)| - 1$, then it holds that $LH^-(\boldsymbol{c}) \setminus LH^-(C \setminus \{\boldsymbol{0}, \boldsymbol{c}\}) = LH^-(\boldsymbol{c}) \setminus LH^-(A_d(C) \cap C(i) \setminus \{\boldsymbol{c}\}) \neq \emptyset$ by counting argument. In this case, $\boldsymbol{c} \in T_{\text{nec}}$ from (4.6). The condition can be improved to $|LH^-(\boldsymbol{c})| > \lceil (|A_d(C) \cap C(i)| - 1)/2 \rceil$ because if $\boldsymbol{c}$ has the common larger half of weight $d/2$ with $\boldsymbol{c}' \in A_d(C)$ then the other codeword $\boldsymbol{c} + \boldsymbol{c}' \in A_d(C)$ does not have common larger halves with $\boldsymbol{c}$, since $l(\boldsymbol{c} + \boldsymbol{c}') \neq l(\boldsymbol{c})$.

From (3.1)–(3.3), the size of $LH^-(\boldsymbol{c})$ for $\boldsymbol{c} \in A_d(C)$ is $\binom{d}{d/2}/2$. Therefore if $\binom{d}{d/2}/2 > \max_{i \in S(C)} \lceil (|A_d(C) \cap C(i)| - 1)/2 \rceil$ holds, any codeword in $A_d(C)$ should be in T_{nec}. Since the value $\max_{i \in S(C)} |A_d(C) \cap C(i)|$ is upper bounded by $|A_d(C)|$, the condition (4.8) follows. □

The condition (4.7) is stronger than (4.8). Therefore, if the distribution of leftmost coordinates of minimum weight codewords is known, it is better to use (4.7). However it is often that only the number of minimum weight codewords is known, so the condition (4.8) is useful.

4.5 Uncorrectable Error Estimation for Half the Minimum Distance

Under the sufficient conditions stated in the previous section, lower bounds on the number of uncorrectable errors of weight half the minimum distance are derived. The corresponding upper bounds are also given unconditionally.

First we give the bound for codes with odd minimum distance.

Theorem 8. *Let C be a linear code with odd minimum distance d. If*

$$\binom{d}{\frac{d+1}{2}} > |A_d(C)| + |A_{d+1}(C)| - 1$$

holds, then

$$\binom{d}{\frac{d+1}{2}}(|A_d(C)| + |A_{d+1}(C)|) - (2|A_d(C)| + |A_{d+1}(C)| - 1)|A_{d+1}(C)| \le |E^1_{\frac{d+1}{2}}(C)| \le \binom{d}{\frac{d+1}{2}}(|A_d(C)| + |A_{d+1}(C)|).$$

Proof. First we observe that $E^1_{(d+1)/2}(C) = LH(A_d(C)) \cup LH^-(A_{d+1}(C))$. From Lemma 13 we have $|LH(\boldsymbol{c}_1) \cap LH(\boldsymbol{c}_2)| = 0$ for $\boldsymbol{c}_1, \boldsymbol{c}_2 \in A_d(C)$. From the proof of Theorem 5 we have $|LH(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)| \le 1$ for $\boldsymbol{c}_1 \in A_d(C)$ and $\boldsymbol{c}_2 \in A_{d+1}(C)$. From the proof of Theorem 6 we have $|LH^-(\boldsymbol{c}_1) \cap LH^-(\boldsymbol{c}_2)| \le 1$ for $\boldsymbol{c}_1, \boldsymbol{c}_2 \in A_{d+1}(C)$.

Since a codeword $\boldsymbol{c} \in A_d(C)$ has at most one common larger half for every $\boldsymbol{c}' \in A_{d+1}(C)$ and does not have common larger halves for any $\boldsymbol{c}' \in A_d(C) \setminus \{\boldsymbol{c}\}$, at least $|LH(\boldsymbol{c})| - |A_{d+1}(C)|$ vectors in $LH(\boldsymbol{c})$ does not have common larger halves. Since a codeword $\boldsymbol{c} \in A_{d+1}(C)$ has at most one common larger half for every $\boldsymbol{c}' \in A_d(C) \cup$

$\{A_{d+1}(C) \setminus \{\boldsymbol{c}\}\}$, at least $|LH^-(\boldsymbol{c})| - |A_d(C)| - |A_{d+1}(C)| + 1$ vectors in $LH^-(\boldsymbol{c})$ does not have common larger halves.

For every $\boldsymbol{c}_1 \in A_d(C)$ and $\boldsymbol{c}_2 \in A_{d+1}(C)$, we have $|LH(\boldsymbol{c}_1)| = |LH^-(\boldsymbol{c}_2)| = \binom{d}{(d+1)/2}$. Therefore we have the lower bound $(\binom{d}{(d+1)/2} - |A_{d+1}(C)|)|A_d(C)| + (\binom{d}{(d+1)/2} - |A_d(C)| - |A_{d+1}(C)| + 1)|A_{d+1}(C)|$.

The upper bound is obtained from the inequality $|E^1_{(d+1)/2}(C)| \leq |LH(A_d(C))| + |LH^-(A_{d+1}(C))| \leq \binom{d}{(d+1)/2}|A_d(C)| + \binom{d}{(d+1)/2}|A_{d+1}(C)|$. □

The difference between the upper and lower bounds is $(2|A_d(C)| + |A_{d+1}(C)| - 1)|A_{d+1}(C)|$. If the fraction $|A_{d+1}(C)|/\binom{d}{(d+1)/2}$ tends to zero as the code length becomes large, the lower bound comes close to the upper one.

Next we give the bound for codes with even minimum distance. The condition in the next theorem is the same as in the previous section.

Theorem 9. *Let C be a linear code with even minimum distance d. If*

$$\frac{1}{2}\binom{d}{\frac{d}{2}} > \left\lceil \frac{|A_d(C)| - 1}{2} \right\rceil$$

holds, then

$$\frac{1}{2}\binom{d}{\frac{d}{2}}|A_d(C)| - \left\lceil \frac{|A_d(C)| - 1}{2} \right\rceil |A_d(C)| \leq |E^1_{\frac{d}{2}}(C)| \leq \frac{1}{2}\binom{d}{\frac{d}{2}}|A_d(C)|.$$

Proof. First we observe that $E^1_{d/2}(C) = LH^-(A_d(C))$. From the proof of Theorem 7, if $\binom{d}{d/2} > |A_d(C)|$ holds then, for every $\boldsymbol{c} \in A_d(C)$, at least $|LH^-(\boldsymbol{c})| - \lceil(|A_d(C)| - 1)/2\rceil$ vectors in $LH^-(\boldsymbol{c})$ does not have common larger halves. Thus we have the lower bound $(\binom{d}{d/2}/2 - \lceil(|A_d(C)| - 1)/2\rceil)|A_d(C)|$.

The upper bound is obtained from the inequality $|LH^-(A_d(C))| \leq \binom{d}{d/2}|A_d(C)|$. □

The difference between the upper and lower bounds is upper bounded by $|A_d(C)|^2/2$. If the fraction $|A_d(C)|/\binom{d}{d/2}$ tends to zero as the code length becomes large, the lower bound comes close to the upper one.

In what follows, we see some BCH codes, Reed-Muller codes, and random linear codes satisfy the sufficient conditions under which T_{nec} contains the minimum weight codewords and thus can be applied to the upper and lower bounds derived in this section.

Primitive BCH codes

By using the weight distribution [15], we can verify that the (n, k) primitive BCH codes satisfy the conditions (4.3) and (4.5) for $n = 127, k \leq 64$ and $n = 63, k \leq 24$.

Table 4.2: The r-th order Reed-Muller code of length 2^m satisfying (4.8).

r	m
1	≥ 4
2	≥ 6
3	≥ 8
4	≥ 10
5	≥ 11
6	≥ 13

Extended Primitive BCH codes

By using the weight distribution [15], we can verify that the (n, k) extended primitive BCH codes satisfy the condition (4.8) for $n = 128, k \leq 64$ and $n = 64, k \leq 24$.

Reed-Muller codes

For the r-th order Reed-Muller code of length 2^m, the minimum distance is 2^{m-r} and the number of minimum weight codewords $|A_{2^{m-r}}(\mathrm{RM}_{m,r})|$ is presented in Theorem 9 of [26, Chapter 13], which is upper bounded by $(2^{m+1} - 2)^r$. Then, for a fixed r, the condition (4.8) is satisfied except for small m. Table 4.2 shows which parameters meets the condition (4.8).

The fraction $|A_d(C)|/\binom{d}{d/2}$ is upper bounded by

$$\frac{|A_d(C)|}{\binom{d}{d/2}} \leq \frac{(2^{m+1} - 2)^r}{2^{2^{m-r}}} \leq 2^{(m+1)r - 2^{m-r}}.$$

Thus for a fixed r the fraction tends to zero as m becomes large. This means the upper and lower bounds in Theorem 9 asymptotically coincide.

Random Linear Codes

A random linear code is a code whose generator matrix has equiprobable entries. That is, first we set a parameter (n, k), and then we choose a generator matrix from all the 2^{nk} possible generator matrices with probability 2^{-nk}. It is known that with high probability the minimum distance equals to $n\delta_{\mathrm{GV}}$, where $1 - H(\delta_{\mathrm{GV}}) = k/n$ and $H(x)$ is the binary entropy function of x. Also it is known that the weight distribution equals to the binomial distribution. Then,

$$|A_d(C)| \approx (2^k - 1)\binom{n}{d}2^{-n} \approx \binom{n}{n\delta_{\mathrm{GV}}}2^{k-n} \approx 2^{n(H(\delta_{\mathrm{GV}})+k/n-1)} \approx 1,$$

where we use the approximation $\binom{n}{n\lambda} \approx 2^{H(\lambda)}$, and

$$|A_{d+1}(C)| \approx (2^k - 1)\binom{n}{d+1}2^{-n} \approx \frac{n-d}{d+1}\binom{n}{n\delta_{\mathrm{GV}}}2^{k-n} \approx 2^{n(H(\delta_{\mathrm{GV}})+k/n-1)} \approx 1.$$

Since

$$\binom{d}{\frac{d}{2}} \approx \sqrt{\frac{2}{\pi d}}2^d \approx 2^{n\delta} \qquad \text{for even } d,$$
$$\binom{d}{\frac{d+1}{2}} \approx \frac{1}{\sqrt{2\pi(d+1)}}2^{d+1} \approx 2^{n\delta} \qquad \text{for odd } d,$$

where $d = n\delta$, the conditions (4.5) and (4.8) are satisfied. Since the fractions $|A_{d+1}(C)|/\binom{d}{(d+1)/2}$ and $|A_d(C)|/\binom{d}{d/2}$ tend to zero, the upper and lower bounds in Theorems 8 and 9 asymptotically coincide.

4.6 Concluding Remarks

A lower bound on the number of uncorrectable errors of weight half the minimum distance has been derived for general linear codes. The conditions for the bounds are not too restrictive, since Reed-Muller codes and random linear codes meet them. Although trial sets does not appear in the theorems, they are the underlying idea for the results. A key observation for the results is that an uncorrectable error of weight half the minimum distance is a larger half of some minimum weight codeword. An uncorrectable error of weight greater than half the minimum distance may be a non-minimal uncorrectable error, that is, such an error is not necessarily a larger half of some codeword. Therefore, it seems difficult to generalize the bounds to the weight greater than half the minimum distance.

Chapter 5

Relations Between Local Weight Distributions

5.1 Introduction

The local weight distribution is the weight distribution of the minimal codewords. Studies on minimal codewords in a linear code are crucial for the performance analysis of the code under ML decoding. For example, the local weight distribution gives a tighter upper bound on the error probability over AWGNC than the usual union bound [17]. Minimal codeword appears in minimum distance decoding algorithms, so called *gradient-like decoding* [4, 23]. The number of minimal codewords in a code determines the complexity of the decoding of the code. In the context of cryptography, Massey [27] showed that the access structure of a secret sharing scheme determined by a linear code is characterized by minimal codewords in the dual code.

Agrell [1] showed an efficient method of examining the minimality of a codeword for a binary linear code and computed the local weight distributions by examining all the codewords for some codes. Ashikhmin and Barg [3] determined the local weight distributions for Hamming codes, extended Hamming codes, second-order Reed-Muller codes. Partial results for the local weight distributions of Reed-Muller codes are given in [9]. Asymptotic analysis for long codes and random codes is given in [2, 3]. Mohri et al. [29, 28] proposed the computational algorithms for cyclic codes. The number of codewords to be examined is reduced in their work. The basic idea for the reduction was suggested in [1]. Using the algorithms, they determined the local weight distributions of all the primitive BCH codes of length 63.

In this chapter, relations between the local weight distributions of a code, its extended code, and its even weight subcode are studied. For a code that contains codewords of weight only multiples of four, the local weight distributions of the extended code and the even weight subcode are determined from that of the original code. Furthermore, if the extended code is transitive invariant, the local weight distribution of the original code is obtained from that of the extended code.

5.2 Known Results and Applications

The definition of local weight distribution is presented in Section 2.1.5. First we give the following relation between the weight distribution and the local weight distribution.

Proposition 2 ([2, 3]). *For an (n, k, d) linear code C,*

$$|L_i(C)| = \begin{cases} |A_i(C)| & \textit{for} \;\; i < 2d, \\ 0 & \textit{for} \;\; i > n - k + 1. \end{cases}$$

If the weight distribution $(|A_0(C)|, |A_1(C)|, \ldots, |A_n(C)|)$ is known, only $|L_w(C)|$ with $2d \leq w \leq n - k + 1$ needs to be computed to obtain the local weight distribution. Generally the complexity of computing the local weight distribution is larger than that of computing the weight distribution. Therefore, Proposition 2 is useful for obtaining local weight distributions. When every weight i in a code satisfies $i < 2d$ or $i > n - k + 1$, the local weight distribution can be obtained from the weight distribution straightforwardly. For example, the local weight distribution of the (n, k) primitive BCH code of length 63 for $k \leq 18$, of length 127 for $k \leq 29$, and of length 255 for $k \leq 45$ can be obtained from their weight distributions.

5.2.1 Known Results

We present known results of the local weight distributions.

Hamming codes

Let C be a $(2^m - 1, 2^m - 1 - m, 3)$ Hamming code and C_{ex} be the $(2^m, 2^m - 1 - m, 4)$ extended Hamming code of C. The local weight distributions of C and C_{ex} are given in [3].

$$|L_i(C)| = \begin{cases} \dfrac{1}{i!} \displaystyle\prod_{j=0}^{i-2}(2^m - 2^j) & \text{for } 3 \le i \le m+1, \\ 0 & \text{for } 0 \le i < 3, \quad m+1 < i \le 2^m - 1. \end{cases}$$

$$|L_i(C_{\text{ex}})| = \begin{cases} \dfrac{1}{i!}\, 2^m \displaystyle\prod_{j=0}^{i-3}(2^m - 2^j) & \text{for } 4 \le i \le m+2, \\ 0 & \text{for } 0 \le i < 4, \quad m+2 < i \le 2^m. \end{cases}$$

Reed-Muller codes

The local weight distribution of the second-order Reed-Muller code $\mathrm{RM}_{m,2}$ is presented in [3].

$$|L_i(\mathrm{RM}_{m,2})| = \begin{cases} 0 & \text{for } i = 2^{m-1} + 2^{m-1-h} \ (h = 0,1,2) \\ & \quad \text{or } i > 2^m - k + 1, \\ |A_i(\mathrm{RM}_{m,2})| - 2^{m+1} + 2 & \\ \qquad -(2^{m-1} - 2)|A_{2^{m-2}}(\mathrm{RM}_{m,2})| & \text{for } i = 2^{m-1}, \\ |A_i(\mathrm{RM}_{m,2})| & \text{otherwise,} \end{cases}$$

where

$$|A_{2^{m-1} \pm 2^{m-1-i}}(\mathrm{RM}_{m,2})| = 2^{i(i+1)} \cdot \frac{(2^m - 1)(2^{m-1} - 1)(2^{m-2i+1} - 1)}{(2^{2i} - 1)(2^{2i-2} - 1)\cdots(2^2 - 1)} \quad \text{for } 1 \le i \le \left[\frac{1}{2}m\right],$$
$$|A_{2^{m-1}}(\mathrm{RM}_{m,2})| = 2^{1+m+\binom{m}{2}} - \sum_{j \ne 2^{m-1}} |A_j(\mathrm{RM}_{m,2})|.$$

Borissov et al. [9] derived partial results for the local weight distributions of $\mathrm{RM}_{m,r}$. They showed that $|L_{2^m - 2^{m-r+1}}(\mathrm{RM}_{m,r})| = 0$ for $m \ge 3$ and that

$$|A_{2^{m-r+1}}(\mathrm{RM}_{m,r})| - |L_{2^{m-r+1}}(\mathrm{RM}_{m,r})| = B^1_{m,r} + B^2_{m,r} + B^3_{m,r} - B^4_{m,r},$$

where

$$B^1_{m,r} = 2^{r-1} \begin{bmatrix} m \\ m-r+1 \end{bmatrix},$$
$$B^2_{m,r} = \frac{2^{r+1} - 4}{4} \begin{bmatrix} m \\ m-r+1 \end{bmatrix} \binom{2^{r+1}}{3},$$

$$
\begin{aligned}
B^3_{m,r} &= 2^{r-1} \begin{bmatrix} m \\ m-r \end{bmatrix} \left(2^r \begin{bmatrix} m \\ m-r \end{bmatrix} - {B^3_{m,r}}' \right), \\
{B^3_{m,r}}' &= \sum_{l=\max\{0,m-2r\}}^{m-r} 2^{(m-r-l)(m-r-l+1)} \begin{bmatrix} m-r \\ l \end{bmatrix} \begin{bmatrix} r \\ m-r-1 \end{bmatrix}, \\
B^4_{m,r} &= 2^{r-1}(2^{m-r+1}-1) \begin{bmatrix} m \\ m-r+1 \end{bmatrix} \\
&\quad + \frac{1}{8} \cdot 2^{r+1}(2^{r+1}-1)(2^{r+1}-2)(2^{r+1}-4) \begin{bmatrix} m \\ m-r+1 \end{bmatrix},
\end{aligned}
$$

and $\begin{bmatrix} m \\ i \end{bmatrix}$ is the *2-ary Gaussian coefficient*, defined by

$$
\begin{bmatrix} m \\ i \end{bmatrix} = \begin{cases} 1 & \text{for } \ i = 0, \\ \displaystyle\prod_{j=0}^{i-1} \frac{2^m - 2^i}{2^i - 2^j} & \text{for } \ i = 1, 2, \ldots, m. \end{cases}
$$

BCH codes

The local weight distributions of the primitive BCH codes of length 63 are presented in [29, 28].

Random codes

We consider the ensemble of random linear codes whose parity check matrices have equiprobable entries. Let $\mathrm{E}[|L_i(C)|]$ be the average number of minimal codewords of weight i taken over the ensemble of random linear codes. Then from [3] we have that

$$
\mathrm{E}[|L_i(C)|] = \begin{cases} \displaystyle\binom{n}{i} \frac{1}{2^{n-k}} \prod_{j=0}^{i-2} (1 - 2^{-(n-k-j)}) & \text{for } \ i \le n-k+1, \\ 0 & \text{otherwise.} \end{cases}
$$

5.2.2 Upper bounds on the Error Probability Using LWDs

This section provides some application of the local weight distribution.

First we show how the local weight distribution gives a tighter upper bound on the error probability over AWGNC. Recall that a codeword $\boldsymbol{c}$ is transmitted as $(s(c_1), \ldots, s(c_n))$ by using a mapping function $s(\cdot)$ defined in Section 2.1.2. Let $s(\boldsymbol{c})$ denote the transmitted

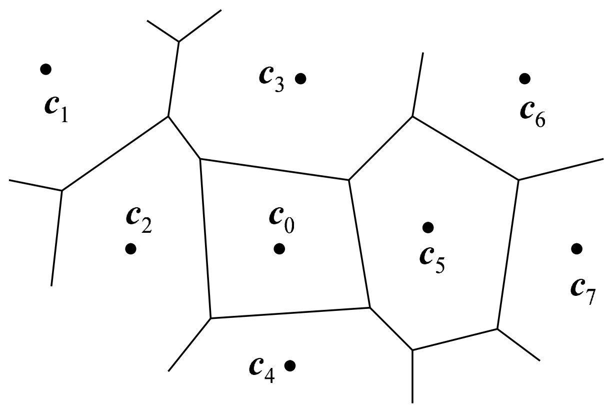


Figure 5.1: Eight vectors in $\mathbb{R}^n$. $\boldsymbol{c}_0$ is the all-zero codeword. The codewords $\boldsymbol{c}_2, \boldsymbol{c}_3, \boldsymbol{c}_4, \boldsymbol{c}_5$ are zero neighbors. The codewords $\boldsymbol{c}_1, \boldsymbol{c}_6, \boldsymbol{c}_7$ are not zero neighbors.

vectors on AWGNC of $\boldsymbol{c}$. The set $S = \{s(\boldsymbol{c}) : \boldsymbol{c} \in C\}$ is called a signal set. We define the Voronoi region of codewords over $\mathbb{R}^n$.

Definition 1 (Voronoi region)**.** *The Voronoi region of $\mathbf{c} \in C$ is a set of closest vectors in $\mathbb{R}^n$ to $\mathbf{c}$, that is,*

$$\{\boldsymbol{x} \in \mathbb{R}^n : d_E(\boldsymbol{x}, \boldsymbol{c}) \leq d_E(\boldsymbol{x}, \boldsymbol{c}') \ \textit{ for all } \ \mathbf{c}' \in C \setminus \{\boldsymbol{c}\}\},$$

where $d_E(\boldsymbol{x}, \boldsymbol{y})$ is the squared Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^n$.

The shape of the Voronoi region determines almost all significant properties for communications [17]. If the signal set S is *geometrically uniform* [17], all the Voronoi regions have the same shape. A signal set of a binary linear code with BPSK modulation is *geometrically uniform.* Then, for a binary linear code C, the Voronoi region of the all-zero codeword can be used as the representative of all codewords in C. A codeword whose Voronoi region shares the facet with that of the all-zero codeword is said to be a *zero neighbor* (refer to Figure 5.1).

Definition 2 (Zero neighbor)**.** *For $\boldsymbol{c} \in C$, define $\boldsymbol{m}_0 \in \mathbb{R}^n$ as $\boldsymbol{m}_0 = \frac{1}{2}(s(\mathbf{0}) + s(\boldsymbol{c}))$. The codeword $\boldsymbol{c}$ is a zero neighbor if and only if*

$$d_E(\boldsymbol{m}_0, s(\boldsymbol{c})) = d_E(\boldsymbol{m}_0, s(\mathbf{0})) < d_E(\boldsymbol{m}_0, s(\boldsymbol{c}')) \quad \textit{for all } \ \mathbf{c}' \in C \setminus \{\mathbf{0}, \boldsymbol{c}\}.$$

Agrell [1] made the following observation.

Proposition 3. *For a binary linear code C,*

$$\boldsymbol{c} \text{ is a zero neighbor in } C \quad \text{if and only if} \quad \boldsymbol{c} \text{ is a minimal codeword in } C.$$

Let's consider the error probability P_e of C over AWGNC after ML decoding. Since C is a linear code, we can assume that the all-zero codeword $\mathbf{0}$ is transmitted. Thus

$$P_e = \Pr\left[\bigcup_{\boldsymbol{c}\in C\setminus\{\mathbf{0}\}} \mathcal{E}_{\mathbf{0},\boldsymbol{c}}\right] \tag{5.1}$$

$$\leq \sum_{\boldsymbol{c}\in C\setminus\{\mathbf{0}\}} \Pr\left[\mathcal{E}_{\mathbf{0},\boldsymbol{c}}\right], \tag{5.2}$$

where $\mathcal{E}_{\mathbf{0},\boldsymbol{c}}$ denotes an event that $\mathbf{0}$ is transmitted and the decoding result of the decoder is $\boldsymbol{c}$, as defined in Section 2.1.2. The inequality (5.2) is called a union upper bound of P_e. We observe that $\mathcal{E}_{\mathbf{0},\boldsymbol{c}} = \{\boldsymbol{r} \in \mathbb{R}^n : d_E(\boldsymbol{r}, s(\boldsymbol{c})) \leq d_E(\boldsymbol{r}, s(\mathbf{0}))\}$. Then the union bound of P_e using the weight distribution of C is written [14] as

$$\begin{aligned} P_e &\leq \sum_{\boldsymbol{v}\in C\setminus\{\mathbf{0}\}} Q\left(\sqrt{w(\boldsymbol{c})\frac{E_b}{\sigma^2}}\right) \\ &= \sum_{i=1}^{n} A_i(C)\, Q\left(\sqrt{i\frac{E_b}{\sigma^2}}\right), \end{aligned} \tag{5.3}$$

where $Q(x)$ is the complementary error function; $Q(x) = \int_x^\infty (2\pi)^{-1/2} \exp(-z^2/2)dz$, E_b is the bit energy, and σ^2 is the variance of Gaussian noise.

If $\boldsymbol{c} \subseteq \boldsymbol{c}'$ for $\boldsymbol{c}, \boldsymbol{c}' \in C$, then $\mathcal{E}_{\mathbf{0},\boldsymbol{c}} \supseteq \mathcal{E}_{\mathbf{0},\boldsymbol{c}'}$. Using the local weight distribution of C, (5.1) and (5.2) can be rewritten by

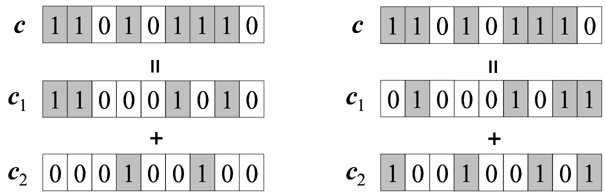
$$P_e = P\left[\bigcup_{\boldsymbol{c}\in C^*} \mathcal{E}_{\mathbf{0},\boldsymbol{c}}\right] \tag{5.4}$$

$$\leq \sum_{\boldsymbol{c}\in C^*} P[\mathcal{E}_{\mathbf{0},\boldsymbol{c}}]. \tag{5.5}$$

Inequality (5.5) is called a minimal union bound [16]. A minimal union bound using the local weight distribution of C is obtained in the same way as (5.3):

$$P_e \leq \sum_{i=1}^{n} L_i(C)\, Q\left(\sqrt{i\frac{E_b}{\sigma^2}}\right). \tag{5.6}$$

The right-hand side of (5.6) is strictly smaller than that of (5.3). Agrell pointed out in [1] that other bounds, related to the union bound, such as Berlekamp's tangential



(a) Decomposable (b) Indecomposable

Figure 5.2: Examples of a decomposable codeword and an indecomposable codeword.

union bound [6], seem to be improved in a similar way. For many bounds on the error probability after ML decoding, see [35, 49] and references therein.

The set of minimal codewords can be used for a minimum distance decoding [4, 23]. The algorithm appears in an optimal hard decision decoding algorithms [23]. The number of minimal codewords in a code determines the complexity of the decoding. This decoding method is a type of *gradient-like decoding* [4]. See [4] for details.

Minimal codewords in a linear code have a link to secret-sharing schemes using error-correcting codes. Massey showed that the set of minimal codewords in the dual code completely specifies the access structure of the secret-sharing scheme [27].

5.3 LWDs of Extended Codes and Even Weight Subcodes

In this section, we consider relations between the local weight distributions of a code C of length n, its extended code C_{ex}, and its even weight subcode C_{even}. For a codeword $\boldsymbol{c} \in C$, let $\boldsymbol{c}^{(\mathrm{ex})}$ be the corresponding extended codeword in C_{ex}. We define a *decomposable* codeword (see Fig. 5.2).

Definition 3 (Decomposable codeword)**.** *$\boldsymbol{c} \in C$ is called decomposable if $\boldsymbol{c}$ can be represented as $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$ where $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C$ and $\boldsymbol{c}_1 \cap \boldsymbol{c}_2 = \mathbf{0}$.*

Clearly, $\boldsymbol{c}$ is not a minimal codeword if and only if $\boldsymbol{c}$ is decomposable. For even weight codewords, we introduce an *only-odd-decomposable* codeword and an *even-decomposable* codeword.

Definition 4. *Let $\boldsymbol{c} \in C$ be a decomposable codeword with even $w(\boldsymbol{c})$. Then $\boldsymbol{c}$ is said to be only-odd-decomposable if all the decompositions of $\boldsymbol{c}$ are of the form $\boldsymbol{c}_1 + \boldsymbol{c}_2$ with the odd weight codewords $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C$. Otherwise, $\boldsymbol{c}$ is said to be even-decomposable.*

Table 5.1: Minimality of $\boldsymbol{c}$ in a linear block code, $\boldsymbol{c}^{(\mathrm{ex})}$ in its extended code, and $\boldsymbol{c}$ in its even weight subcode.

$\boldsymbol{c}$ in C			$\boldsymbol{c}^{(\mathrm{ex})}$ in C_{ex}		$\boldsymbol{c}$ in C_{even}	
Minimal	Weight	Decomposability	Minimal	Lemma 15	Minimal	Lemma 16
Yes	Odd	Not decomp.	Yes	(1)	N/A	N/A
	Even				Yes	(1)
No	Odd	Decomp.	No	2-(a)	N/A	N/A
	Even	Only-odd-decomp.	Yes	2-(b)	Yes	(2)
	Even	Even-decomp.	No		No	

When $\boldsymbol{c}$ is even-decomposable, there is a decomposition of $\boldsymbol{c}$, $\boldsymbol{c}_1 + \boldsymbol{c}_2$, such that both $w(\boldsymbol{c}_1)$ and $w(\boldsymbol{c}_2)$ are even. Then $\boldsymbol{c}^{(\mathrm{ex})}$ is decomposable into $\boldsymbol{c}_1^{(\mathrm{ex})} + \boldsymbol{c}_2^{(\mathrm{ex})}$. On the other hand, for an only-odd-decomposable codeword $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$, $\boldsymbol{c}^{(\mathrm{ex})}$ is not decomposable into $\boldsymbol{c}_1^{(\mathrm{ex})} + \boldsymbol{c}_2^{(\mathrm{ex})}$ for any decompositions.

The relation between C and C_{ex} with respect to minimality is given in the following theorem, which is also summarized in Table 5.1.

Lemma 15. *1. For a minimal codeword $\boldsymbol{c}$ in C, $\boldsymbol{c}^{(\mathrm{ex})}$ is a minimal codeword in C_{ex}.*

2. For a codeword $\boldsymbol{c}$ which is not a minimal codeword in C, the following (a) and (b) hold:

(a) When $w(\boldsymbol{c})$ is odd, $\boldsymbol{c}^{(\mathrm{ex})}$ is not a minimal codeword in C_{ex}.

(b) When $w(\boldsymbol{c})$ is even, $\boldsymbol{c}^{(\mathrm{ex})}$ is a minimal codeword in C_{ex} if and only if $\boldsymbol{c}$ is only-odd-decomposable in C.

Proof. 1. Suppose that $\boldsymbol{c}^{(\mathrm{ex})}$ is not a minimal codeword in C_{ex}. Then $\boldsymbol{c}^{(\mathrm{ex})}$ is decomposable into $\boldsymbol{c}_1^{(\mathrm{ex})}+\boldsymbol{c}_2^{(\mathrm{ex})}$. Hence, $\boldsymbol{c}$ is decomposable into $\boldsymbol{c}_1+\boldsymbol{c}_2$, contradicting the indecomposability of $\boldsymbol{c}$.

2. Suppose that $\boldsymbol{c}$ is decomposed into $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$. (a) Since $w(\boldsymbol{c})$ is odd, the sum of the parity bits in $\boldsymbol{c}_1^{(\mathrm{ex})}$ and $\boldsymbol{c}_2^{(\mathrm{ex})}$ is one. Also, the parity bit in $\boldsymbol{c}^{(\mathrm{ex})}$ is one. Then $\boldsymbol{c}^{(\mathrm{ex})}$ is decomposable into $\boldsymbol{c}_1^{(\mathrm{ex})}+\boldsymbol{c}_2^{(\mathrm{ex})}$, and $\boldsymbol{c}^{(\mathrm{ex})}$ is not a minimal codeword in C_{ex}. (b) Since $w(\boldsymbol{c})$ is even, the parity bit in $\boldsymbol{c}^{(\mathrm{ex})}$ is zero. (If part) Suppose that $\boldsymbol{c}^{(\mathrm{ex})}$ is not a minimal codeword in C_{ex}. Then there exists a decomposition $\boldsymbol{c}^{(\mathrm{ex})} = \boldsymbol{c}_1^{(\mathrm{ex})} + \boldsymbol{c}_2^{(\mathrm{ex})}$. Because the parity bit in

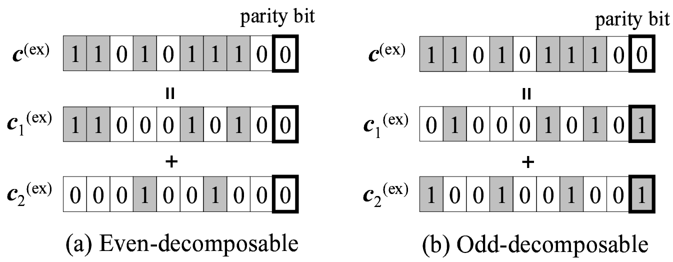


Figure 5.3: Examples of an even-decomposable codeword and an odd-decomposable codeword mentioned in the proof of Lemma 15-2-(b).

$\boldsymbol{c}^{(\mathrm{ex})}$ is zero, the parity bits in $\boldsymbol{c}_1^{(\mathrm{ex})}$ and $\boldsymbol{c}_2^{(\mathrm{ex})}$ must be zero. Thus, $\boldsymbol{c}$ is even-decomposable into $\boldsymbol{c}_1+\boldsymbol{c}_2$, contradicting the assumption that $\boldsymbol{c}$ is only-odd-decomposable. (Only if part) Suppose that $\boldsymbol{c}$ is even-decomposable. Then there is a decomposition such that the parity bits in both $\boldsymbol{c}_1^{(\mathrm{ex})}$ and $\boldsymbol{c}_2^{(\mathrm{ex})}$ are zero. For such a decomposition, $\boldsymbol{c}^{(\mathrm{ex})}$ is decomposable into $\boldsymbol{c}_1^{(\mathrm{ex})} + \boldsymbol{c}_2^{(\mathrm{ex})}$, and $\boldsymbol{c}^{(\mathrm{ex})}$ is not a minimal codeword in C_{ex}. (see Fig. 5.3). □

From 2-(b) of Lemma 15, there may be codewords that are not minimal codewords in C, although their extended codewords are minimal codewords in C_{ex}. Such codewords are the only-odd-decomposable codewords. For investigating relations of local weight distributions between a code and its extended code, only-odd decomposable codewords are important.

Let $N_i(C)$ denote the set of only-odd-decomposable codewords with weight i in C. The following theorem is a direct consequence of Lemma 15.

Theorem 10. *For a code C of length n and an integer i with $0 \le i \le \lfloor n/2 \rfloor$,*

$$|L_{2i}(C_{\mathrm{ex}})| = |L_{2i-1}(C)| + |L_{2i}(C)| + |N_{2i}(C)|. \tag{5.7}$$

From Theorem 10, if no only-odd-decomposable codeword exists in C, then the local weight distributions of C_{ex} are obtained from that of C. Next we give a useful sufficient condition under which no only-odd-decomposable codeword exists.

Theorem 11. *If all the weights of codewords in C_{ex} are multiples of four, then no only-odd-decomposable codeword exists in C.*

Proof. If $\boldsymbol{c} \in C$ is an only-odd-decomposable codeword and is decomposed into $\boldsymbol{c}_1+\boldsymbol{c}_2$, the weights of $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ can be represented as $w(\boldsymbol{c}_1) = 4i-1$ and $w(\boldsymbol{c}_2) = 4j-1$ where i and j are integers. Then $w(\boldsymbol{c}) = w(\boldsymbol{c}_1+\boldsymbol{c}_2) = w(\boldsymbol{c}_1)+w(\boldsymbol{c}_2) = (4i-1)+(4j-1) = 4i+4j-2$, contradicting the fact that $w(\boldsymbol{c})$ is a multiple of four. □

For example, all the weights of codewords in the $(128, k)$ extended primitive BCH code with $k \leq 57$ are multiples of four. The parameters of the Reed-Muller codes with which all the weights of codewords are multiples of four are given by Corollary 13 of Chapter 15 in [26]. From the corollary, the third-order Reed-Muller codes of length $n \geq 128$ have only codewords whose weights are multiples of four.

Although the local weight distribution of C_{ex} for these codes can be obtained from that of C by using Theorem 10, in order to obtain the local weight distribution of C from that of C_{ex}, we need to know the number of minimal codewords with parity bit one. In Section 5.4, we will show a method of obtaining the number of minimal codewords with parity bit one for a class of transitive invariant codes.

A similar relation to that between C and C_{ex} holds between C and C_{even}. This relation is given in Lemma 16 without proof (see Table 5.1).

Lemma 16. *1. For an even weight minimal codeword* $\mathbf{c}$ *in* C*,* $\mathbf{c}$ *is a minimal codeword in* C_{even}*.*

2. For an even weight codeword $\mathbf{c}$ *which is not a minimal codeword in* C*,* $\mathbf{c}$ *is a minimal codeword in* C_{even} *if and only if* $\mathbf{c}$ *is only-odd-decomposable in* C*.*

From Lemma 16, we derive Theorem 12.

Theorem 12. *For a code* C *of length* n *and an integer* i *with* $0 \leq i \leq \lfloor n/2 \rfloor$*,*

$$|L_{2i}(C_{\mathrm{even}})| = |L_{2i}(C)| + |N_{2i}(C)|. \tag{5.8}$$

5.4 LWDs From Transitive Invariant Extended Codes

A transitive invariant code is a code which is invariant under a transitive group of permutations. A group of permutations is said to be transitive if for any two symbols in a codeword there exists a permutation that interchanges them [31]. The extended primitive BCH codes and Reed-Muller codes are transitive invariant codes. For a transitive invariant C_{ex}, a relation between the weight distributions of C and C_{ex} is presented in Theorem 8.15 in [31]. A similar relation holds for local weight distribution.

Lemma 17. *If* C_{ex} *is a transitive invariant code of length* $n+1$*, then the number of minimal codewords in* C *of weight* i *with parity bit one is* $\frac{i}{n+1}|L_i(C_{\mathrm{ex}})|$*.*

Proof. This lemma can be proved in a similar way to the proof of Theorem 8.15. Arrange all minimal codewords with weight i in a column. Next, interchange the j-th column

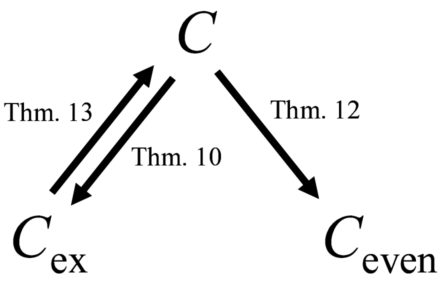


Figure 5.4: The directions of determining the local weight distributions between C, C_{ex}, and C_{even}.

and the last column, which is the parity bit column, for all these codewords with the permutation. All the resulting codewords have weight i and must be the same as the original set of codewords. Thus, the number of ones in the j-th column and that in the last column are the same. Denote this number l_i, which is the same as the number of minimal codewords of weight i with parity bit one. Then the number of total ones in the original set of codewords is $(n+1)\,l_i$, or $|L_i(C_{\mathrm{ex}})|$ times the weight i. Thus, $(n+1)\,l_i = i|L_i(C_{\mathrm{ex}})|$, and $l_i = \frac{i}{n+1}|L_i(C_{\mathrm{ex}})|$. $\square$

It is clear that there are $\frac{n+1-i}{n+1}|L_i(C_{\mathrm{ex}})|$ minimal codewords with weight i whose parity bit is zero from this lemma. The following theorem is obtained from Lemmas 15 and 17.

Theorem 13. *Let C_{ex} be a transitive invariant code of length $n+1$. Then*

$$|L_i(C)| \;=\; \begin{cases} \dfrac{i+1}{n+1}|L_{i+1}(C_{\mathrm{ex}})| & \textit{for odd } i, \\[2ex] \dfrac{n+1-i}{n+1}|L_i(C_{\mathrm{ex}})| - |N_i(C)| & \textit{for even } i. \end{cases} \tag{5.9}$$

Therefore, for a transitive invariant code C_{ex} having no only-odd-decomposable codeword in C, the local weight distributions of C can be obtained from that of C_{ex}. After computing the local weight distribution of C, that of C_{even} can be obtained by using Theorem 12.

5.5 Concluding Remarks

In this chapter, the relations between the local weight distributions of a code, its extended code, and its even weight subcode are revealed. Also the relation between the local weight distributions of an transitive invariant code and its punctured code are investigated. See Figure 5.4 for the directions of determining the local weight distributions.

Borissov and Manev [9] also studied a similar relation. Here we describe the differences. They derived relations between the local weight distributions of a code, its extended code, and its even weight subcode, but they did not clarify the existence of only-odd-decomposable codewords, which play an important role in our results. Therefore, our results in this chapter contain those in [9] about the relations between the distributions, although their results are not the main results of [9].

Chapter 6

Algorithms for Computing Local Weight Distributions

6.1 Introduction

In this chapter, a method for computing the local weight distribution using the automorphism group of the code is presented. The complexity of computing the local weight distribution, as well as that for the weight distribution, becomes large as the dimension of the code is large. Agrell noted in [1] that the automorphism group of codes helps reduce the complexity. Using the automorphism group of cyclic codes, i.e., cyclic permutations, Mohri et al. obtained the local weight distributions of the $(63, k)$ primitive BCH codes for $k \leq 45$ [28, 29]. This invariance property for cyclic permutations can be generalized to an invariance property for any group of permutations. Using the invariance property for a larger group of permutations, we can reduce the number of representative codewords. However, it is not easy to obtain the representative codewords and the number of the equivalent codewords.

In order to use the generalized invariance property, the invariance property is applied to the set of cosets of a subcode rather than the set of codewords. This application reduces the complexity of finding the representatives, which is much smaller than the complexity of checking whether every representative is a minimal codeword. This idea is used in [18] for computing the weight distribution of extended binary primitive BCH codes. In this chapter, we show that this idea can be applicable to computing local weight distribution.

Section 6.2 presents the invariance property and Section 6.3 shows the coset partitioning technique for computing the local weight distribution. These are key ideas of our algorithm. In Section 6.4, we describe the proposed algorithm and its complexity.

In Section 6.5, we improve the algorithm by considering the code tree structure and the invariance property in cosets. We apply the invariance property in cosets to the $(256, 93)$ third-order Reed-Muller code for computing its local weight distribution in Section 6.5.3. The tables of the local weight distributions determined in our work are listed in Section 6.6.

6.2 Invariance Property

The algorithms in [28, 29] uses the following invariance property for cyclic permutations.

Proposition 4 ([28, 29])**.** *Let C be a binary cyclic code. A codeword $\boldsymbol{c} \in C$ is a minimal codeword if and only if any cyclic permuted codeword of $\boldsymbol{c}$ is a minimal codeword.*

Corollary 4. *Let C be a binary cyclic code, and $\sigma^i \boldsymbol{c}$ be an i times cyclic-permuted codeword of $\boldsymbol{c} \in C$. Consider a set $P = \{\boldsymbol{c}, \sigma\boldsymbol{c}, \sigma^2\boldsymbol{c}, \dots, \sigma^{p(\sigma,\boldsymbol{c})-1}\boldsymbol{c}\}$, where $p(\sigma, \boldsymbol{c})$ is the period of σ, which is the minimum i such that $\sigma^i \boldsymbol{c} = \boldsymbol{c}$. Then (1) if $\boldsymbol{c}$ is a minimal codeword, all codewords in the set P are minimal codewords; and otherwise, (2) all codewords in P are not minimal codewords.*

In the algorithms, the representative codeword of cyclic permutations (a representative codeword of P in Corollary 4) and the number of the equivalent codewords (the size of P) are generated efficiently. The complexity is about $1/n$ that of the brute force method. The local weight distributions of the $(63, k)$ primitive BCH codes with $k = 51, 57$ are obtained by using another algorithm [29]. The latter algorithm generates the representative codewords once or more, although the former algorithm generates the representative codewords only once.

The following proposition implies that the algorithms in [28, 29] can be applied to extended cyclic codes straightforwardly.

Proposition 5. *Let C and C_{ex} be a binary cyclic code and its extended code, respectively. For $\boldsymbol{c} \in C$, let $\boldsymbol{c}^{(\text{ex})}$ be the corresponding extended codeword in C_{ex}, that is, $\boldsymbol{c}^{(\text{ex})}$ is obtained from $\boldsymbol{c}$ by adding the over-all parity bit. For any cyclic permuted codeword $\sigma^i\boldsymbol{c}$ of $\boldsymbol{c}$, $(\sigma^i\boldsymbol{c})^{(\text{ex})}$ is a minimal codeword in C_{ex} if and only if $\boldsymbol{c}^{(\text{ex})}$ is a minimal codeword in C_{ex}.*

Proof. (If part) Suppose that $(\sigma^i\boldsymbol{c})^{(\text{ex})}$ is not a minimal codeword in C_{ex}. There exists $\boldsymbol{u} \in C$ such that $(\sigma^i\boldsymbol{u})^{(\text{ex})} \subset (\sigma^i\boldsymbol{c})^{(\text{ex})}$. Then $\boldsymbol{u}^{(\text{ex})} \subset \boldsymbol{c}^{(\text{ex})}$, and this contradicts the fact that $\boldsymbol{c}^{(\text{ex})}$ is a minimal codeword in C_{ex}. (Only if part) Suppose that $\boldsymbol{c}^{(\text{ex})}$ is not a minimal codeword in C_{ex}. There exists $\boldsymbol{u} \in C$ such that $\boldsymbol{u}^{(\text{ex})} \subset \boldsymbol{c}^{(\text{ex})}$. Hence, $(\sigma^i\boldsymbol{u})^{(\text{ex})} \subset (\sigma^i\boldsymbol{c})^{(\text{ex})}$, and this contradicts the fact that $(\sigma^i\boldsymbol{c})^{(\text{ex})}$ is a minimal codeword in C_{ex}. $\square$

From Corollary 4 and Proposition 5, the minimalities of codewords in $P' = \{\boldsymbol{c}^{(\mathrm{ex})}, (\sigma\boldsymbol{c})^{(\mathrm{ex})}, (\sigma^2\boldsymbol{c})^{(\mathrm{ex})}, \ldots, (\sigma^{p(\sigma,\boldsymbol{c})-1}\boldsymbol{c})^{(\mathrm{ex})}\}$ are the same. To compute the local weight distribution of an extended cyclic code C_{ex}, we only have to check minimality for the representative extended codewords of cyclic permutations. Thus, we can compute the local weight distribution of an extended cyclic code in the same way as that in the algorithms in [28, 29] for representative codewords with respect to the cyclic group of permutations. However, extended primitive BCH codes are closed under the affine group of permutations, which are larger than the cyclic group of permutations. Using a larger group of permutations, the complexity of computing the local weight distribution may be reduced. This is a basic observation for the computational approach.

An invariance property of minimality under the automorphism group is given in the following theorem.

Theorem 14 (Invariance property)**.** *Let $\pi \in \mathrm{Aut}(C)$ and $\boldsymbol{c} \in C$. Then $\pi\boldsymbol{c}$ is a minimal codeword in C if and only if $\boldsymbol{c}$ is a minimal codeword in C.*

Proof. Suppose that $\boldsymbol{c}$ is a minimal codeword and $\pi\boldsymbol{c}$ is not a minimal codeword. There exists a nonzero codeword $\boldsymbol{c}' \in C$ such that $\pi\boldsymbol{c} \supset \boldsymbol{c}'$. Since $\mathrm{Aut}(C)$ is a group, there exists $\boldsymbol{c}'' \in C$ such that $\boldsymbol{c}' = \pi\boldsymbol{c}''$. Thus $\pi\boldsymbol{c} \supset \pi\boldsymbol{c}''$, and $\boldsymbol{c} \supset \boldsymbol{c}''$, contradicting the fact that $\boldsymbol{c}$ is a minimal codeword. ☐

This theorem derives the following corollary.

Corollary 5. *For $\boldsymbol{c} \in C$, consider a set $P = \{\pi\boldsymbol{c} : \forall\pi \in \mathrm{Aut}(C)\}$. Then (1) if $\boldsymbol{c}$ is a minimal codeword, then all codewords in S are minimal codewords; otherwise, (2) all codewords in P are not minimal codewords.*

To use this generalized invariance property, we apply the invariance property to the set of cosets of a subcode rather than the set of codewords.

6.3 Coset Partitioning

For a binary (n, k) linear code C and its linear subcode C' with dimension k', let C/C' denote the set of cosets of C' in C, that is, $C/C' = \{\boldsymbol{c} + C' : \boldsymbol{c} \in C \setminus C'\}$. Then

$$|C/C'| = 2^{k-k'} \quad \text{and} \quad C = \bigcup_{D \in C/C'} D. \tag{6.1}$$

Definition 5 (Local weight subdistribution for cosets)**.** *The local weight subdistribution for a coset $D \in C/C'$ (with respect to C) is the weight distribution of minimal codewords of C in D. The local weight subdistribution for D is* $(|LS_0(D)|, |LS_1(D)|, \ldots, |LS_n(D)|)$*, where*

$$LS_i(D) = L_i(C) \cap D,$$

with $0 \leq i \leq n$*.*

Then, from (6.1), the local weight distribution of C is given as the sum of the local weight subdistributions for the cosets in C/C', that is,

$$|L_i(C)| = \sum_{D \in C/C'} |LS_i(D)|.$$

The next theorem gives an invariance property under permutations for cosets.

Theorem 15 (Invariance property for cosets)**.** *For $D_1, D_2 \in C/C'$, the local weight subdistribution for D_1 and that for D_2 are the same if there exists $\pi \in$* Aut(C) *such that $\pi D_1 = D_2$.*

Proof. For any codewords $\boldsymbol{c} \in D_1$, from Theorem 14, $\pi\boldsymbol{c} \in D_2$ is a minimal codeword if and only if $\boldsymbol{c}$ is a minimal codeword. Therefore, the local weight subdistribution for D_1 and that for D_2 are the same. □

This theorem is a condition for cosets having the same local weight subdistribution. The following lemma gives the set of all permutations by which every coset in C/C' is permuted into one in C/C'.

Lemma 18. *For a linear code C and its linear subcode C',*

$$\{\pi : \pi D \in C/C' \textit{ for all } D \in C/C'\} = \mathrm{Aut}(C) \cap \mathrm{Aut}(C').$$

Proof. Let $\pi \in \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$. For a coset $\boldsymbol{c}_1 + C' \in C/C'$, suppose that $\pi\boldsymbol{c}_1 \in \boldsymbol{c}_2 + C'$. For any codeword $\boldsymbol{c}_1 + \boldsymbol{u}_1 \in \boldsymbol{c}_1 + C'$,

$$\begin{aligned} \pi(\boldsymbol{c}_1 + \boldsymbol{u}_1) &= \pi\boldsymbol{c}_1 + \pi\boldsymbol{u}_1 \\ &= \boldsymbol{c}_2 + \boldsymbol{u}_2 + \pi\boldsymbol{u}_1, \quad \boldsymbol{u}_2 \in C', \\ &= \boldsymbol{c}_2 + (\boldsymbol{u}_2 + \pi\boldsymbol{u}_1) \ \in \ \boldsymbol{c}_2 + C'. \end{aligned}$$

Thus $\pi\boldsymbol{c}_1 + C' = \boldsymbol{c}_2 + C' \in C/C'$. Then $\{\pi : \pi D \in C/C' \text{ for all } D \in C/C'\} \supseteq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$.

Let $\pi \in \{\rho : \rho D \in C/C' \text{ for all } D \in C/C'\}$. For any codeword $\boldsymbol{c} \in C$, $\boldsymbol{c}$ must be in either coset in C/C', and then $\pi\boldsymbol{c} \in C$. Thus, $\pi \in \mathrm{Aut}(C)$. C' itself is one of the cosets in C/C'. For any codeword $\boldsymbol{u} \in C'$, $\pi\boldsymbol{u} \in C'$ because $\pi C' = C'$. Thus $\pi \in \mathrm{Aut}(C')$. Then $\{\pi : \pi D \in C/C' \text{ for any } D \in C/C'\} \subseteq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$. □

$\mathrm{Aut}(C) \cap \mathrm{Aut}(C')$ (or even $\mathrm{Aut}(C)$) is generally not known. Only subgroups of $\mathrm{Aut}(C) \cap \mathrm{Aut}(C')$ are known. Therefore, we use a subgroup.

Definition 6. *Let $\Pi \subseteq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$. For $D_1, D_2 \in C/C'$, we denote $D_1 \sim_\Pi D_2$ if and only if there exists $\pi \in \Pi$ such that $\pi D_1 = D_2$.*

Proposition 6. *The relation "$\sim_\Pi$" is an equivalence relation on C/C' if Π forms a group.*

Proof. Let $D_1, D_2, D_3 \in C/C'$.
(Reflexivity: $D_1 \sim_\Pi D_1$) Since the identity permutation π_0 is in Π, $D_1 \sim_\Pi D_1$.
(Symmetry: $D_1 \sim_\Pi D_2 \to D_2 \sim_\Pi D_1$) Suppose that $D_1 \sim_\Pi D_2$ and $\pi D_1 = D_2$ for $\pi \in \Pi$. Since Π forms a group, there exists $\rho \in \Pi$ such that $\rho\pi D_1 = D_1$. Then $\rho D_2 = D_1$, and $D_2 \sim_\Pi D_1$.
(Transitivity: $D_1 \sim_\Pi D_2, D_2 \sim_\Pi D_3 \to D_1 \sim_\Pi D_3$) Suppose that $D_1 \sim_\Pi D_2$ and $D_2 \sim_\Pi D_3$. There exists $\pi, \rho \in \Pi$ such that $\pi D_1 = D_2$, $\rho D_2 = D_3$. Then $D_3 = \rho D_2 = \rho\pi D_1$. Since $\rho\pi \in \Pi$, $D_1 \sim_\Pi D_3$. □

When the set of cosets are partitioned into the equivalence classes by the relation "$\sim_\Pi$", the local weight subdistributions for cosets which belong to the same equivalence class are the same.

We give a useful theorem for partitioning the set of cosets into equivalence classes by the relation "$\sim_\Pi$."

Theorem 16. *Let $\Pi \subseteq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$. For $D_1, D_2 \in C/C'$ and $\pi \in \Pi$, we have $D_1 \sim_\Pi D_2$ if $\pi\boldsymbol{c}_1 \in D_2$ for any $\boldsymbol{c}_1 \in D_1$.*

Proof. Let $\pi\boldsymbol{c}_1 = \boldsymbol{c}_2 \in D_2$. Any codeword in D_1 is represented by $\boldsymbol{c}_1 + \boldsymbol{c}$ ($\boldsymbol{c} \in C'$). Then

$$\begin{aligned}\pi(\boldsymbol{c}_1 + \boldsymbol{c}) &= \pi\boldsymbol{c}_1 + \pi\boldsymbol{c} \\ &= \boldsymbol{c}_2 + \pi\boldsymbol{c}.\end{aligned}$$

Since $\pi \in \mathrm{Aut}(C')$, $\pi\boldsymbol{c}$ is in C'. Thus $\pi D_1 = D_2$. □

Theorem 16 implies that, to partition the set of cosets into equivalence classes, we only need to check whether the representative codeword of a coset is permuted into another coset. After partitioning cosets into equivalence classes, the local weight subdistribution for only one coset in each equivalence class needs to be computed. Thereby the computational complexity is reduced. That is,

$$|L_i(C)| = \sum_{D \in RC_\Pi(C/C')} e_\Pi(D) \cdot |LS_i(D)|, \tag{6.2}$$

where $RC_\Pi(C/C')$ is the set of representative cosets obtained by partitioning C/C' into the equivalence classes with the set of permutations $\Pi \subseteq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$, and $e_\Pi(D)$ is the number of equivalent cosets to D when using Π for partitioning cosets.

6.4 An Algorithm for Computing LWDs

On the basis of the method of partitioning the set of cosets described in the previous section, we can compute the local weight distribution as follows (refer to Figure 6.1):

1. Choose a subcode C' and $\Pi \subseteq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$.
2. Partition C/C' into the equivalence classes with permutations in Π, and obtain $RC_\Pi(C/C')$ and $e_\Pi(D)$ for every $D \in RC_\Pi(C/C')$.
3. Compute the local weight subdistributions for every $D \in RC_\Pi(C/C')$.
4. Compute the local weight distribution of C from (6.2).

Figure 6.2 shows the details of the algorithm. Step 2 can be skipped if $RC_\Pi(C/C')$ and $e_\Pi(D)$ for $D \in RC_\Pi(C/C')$ are already known.

6.4.1 Coset Partitioning

Our implementation of Step 2 of the algorithm is based on Theorem 16 and Proposition 6. Let H' be a parity check matrix of C' with

$$H' = \begin{pmatrix} H_0 \\ H \end{pmatrix},$$

where H is a parity check matrix of C and H_0 is an $n \times (k - k')$ matrix. To partition cosets efficiently, we use the following condition

$$\pi\boldsymbol{c} + C' = \boldsymbol{c}' + C' \quad \text{if and only if} \quad (\pi\boldsymbol{c})H_0^T = \boldsymbol{c}'H_0^T,$$

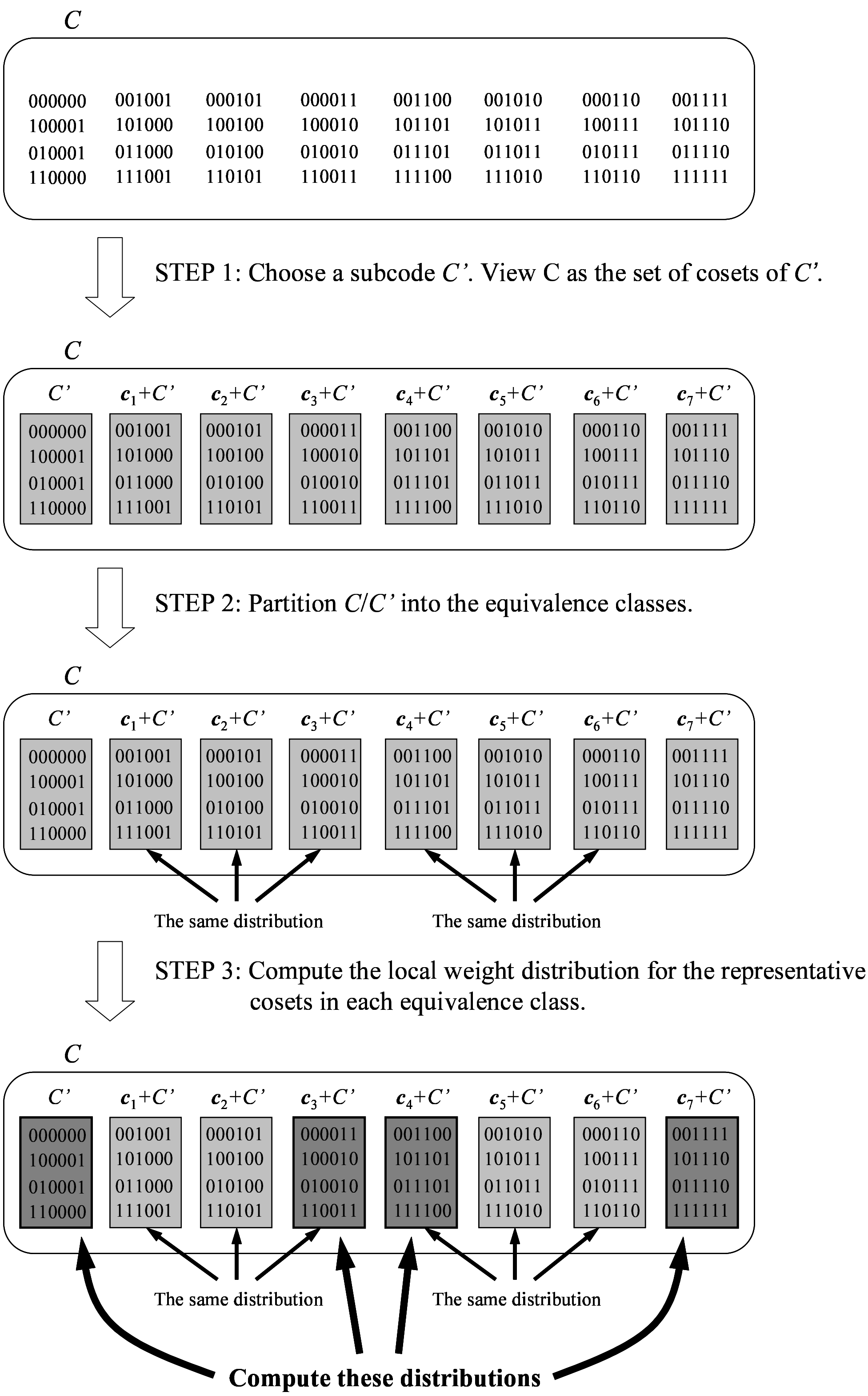


Figure 6.1: Diagram of the procedure of the proposed algorithm.

Input: G : a generator matrix of an (n, k, d) linear code C.
G' : a generator matrix of a linear subcode C'.
$RC_\Pi(C/C')$: the set of representative cosets obtained by partitioning C/C' into the equivalence classes with Π.
$e_\Pi(D)$ for $D \in RC_\Pi(C/C')$: the number of equivalent cosets to D when using Π for partitioning cosets.

Output: $L[i] (0 \leq i \leq n)$: the local weight distribution of C.

Algorithm:

For $i \leftarrow 0$ to n:
  $L[i] \leftarrow 0$.
Generate the cosets C/C'.
Partition the cosets into the equivalent classes.
For every representative coset $D \in RC_\Pi(C/C')$:
  $num \leftarrow e_\Pi(D)$.
  For every codeword $\boldsymbol{u}$ in D:
    $w \leftarrow$ the Hamming weight of $\boldsymbol{u}$.
    If $w < 2d$:
      $L[w] \leftarrow L[w] + num$.
    If $\boldsymbol{u}$ is turned out to be a minimal codeword:
      $L[w] \leftarrow L[w] + num$.

Figure 6.2: An algorithm for computing the local weight distribution.

where H_0^T represents the transpose of H_0.

Using a table with size $2^{k-k'}$, we need to compute the syndromes of length $k-k'$ for all the permuted coset leaders to partition these cosets into the equivalence classes. The computational complexity of partitioning cosets into the equivalence classes is $O(n(k-k')2^{k-k'}|\Pi|)$. If Π forms a group, the actual complexity would be much small. Suppose that $\pi\boldsymbol{c}+C' = \boldsymbol{c}'+C'$. After we found the equivalent cosets of $\boldsymbol{c}+C'$, including $\boldsymbol{c}'+C'$, we need not to find the equivalent cosets for $\boldsymbol{c}'+C'$ because the equivalent cosets of $\boldsymbol{c}'+C'$ are equal to that of $\boldsymbol{c}+C'$ when Π forms a group. Then the complexity of partitioning cosets into the equivalence classes is $O(n(k-k')e|\Pi| + 2^{k-k'})$ where e is the number of equivalence classes in C/C'. The complexity $O(2^{k-k'})$ is for computing syndromes and the bookkeeping operations for the $2^{k-k'}$ coset leaders. Since e seems to be much smaller than $2^{k-k'}$, although we cannot know e before running a coset partitioning algorithm, the actual complexity of partitioning cosets into into equivalence classes would be much small when Π forms a group.

6.4.2 Checking Minimality

We show two algorithms for checking minimality of codewords. For $\boldsymbol{v} \in C'$, let

$$C_{\mathrm{cov}}(\boldsymbol{c}) = \{\boldsymbol{c}' \in C : \boldsymbol{c}' \subseteq \boldsymbol{c}\}.$$

Then $C_{\mathrm{cov}}(\boldsymbol{c})$ is a linear subcode of C for $\boldsymbol{c} \in C$. Note that a codeword $\boldsymbol{c}$ is a minimal codeword if and only if $C(\boldsymbol{c}) = \{\boldsymbol{0}, \boldsymbol{c}\}$. Hence checking the minimality of $\boldsymbol{c}$ is whether the dimension of $C(\boldsymbol{v})$, denoted by $\dim(C(\boldsymbol{c}))$, is one or not. To obtain the dimension of $C_{\mathrm{cov}}(\boldsymbol{c})$ we can consider two methods: methods G and H.

G Method

G Method uses a generator matrix of C, denoted by G. The G Method derives a generator matrix of $C_{\mathrm{cov}}(\boldsymbol{c})$ from G and checks the dimension of $C_{\mathrm{cov}}(\boldsymbol{c})$. The algorithm of the G Method is shown in Figure 6.4. This algorithm is equivalent to the following G Rule presented in [2].

G Rule : Let $G_0(C, \boldsymbol{c})$ denote the matrix formed by the columns of a generator matrix G of C corresponding to positions where a given codeword $\boldsymbol{c} \in C$ has zeros. Then $\boldsymbol{c}$ is a minimal codeword if and only if the rank of $G_0(C, \boldsymbol{c})$ is $k-1$ where k is the dimension of C.

Input: G : a generator matrix of C, H : a parity check matrix of C.
G' : a generator matrix of C', H' : a parity check matrix of C'
Π : a subgroup of $\mathrm{Aut}(C) \cap \mathrm{Aut}(C')$.

Output: class$[i]$ $(1 \leq i \leq 2^{k-k'})$: $e_\Pi(D)$ for $D \in RC_\Pi(C/C')$.

Algorithm:

Generate the coset leaders, $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{2^{k-k'}}$ from G and G'.
Generate H_0 from H and H' to calculate the syndrome of codewords.
For $i \leftarrow 1$ to $2^{k-k'}$:
    class$[i] \leftarrow 0$.
    synd2index$[\boldsymbol{v}_i H_0^T] \leftarrow i$.
For $i \leftarrow 1$ to $2^{k-k'}$:
    If class$[i] = 0$:
        class$[i] \leftarrow 1$.
        For every $\pi \in \Pi$:
            If class[synd2index$[\boldsymbol{\pi v_i} H_0^T]$] $= 0$:
                class[synd2index$[\pi \boldsymbol{v}_i H_0^T]$] $= -1$.
                class$[i] \leftarrow$ class$[i] + 1$.

Figure 6.3: An algorithm for coset partitioning.

Input: $\boldsymbol{c} \in C$: a codeword to be checked.
$\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k$: the rows of a generator matrix of C.

Output: 1 : if $\boldsymbol{c}$ is a minimal codeword,
0 : otherwise.

Algorithm:

```
i ← 0.
For every p in {1, 2, ..., n} \ S(c):
    first ← 1.
    For j ← i + 1 to n:
        If the p-th element in g_j is 0:
            If first = 1 then:
                pivot ← j.
                first ← 0.
            else:
                g_j ← g_j + g_pivot.
    If first = 0 then:
        Swap(g_i, g_pivot).
        i ← i + 1.
        If i = k − 1 then:
            return 1.
return 0.
```

Figure 6.4: G Method : An algorithm for checking minimality using a generator matrix G.

H Method

H Method uses a parity check matrix of C, denoted by H. The H Method derives a parity check matrix of $C_{\rm cov}(\boldsymbol{c})$ from H and checks the dimension of $C_{\rm cov}(\boldsymbol{c})$. The algorithm of the H Method is shown in Figure 6.5. This algorithm is equivalent to the following H Rule presented in [2].

H Rule : Let $H_1(C, \boldsymbol{c})$ denote the matrix formed by the columns of a parity check matrix H of C corresponding to positions where a given codeword $\boldsymbol{c} \in C$ has ones. Then $\boldsymbol{c}$ is a minimal codeword if and only if the rank of $H_1(C, \boldsymbol{c})$ is $w(\boldsymbol{c}) - 1$.

6.4.3 Complexity

Here, we analyze the computational complexity of the algorithm. Let C be an (n, k) linear code and C' be an (n, k') linear subcode of C.

Time complexity

We can use G Method and H Method to check whether a given codeword is a minimal codeword. The time complexity of checking one codeword is $O(n^2k)$ and $O(n^2(n-k))$ for G Method and H Method, respectively. Thus, we use G Method for codes with rate less than 1/2. Since the number of codewords in each coset is $2^{k'}$, the total number of codewords to be checked by the procedure is $|RC_\Pi(C/C')|2^{k'}$. Hence, the time complexity of Step (3) of the proposed algorithm using G Method is $O(n^3 2^{k'}|RC_\Pi(C/C')|)$. The time complexity of Step (2), partitioning into the equivalence classes, is $O(n(k-k')2^{k-k'}|\Pi|)$.

Therefore, The time complexity of the entire algorithm is $O(n^3 2^{k'}|RC_\Pi(C/C')| + n(k-k')2^{k-k'}|\Pi|)$. When k' is chosen as $k' > k/2$, then $2^{k'} > 2^{k-k'}$, and the complexity of partitioning into the equivalence classes is much smaller than of computing the local weight subdistributions for cosets.

Space complexity

The space complexity of checking minimality is very small, because we need space to store only a generator matrix or a parity check matrix of C, which is $O(n^2)$. On the other hand, the space complexity of partitioning cosets into equivalence classes is much larger. We need space to store the entries proportional to $2^{k-k'}$, which is $O((k-k')2^{k-k'})$. We need $O(n(n-k'))$ space to store the parity check matrices of C and C'. The space complexity of the entire algorithm is $O(n^2 + (k-k')2^{k-k'})$.

Input: $\boldsymbol{c} \in C$: a codeword to be checked.
$\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_{n-k}$: the rows of a parity check matrix of H.

Output: 1 : if $\boldsymbol{c}$ is a minimal codeword,
0 : otherwise.

Algorithm:

```
i ← 0.
For every p in S(c):
    first ← 1.
    For j ← i + 1 to n − k:
        If the p-th element in h_j is 0:
            If first = 1 then:
                pivot ← j.
                first ← 0.
            else:
                h_j ← h_j + h_pivot.
    If first = 0 then:
        Swap(h_i, h_pivot).
        i ← i + 1.
If i = w − 1 then:
    return 1.
else:
    return 0.
```

Figure 6.5: H Method : An algorithm for checking minimality using a parity check matrix H.

6.4.4 Selection of a Subcode

To reduce the number of codewords that are checked minimality, we need to choose the subcode C' properly for which the number of permutations in $\Pi \subseteq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$ is larger.

If there are several subcodes with the same Π, then the subcode with the smaller dimension should be chosen to minimize the number of codewords that need to be checked, as long as the complexity of partitioning cosets into equivalence classes is relatively small.

6.5 Improvements of the Algorithm

In this section, some improvements of the proposed algorithm for computing the local weight distribution are shown.

6.5.1 Code Tree Structure

We consider reducing the complexity of checking minimality in a coset of C' by using the code tree structure of the coset. For simplicity, we consider C' itself as the coset. Recall that checking the minimality of $\boldsymbol{c}$ is whether $\dim(C_{\mathrm{cov}}(\boldsymbol{c}))$ is one or not. For $\boldsymbol{c} \in C'$ and i with $1 \leq i \leq n$, let

$$C_{\mathrm{cov}}(\boldsymbol{c}, i) = \{\boldsymbol{c}' \in C : S(\boldsymbol{c}') \cap \{1, \ldots, i\} \subseteq S(\boldsymbol{c}) \cap \{1, \ldots, i\}\}.$$

Therefore, $C_{\mathrm{cov}}(\boldsymbol{c}, n) = C_{\mathrm{cov}}(\boldsymbol{c})$. An implementation to construct $C_{\mathrm{cov}}(\boldsymbol{c})$ is as follows: Construct $C_{\mathrm{cov}}(\boldsymbol{c}, 1)$ from C, and $C_{\mathrm{cov}}(\boldsymbol{c}, 2)$ from $C_{\mathrm{cov}}(\boldsymbol{c}, 1)$, and $C(\boldsymbol{c}, 3)$ from $C_{\mathrm{cov}}(\boldsymbol{c}, 2)$, and so on. This procedure can be done by using the generator matrix of C.

A code tree of a binary (n, k) code is an edge-labeled tree with depth n. Either 0 or 1 is labeled on each edge. For the code tree of a code C, the sequence of edge labels along each path from the root to a leaf is a codeword of C. There are 2^k leaves on the tree. For example, the code tree of $C = \{0000, 0011, 1001, 1010\}$ is shown in Figure 6.6.

Now, we consider reducing the complexity of computing $C_{\mathrm{cov}}(\boldsymbol{c})$ for $\boldsymbol{c} \in C'$. For i with $1 \leq i \leq n$, let

$$C'^{\mathrm{f}}_i = \{(c'_1, c'_2, \ldots, c'_n) \in C' : c'_j = 0 \text{ for } 1 \leq j \leq i\}.$$

C'^{f}_i is the future subcode of C' at time i. For $\boldsymbol{c} \in C'$, $\boldsymbol{c} + C'^{\mathrm{f}}_i$ shares the same path to depth i in the code tree. This means, if we construct $C_{\mathrm{cov}}(\boldsymbol{c}, i)$ once, we do not need to construct $C_{\mathrm{cov}}(\boldsymbol{c}', i)$ for other $\boldsymbol{c}' \in \boldsymbol{c} + C'^{\mathrm{f}}_i$ later because $C_{\mathrm{cov}}(\boldsymbol{c}, i) = C_{\mathrm{cov}}(\boldsymbol{c}', i)$. We can

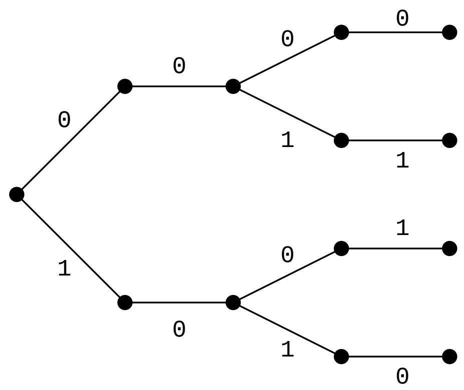


Figure 6.6: The code tree of the code $\{0000, 0011, 1001, 1010\}$.

save the computational complexity of constructing $C_{\text{cov}}(\boldsymbol{c}', i)$ from C for each $\boldsymbol{c}' \in \boldsymbol{c} + C'^{\text{f}}_i$. However, computing $C_{\text{cov}}(\boldsymbol{c}', i)$ for all $\boldsymbol{c}' \in C'$ along with the code tree is space-consuming. Therefore, we take the following method of checking minimality of them.

- Choose an integer i with $1 \leq i \leq n$.
- For each coset $\boldsymbol{c} + C'^{\text{f}}_i \in C'/C'^{\text{f}}_i$, construct $C_{\text{cov}}(\boldsymbol{c}, i)$ from C.
  - For each $\boldsymbol{c}' \in \boldsymbol{c} + C'^{\text{f}}_i$, construct $C_{\text{cov}}(\boldsymbol{c}')$ from $C_{\text{cov}}(\boldsymbol{c}, i)$ and investigate $\dim(C_{\text{cov}}(\boldsymbol{c}'))$.

We can construct the generator matrix of C'^{f}_i by row operations of the generator matrix of C' (see Figure 6.7). In Figure 6.7, the dimension of C'^{f}_i is k'^{f}_i. We should choose i properly in order to make C'^{f}_i large and the complexity of examining the dimension of $C_{\text{cov}}(\boldsymbol{c}')$ from $C_{\text{cov}}(\boldsymbol{c}, i)$ for each $\boldsymbol{c}' \in C'^{\text{f}}_i$ small; that is, make k'^{f}_i large and i large. The $k'^{\text{f}}_i \cdot i$ zero matrix in Figure 6.7 varies depending on the code tree structure of C'. For extended binary primitive BCH codes, permuting the symbol positions of codewords properly makes the $k'^{\text{f}}_i \cdot i$ matrix larger [24]. To choose i properly, we should estimate the effect by using the above technique.

Estimating precisely how the computational complexity is reduced is not easy. We will estimate the effect roughly. When $\dim(C_{\text{cov}}(\boldsymbol{c}')) = 1$, $\dim(C_{\text{cov}}(\boldsymbol{c}'))$ is found to be one before constructing $C_{\text{cov}}(\boldsymbol{c}')$, since $C_{\text{cov}}(\boldsymbol{c}', j)$ for $i \leq j \leq n$ may be equal to $C_{\text{cov}}(\boldsymbol{c}')$ for certain i with $i < n$. Let i_{end} be the average position i at which $\dim(C_{\text{cov}}(\boldsymbol{c}, i))$ is found to be one or not for $\boldsymbol{c} \in C$. We observe that the number of minimal codewords is much more than that of non-minimal codewords. For example, the rate of the number of

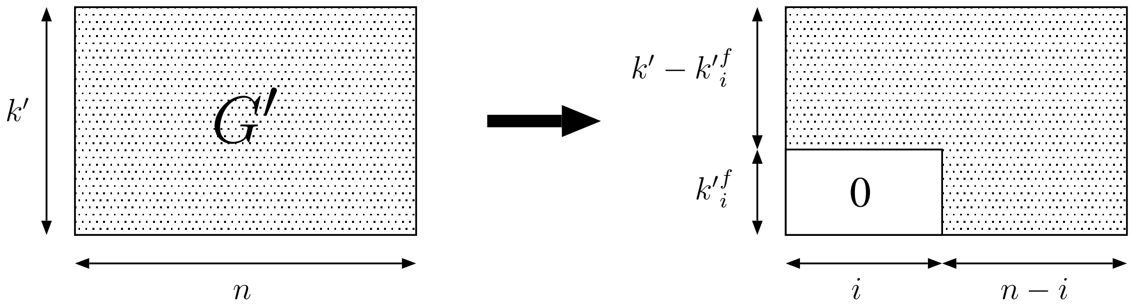


Figure 6.7: A way of constructing C'^{f}_{i} from the generator matrix G'.

minimal codewords to the number of all codewords is $0.9994\cdots$ for the $(128, 43)$ primitive BCH code. For any $\boldsymbol{c} \in C$ and $1 \leq i \leq i_{\mathrm{end}}$, assume:

$$\dim(C_{\mathrm{cov}}(\boldsymbol{c}, i)) = \frac{i_{\mathrm{end}} - i}{i_{\mathrm{end}}}(k-1) + 1.$$

This equation means that $\dim(C_{\mathrm{cov}}(\boldsymbol{c}, i))$ decreases linearly with i and is equal to k (or 1) when $i = 0$ (or $i = i_{\mathrm{end}}$). The complexity of computing $C_{\mathrm{cov}}(\boldsymbol{c}, i+1)$ from $C_{\mathrm{cov}}(\boldsymbol{c}, i)$ is proportional to $\dim(C_{\mathrm{cov}}(\boldsymbol{c}, i))$. Thus, the complexity is given as $a \cdot \dim(C_{\mathrm{cov}}(\boldsymbol{c}, i))$ where a is a nonzero constant.

Consider the case i_0 is chosen as i for using the technique described in this section. Let U_1 be the complexity of computing $C_{\mathrm{cov}}(\boldsymbol{c})$, which is equal to the complexity of checking minimality without the technique, U_2 be the complexity of computing $C_{\mathrm{cov}}(\boldsymbol{c}, i_{end})$ from $C_{\mathrm{cov}}(\boldsymbol{c}, i_0)$, and U_3 be the average complexity of computing $C_{\mathrm{cov}}(\boldsymbol{c}, i_0)$. Then

$$\begin{aligned} U_1 &= \frac{a(\dim(C) - 1)\, i_{\mathrm{end}}}{2} = \frac{a(k-1) i_{\mathrm{end}}}{2}, \\ U_2 &= \frac{a(\dim(C, i_0) - 1)(i_{\mathrm{end}} - i_0)}{2} \\ &= \frac{a(i_{\mathrm{end}} - i_0)^2 (k-1)}{2\, i_{\mathrm{end}}}, \\ U_3 &= U_1 - U_2. \end{aligned}$$

Let R_{i_0} be the relative complexity of checking minimality with the technique and without the technique. Then

$$R_{i_0} = \frac{U_3 + U_2 \cdot 2^{k'^{\mathrm{f}}_{i_0}}}{U_1 \cdot 2^{k'^{\mathrm{f}}_{i_0}}} = \left(1 - \frac{U_2}{U_1}\right) \frac{1}{2^{k'^{\mathrm{f}}_{i_0}}} + \frac{U_2}{U_1},$$

where

$$\frac{U_2}{U_1} = \left(\frac{i_{\mathrm{end}} - i_0}{i_{\mathrm{end}}}\right)^2.$$

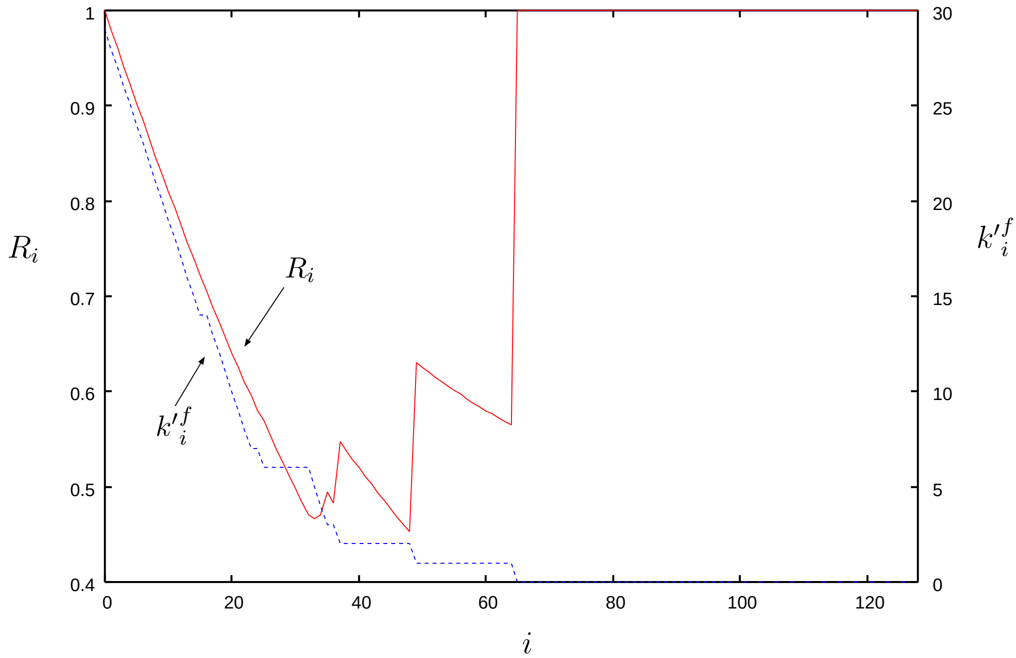


Figure 6.8: Relative complexity R_i with $i_{\text{end}} = 100$ and the dimension k'^f_i of C'^f_i for the $(128, 50)$ extended BCH code using the $(128, 29)$ code as a subcode.

We estimated R_{i_0} for the case of the $(128, 50)$ extended BCH code. In this case, the $(128, 29)$ code is chosen as the subcode C' and the number of representative cosets is 258. To determine i_{end}, we use $2^{15} \cdot 258$ codewords by choosing 2^{15} codewords randomly from each of the 258 representative cosets. For every codeword $\boldsymbol{c}$ in such codewords, we examined the position in which $\dim(C_{\text{cov}}(\boldsymbol{c}))$ is found to be one or not. Then the average was 100, that is, $i_{\text{end}} = 100$. Since $k'^{\text{f}}_{i_0}$ depends on i_0, we investigated $k'^{\text{f}}_{i_0}$ and computed R_{i_0} for every i_0 $(1 \leq i_0 \leq n)$ (see Figure 6.8). In this investigation, we use the permutation technique for making $k'^{\text{f}}_{i_0}$ and i_0 larger proposed in [18] for extended BCH codes. From Figure 6.8, the complexity of checking minimality would reduced by 1/2 for $i_0 = 33$ and 48. $k'^{\text{f}}_{i_0} = 5, 2$ for $i_0 = 33, 48$, respectively. Actually, for the $(128, 50)$ extended BCH code and the $(128, 29)$ extended BCH subcode, the complexity is reduced by about 1/2 when we choose $i_0 = 48$.

If the dimension of the subcode is small, k'^{f}_i may become small and the effect of using the code tree structure is small. We should choose the subcode by considering the effect of using the code tree structure.

6.5.2 Invariance Property in Cosets

In the proposed algorithm, the invariance property for minimality is applied to the set of cosets of a subcode rather than the set of codewords. This reduces the complexity of finding the representatives. However, we do not use the invariance property completely. That is, the invariance property is not used for codewords in cosets. In computing the local weight subdistribution for a coset, we can apply the invariance property to codewords in the coset. An invariance property in a coset is given in the following theorem.

Theorem 17. *For a coset* $\boldsymbol{c} + C' \in C/C'$, $\pi \in \{\rho : \rho\boldsymbol{c} \in \boldsymbol{c} + C'\}$, *and* $\boldsymbol{c}' \in \boldsymbol{c} + C'$, $\pi\boldsymbol{c}'$ *is a minimal codeword in* C *if and only if* $\boldsymbol{c}'$ *is a minimal codeword in* C.

No efficient way is known for generating the representative codewords in a coset as in a code. Therefore, we use a similar method; Just as we applied the invariance property to the set of cosets in a code rather than the set of codewords in the code, we apply the invariance property to the set of cosets in a coset rather than the set of codewords in the coset. Thus, we consider a coset $\boldsymbol{c} + C' \in C/C'$ the set of cosets of C'', where C'' is a linear subcode of C'.

For a coset $\boldsymbol{c}+C' \in C/C'$, let $(\boldsymbol{c}+C')/C''$ denote the set of all cosets of C'' in $\boldsymbol{c}+C'$, that is, $(\boldsymbol{c} + C')/C'' = \{\boldsymbol{c} + \boldsymbol{c}' + C'' : \boldsymbol{c}' \in C' \setminus C''\}$. Then

$$|(\boldsymbol{c} + C')/C''| = 2^{k'-k''} \quad \text{and} \quad \boldsymbol{c} + C' = \bigcup_{E \in (\boldsymbol{c}+C')/C''} E,$$

where k' and k'' are the dimensions of C' and C''. We also call the weight distribution of minimal codewords in $E \in (\boldsymbol{c} + C')/C''$ the local weight subdistribution for E. The following theorem gives an invariance property for cosets in $(\boldsymbol{c} + C')/C''$.

Theorem 18. *For* $E_1, E_2 \in (\boldsymbol{c} + C')/C''$, *the local weight subdistribution for* E_1 *and that for* E_2 *are the same if there exists* $\pi \in \{\rho : \rho\boldsymbol{c} \in \boldsymbol{c} + C', \rho \in \mathrm{Aut}(C) \cap \mathrm{Aut}(C')\}$ *such that* $\pi E_1 = E_2$.

We consider partitioning $(\boldsymbol{c} + C')/C''$ into equivalence classes. Permutations which are used to partition cosets into equivalence classes are presented in the following lemma.

Lemma 19. *For a coset* $\boldsymbol{c} + C' \in C/C'$,

$$\begin{aligned}\{\pi : \pi E \in (\boldsymbol{c} + C')/C'' \text{ for all } E \in (\boldsymbol{c} + C')/C''\}& \\ = \{\rho : \rho\boldsymbol{c} \in \boldsymbol{c} + C',\ \rho \in \mathrm{Aut}(C) \cap \mathrm{Aut}(C') \cap \mathrm{Aut}(C'')\}&.\end{aligned}$$

Proof. Let $\pi \in \{\rho : \rho\boldsymbol{c} \in \boldsymbol{c} + C',\ \rho \in \mathrm{Aut}(C) \cap \mathrm{Aut}(C') \cap \mathrm{Aut}(C'')\}$. For a coset $\boldsymbol{c} + \boldsymbol{c}_1 + C'' \in (\boldsymbol{c} + C')/C''$, suppose that $\pi\boldsymbol{c} = \boldsymbol{c} + \boldsymbol{c}_2, \boldsymbol{c}_2 \in C'$ and $\pi\boldsymbol{c}_1 = \boldsymbol{c}_3 \in C'$. For any codeword $\boldsymbol{c} + \boldsymbol{c}_1 + \boldsymbol{c}_1{}' \in \boldsymbol{c} + \boldsymbol{c}_1 + C''$, $\boldsymbol{c}_1{}' \in C''$,

$$\begin{aligned}
\pi(\boldsymbol{c} + \boldsymbol{c}_1 + \boldsymbol{c}_1{}') &= \pi\boldsymbol{c} + \pi\boldsymbol{c}_1 + \pi\boldsymbol{c}_1{}' \\
&= \boldsymbol{c} + \boldsymbol{c}_2 + \boldsymbol{c}_3 + \boldsymbol{c}_2{}', \qquad \pi\boldsymbol{c}_1{}' = \boldsymbol{c}_2{}' \in C'' \\
&= \boldsymbol{c} + (\boldsymbol{c}_2 + \boldsymbol{c}_3) + \boldsymbol{c}_2{}' \\
&\in\ \boldsymbol{c} + (\boldsymbol{c}_2 + \boldsymbol{c}_3) + C''.
\end{aligned}$$

Thus, $\pi(\boldsymbol{c} + \boldsymbol{c}_1 + C'') = \boldsymbol{c} + (\boldsymbol{c}_2 + \boldsymbol{c}_3) + C'' \in (\boldsymbol{c} + C')/C''$. Therefore, $\{\pi : \pi E \in (\boldsymbol{c}+C')/C'' \text{ for all } E \in (\boldsymbol{c}+C')/C''\} \supseteq \{\rho : \rho\boldsymbol{c} \in \boldsymbol{c}+C',\ \rho \in \mathrm{Aut}(C)\cap\mathrm{Aut}(C')\cap\mathrm{Aut}(C'')\}$.

Let $\pi \in \{\rho : \rho E \in (\boldsymbol{c} + C')/C'' \text{ for all } E \in (\boldsymbol{c} + C')/C''\}$. For any codeword $\boldsymbol{c} + \boldsymbol{c}_1 \in \boldsymbol{c} + C'$, $\boldsymbol{c} + \boldsymbol{c}_1$ must be in either coset in $(\boldsymbol{c} + C')/C''$, thus, $\pi(\boldsymbol{c} + \boldsymbol{c}_1) \in \boldsymbol{c} + C''$ and $\pi \in \mathrm{Aut}(C)$. For $\boldsymbol{c}+\boldsymbol{c}_1+C'' \in (\boldsymbol{c}+C')/C''$, let $\boldsymbol{c}+\boldsymbol{c}_1+\boldsymbol{c}_1{}', \boldsymbol{c}+\boldsymbol{c}_1+\boldsymbol{c}_2{}' \in \boldsymbol{c}+\boldsymbol{c}_1+C''$. $\pi(\boldsymbol{c}+\boldsymbol{c}_1+\boldsymbol{c}_1{}') = \pi\boldsymbol{c}+\pi\boldsymbol{c}_1+\pi\boldsymbol{c}_1{}'$ and $\pi(\boldsymbol{c}+\boldsymbol{c}_1+\boldsymbol{c}_2{}') = \pi\boldsymbol{c}+\pi\boldsymbol{c}_1+\pi\boldsymbol{c}_2{}'$ must be in the same coset of $\boldsymbol{c} + \boldsymbol{c}_2 + C''$. Hence, $\pi \in \mathrm{Aut}(C')$ and $\pi \in \mathrm{Aut}(C'')$. Therefore, $\{\pi : \pi E \in (\boldsymbol{c} + C')/C'' \text{ for all } E \in (\boldsymbol{c}+C')/C''\} \subseteq \{\rho : \rho\boldsymbol{c} \in \boldsymbol{c}+C',\ \rho \in \mathrm{Aut}(C)\cap\mathrm{Aut}(C')\cap\mathrm{Aut}(C'')\}$. ☐

To partition cosets into equivalence classes, we will use permutations presented in Lemma 19. Although $\mathrm{Aut}(C)$, $\mathrm{Aut}(C')$, and $\mathrm{Aut}(C'')$ are known, we need to obtain permutations π that satisfy $\pi\boldsymbol{c} \in \boldsymbol{c} + C'$. However, finding such permutations is difficult in general.

6.5.3 Computing the LWD of the $(256, 93)$ Reed-Muller Code

We will apply the invariance property to codewords in cosets for computing the local weight distribution of Reed-Muller codes, in particular the $(256, 93)$ third-order Reed-Muller code.

We need to find permutations in Lemma 19 to apply the invariance property. The permutations called *binary shifts* are such permutations for Reed-Muller codes.

Binary Shifts in Reed-Muller Codes

Let $\mathrm{RM}_{m,r}$ denote the r-th order Reed-Muller code of length 2^m. The $(256, 93)$ third-order Reed-Muller code is $\mathrm{RM}_{8,3}$. Since $\mathrm{RM}_{8,1} \subset \mathrm{RM}_{8,2} \subset \mathrm{RM}_{8,3}$, we choose $\mathrm{RM}_{8,2}$ as a subcode C' and $\mathrm{RM}_{8,1}$ as C''. The general affine group $GA(m)$ is an automorphism group of $\mathrm{RM}_{m,r}$. We choose $GA(8)$ as the permutation set Π. In [21, 38], the set of the representative cosets,

$RC_{GA(8)}(\mathrm{RM}_{8,3}/\mathrm{RM}_{8,2})$, and the numbers of the equivalent cosets, $e_{GA(8)}(D)$, for each $D \in RC_{GA(8)}(\mathrm{RM}_{8,3}/\mathrm{RM}_{8,2})$ are presented. The set of cosets $\mathrm{RM}_{8,3}/\mathrm{RM}_{8,2}$ is classified into 32 equivalence classes. We will compute the local weight subdistributions for the 32 representative cosets. To compute the local weight subdistributions for each representative coset $f + \mathrm{RM}_{8,2}$, we need to find a permutation set $\{\rho : \rho f \in f + \mathrm{RM}_{8,2},\ \rho \in GA(8)\}$ for each coset. Note that a polynomial f represents the corresponding codeword in Reed-Muller codes.

Recall that $GA(m)$ is the set of permutations that replace the codeword $f(x_1, \ldots, x_m)$ by $f\left(\sum a_{1j}x_j + b_j, ..., \sum a_{mj}x_j + b_m\right)$, where $A = (a_{ij})$ is an invertible $m \times m$ binary matrix and $(b_1, ..., b_m)$ is a binary m-tuple. An affine permutation is called a *binary shift* if A is the identity matrix E. Let $BS(m)$ denote $GA(m)$ with $A = E$.

The set of binary shifts is suitable for the permutation set described in Lemma 19 because, for any coset $f + \mathrm{RM}_{8,2}$, a binary shift π satisfies $\pi f \in f + \mathrm{RM}_{8,2}$ clearly. Let $C_{BS}(\boldsymbol{v})$ be a set of codewords permuted by the binary shifts, that is, $C_{BS}(\boldsymbol{v}) = \{\pi\boldsymbol{v} : \pi \in BS(m)\}$.

Lemma 20 ([15, 22])**.** *Let f be an m-variable Boolean polynomial of degree r. For a coset $f + \mathrm{RM}_{m,r-1}$, $C_{BS}(f)$ is a linear subspace of $f + \mathrm{RM}_{m,r-1}$.*

Lemma 21 ([15, 22])**.** *Let f be an m-variable Boolean polynomial of degree r, and $\beta_i \in BS(m)$ be the permutation that only replaces x_i by $x_i + 1$. For a coset $f + \mathrm{RM}_{m,r-1}$, $\beta_i f$ for $1 \le i \le m$ are bases of $C_{BS}(f)$.*

Lemma 22. *For $f + \mathrm{RM}_{m,r-1} \in \mathrm{RM}_{m,r}/\mathrm{RM}_{m,r-1}$, let $f + f_1 + \mathrm{RM}_{m,r-1}$ be a coset in $(f + \mathrm{RM}_{m,r-1})/\mathrm{RM}_{m,r-2}$. The local weight subdistribution of $f + f_1 + \mathrm{RM}_{m,r-1}$ and that of $f + f_1 + g + \mathrm{RM}_{m,r-1}$ for any $g \in C_{BS}(f_1)$ are the same.*

From Lemma 22, each coset in $(f + \mathrm{RM}_{m,r-1})/\mathrm{RM}_{m,r-2}$ has $|C_{BS}(f)| = 2^{\dim(C_{BS}(f))}$ equivalent cosets. Therefore, for each coset $f + \mathrm{RM}_{m,r-1} \in \mathrm{RM}_{m,r}/\mathrm{RM}_{m,r-1}$, the number of cosets in $(f + \mathrm{RM}_{m,r-1})/\mathrm{RM}_{m,r-2}$ we have to compute their local weight subdistributions will be reduced by $1/|C_{BS}(f)|$.

For the 32 representative cosets $f_i + \mathrm{RM}_{8,2} \in \mathrm{RM}_{8,3}/\mathrm{RM}_{8,2}$ for $1 \le i \le 32$, we computed the dimension of $C_{BS}(f_i)$. The computation is just investigating the number of independent vectors in candidate bases, which are presented in Lemma 21. The 32 representative cosets and the dimension of $C_{BS}(f_i)$ is listed in Table 6.1. In this table, we follow the notations in [22, 38]; The monomial $x_{i_1}x_{i_2}x_{i_3}$ is represented as $i_1i_2i_3$ for convenience. For most cases, the dimension of $C_{BS}(f_i)$ is 8 and thus the time complexity of computing the local weight subdistribution for $f_i + \mathrm{RM}_{8,2}$ is reduced by 1/256. For the case that $i = 1, 2, 3$ ($f_1 = 0$, $f_2 = x_1x_2x_3$, $f_3 = x_1x_2x_3 + x_2x_4x_5$), above binary shift set method is not very effective for their small $\dim(C_{BS}(f_i))$. For many of $f_i + \mathrm{RM}_{8,2}$ including those with $i \le 3$, we can find permutations such that $\pi f_i \in f_i + \mathrm{RM}_{8,2}$ because of their simple forms of polynomials.

Borissov and Manev [8] gave another approach for determining the local weight subdistributions for the four cosets $i = 1, 2, 3$ and 7. To describe their results, first we present the necessary and sufficient condition for minimality in Reed-Muller codes. Let P_m be the set of Boolean polynomials with m variables $x_1, x_2, \ldots, x_m$.

Lemma 23. *For $f, g \in P_m$, if $f \subseteq g$ then $gf = f$. Otherwise, $gf \subset f$.*

Theorem 19. *For a code C of length 2^m, $f \in C$ is not minimal in C if and only if there exists $g \in P_m$ such that $gf \in C \setminus \{0, f\}$.*

Proof. (If part) From Lemma 23, $gf \neq f$ means $gf \subset f$. The existence of $gf \in C$ such that $gf \subset f$ leads the non-minimality of f.
(Only if part) Non-minimality of f implies the existence of $f' \in C \setminus \{0\}$ such that $f' \subset f$. Then f' is g because $f'f = f' \neq f$ from Lemma 23. □

Corollary 6. *A Boolean polynomial $f \in \mathrm{RM}_{m,r}$ is minimal in $\mathrm{RM}_{m,r}$ if and only if, for any $g \in \mathrm{RM}_{m,r}$, $gf \notin \mathrm{RM}_{m,r} \setminus \{0, f\}$.*

Let's turn to determining the local weight distribution of $\mathrm{RM}_{8,3}$.

Theorem 20 ([8])**.** *For the coset $0 + \mathrm{RM}_{m,r-1} \in \mathrm{RM}_{m,r}/\mathrm{RM}_{m,r-1}$, any codeword in $0 + \mathrm{RM}_{m,r-1}$ is not minimal in $\mathrm{RM}_{m,r}$.*

Proof. For any codeword $f \in 0 + \mathrm{RM}_{m,r-1}$, we can pick a variable x_i $(1 \le i \le m)$ such that $x_i f \neq f$ because the order of f is $r - 1$ or less. From Theorem 19, f is not minimal in $\mathrm{RM}_{m,r}$. □

Theorem 21 ([8])**.** *For the coset $x_1x_2x_3 + \mathrm{RM}_{m,2} \in \mathrm{RM}_{m,3}/\mathrm{RM}_{m,2}$, $f \in x_1x_2x_3 + \mathrm{RM}_{m,2}$ is minimal in $\mathrm{RM}_{m,3}$ if and only if f is of the form $f = (x_1 + a_1)(x_2 + a_2)(x_3 + a_3)$, where $a_i \in \{0, 1\}$ for $1 \le i \le 3$.*

Proof. Let $f = x_1x_2x_3 + f'$ with $f \in \mathrm{RM}_{m,2}$.

In the case f' contains x_i $(4 \le i \le m)$, we can pick up a variable $x_j (1 \le j \le 3)$ such that $x_jf' \neq f'$. Then $x_jf = x_1x_2x_3 + x_jf' \neq f$ and $x_jf \neq 0$. $x_jf \in \mathrm{RM}_{m,r}$ because the order of x_jf is 3 or less. From Theorem 19, f is not minimal in $\mathrm{RM}_{m,3}$.

In the case f' contains only x_i $(1 \leq i \leq 3)$, if f is of the form $f = (x_1 + a_1)(x_2 + a_2)(x_3 + a_3)$, $a_i \in \{0, 1\}$, then $x_j f = f$ or $x_j f = 0$ for j $(1 \leq j \leq 3)$. Otherwise, we can pick j such that $x_j f \neq f$, $x_j f \neq 0$ and $x_j f \in \mathrm{RM}_{m,3}$. In this case, f is not minimal in $\mathrm{RM}_{m,3}$. □

Theorem 22 ([8]). *For the coset $x_1x_2x_3 + x_2x_4x_5 + \mathrm{RM}_{m,2} \in \mathrm{RM}_{m,3}/\mathrm{RM}_{m,2}$, suppose $f = x_1x_2x_3 + x_2x_4x_5 + f' \in x_1x_2x_3 + x_2x_4x_5 + \mathrm{RM}_{m,2}$. Then f is not minimal in $\mathrm{RM}_{m,3}$ except for the following two cases:*

1) *f is of the form $f = x_2((x_1x_3 + x_4x_5) + g)$, where g is a first-order Boolean polynomial.*

2) *f is of the form $f = (x_2 + 1)((x_1x_3 + x_4x_5) + g)$, where g is a first-order Boolean polynomial.*

Proof. $x_2 f = f$ for the case 1) and $x_2 f = 0$ for the case 2). Except for these two cases, $x_2 f \neq f$, $x_2 f \neq 0$, and $x_2 f \in \mathrm{RM}_{m,3}$. Thus f is not minimal in $\mathrm{RM}_{m,3}$ except for the two cases. □

A similar argument to Theorem 22 can be applied to the coset $x_1x_2x_7 + x_3x_4x_7 + x_5x_6x_7 + \mathrm{RM}_{m,2}$ and the same result holds for it.

From Theorem 20, there is no minimal codeword in $0 + \mathrm{RM}_{m,8,2}$. From Theorem 21, the local weight subdistribution for $x_1x_2x_3 + \mathrm{RM}_{m,8,2}$ is determined immediately because the codewords of the form $f = (x_1 + a_1)(x_2 + a_2)(x_3 + a_3)$ have the minimum weight and they are minimal codewords in $\mathrm{RM}_{m,8,3}$. For the coset $x_1x_2x_3 + x_2x_4x_5 + \mathrm{RM}_{m,2}$, codewords for which one should check minimality are restricted to two cases, which are described in Theorem 22. There are only 2^m patterns for a polynomial g in both cases. Checking minimality for these $2 \cdot 2^m$ codewords determines the local weight subdistribution for this coset.

6.6 Tables of LWDs

The local weight distributions determined by using the relations and the algorithms presented in Chapters 5 and 6 are listed. The local weight distributions of the $(128, k)$ extended primitive BCH codes for $k \leq 50$, those of the $(127, k)$ primitive BCH codes for $k = 36, 43, 50$, and those of the third-order Reed-Muller codes of length 128 and 256 are presented in Tables 6.2, 6.3, 6.4, respectively.

The local weight distributions of the $(128, k)$ extended primitive BCH codes for $k = 8, 15, 22, 29$ are immediately determined from their weight distribution because of Proposition 2. The distribution for $k = 36, 43, 50$ are determined by the algorithm proposed in this chapter. It took about 440 hours (CPU time) to compute the distribution of the $(128, 50)$ code with a 1.6 GHz Opteron processor. In this case, the $(128, 29)$ code is used as the subcode, and it took only one minute to partition cosets into the equivalence classes.

The local weight distributions of the $(127, k)$ primitive BCH codes are determined from those of the corresponding extended codes by Theorems 13. Note that $N_i(C)$ in Theorem 13 is equal to zero for all i in these cases.

The local weight distributions of the third-order Reed-Muller codes are determined by the algorithms presented in this chapter. For the $(128, 64)$ third-order Reed-Muller code, the $(128, 29)$ second-order Reed-Muller code is used as a sublinear code. The representative codewords of cosets for this case are presented in [21]. A method of obtaining the number of equivalent cosets are presented in [38]. Thus, the process of obtaining the representative cosets and the number of equivalent cosets are different from that for extended primitive BCH codes. Note that the computing time for this process is vanishingly small. The local weight distribution of the $(256, 93)$ third-order Reed-Muller code is determined by a method described in Section 6.5.3.

The local weight distributions of the punctured third-order Reed-Muller codes of length 127 and 255 are determined from those of the corresponding Reed-Muller codes by Theorems 13.

Although the tables are not listed, the local weight distributions of the even weight subcodes of the $(127, k)$ primitive BCH codes for $k \leq 50$ and the punctured Reed-Muller code of length $127, 255$ are determined from Theorem 12.

6.7 Concluding Remarks

The local weight distributions of some primitive BCH codes, extended primitive BCH codes, Reed-Muller codes, punctured Reed-Muller codes, and even weight subcodes of primitive BCH codes and punctured Reed-Muller codes are determined by using the algorithms presented in this chapter and the relations presented in Chapter 5. It is known that the local weight distribution gives tighter upper bound than the usual union bounds using the weight distribution on the error probability after ML decoding over AWGNC.

Yasuda et al. [41, 42] studied on what bounds the local weight distribution can be substituted for the weight distribution and how good the bounds given by the local weight distribution are. They showed that the Séguin lower bound [36] and the Poltyrev upper bound [32] can be improved by using the local weight distributions instead for the weight distributions. Experimental results for some primitive BCH codes, Hamming codes, Golay codes, and Reed-Muller codes showed that the Séguin bound is somewhat improved, in particular for high-rate codes, but the Poltyrev bound is little improved. Since the difference between the local weight distribution and the weight distribution is large if the code rate is high as seen in the results for random linear codes, the improvements of the bounds will be large for high rate. However, for a fixed code length, the complexity of computing the local weight distribution is larger for codes of higher rate. Besides, to the local weight distribution, we cannot simply apply the MacWilliams identity [25], which is used for determining the weight distribution for high-rate codes from those of the corresponding low-rate dual codes. Therefore, it is desirable to develop a method of computing the local weight distribution of high-rate codes. For cyclic codes, an algorithm for computing the local weight distribution effective for high-rate one was proposed in [29]. This algorithm is based on the algorithm proposed in [5]. Since the size of cyclic permutations is not large, an algorithm for computing the local weight distribution is desirable for high-rate codes that are closed under large automorphism groups.

Table 6.1: The dimension of $C_{BS}(f_i)$ for representative coset $f_i + \mathrm{RM}_{8,2} \in \mathrm{RM}_{8,3}/\mathrm{RM}_{8,2}$

i	f_i	$\dim(C_{BS}(f_i))$
1	0	0
2	123	3
3	123+245	5
4	123+456	6
5	123+245+346	6
6	123+145+246+356+456	6
7	127+347+567	7
8	123+456+147	7
9	123+245+346+147	7
10	123+456+147+257	7
11	123+145+246+356+456+167	7
12	123+145+246+356+456+167+247	7
13	123+456+178	8
14	123+456+178+478	8
15	123+245+678+147	8
16	123+245+346+378	8
17	123+145+246+356+456+178	8
18	123+145+246+356+456+167+238	8
19	123+145+246+356+456+158+237+678	8
20	123+145+246+356+456+278+347+168	8
21	145+246+356+456+278+347+168+237+147	8
22	123+234+345+456+567+678+128+238+348+458+568+178	8
23	123+145+246+356+456+167+578	8
24	123+145+246+356+456+167+568	8
25	123+145+246+356+456+167+348	8
26	123+456+147+257+268+278+348	8
27	123+456+147+257+168+178+248+358	8
28	127+347+567+258+368	8
29	123+456+147+368	8
30	123+456+147+368+578	8
31	123+456+147+368+478+568	8
32	123+456+147+168+258+348	8

Table 6.2: The local weight distributions of the $(128, k)$ extended primitive BCH codes.

$k = 8$

i	$\lvert L_i \rvert$
64	254

$k = 15$

i	$\lvert L_i \rvert$
56	8 192
64	16 638
72	8 192

$k = 22$

i	$\lvert L_i \rvert$
48	42 672
56	877 824
64	2 353 310
72	877 824
80	42 672

$k = 29$

i	$\lvert L_i \rvert$
44	373 888
48	2 546 096
52	16 044 672
56	56 408 320
60	116 750 592
64	152 623 774
68	116 750 592
72	56 408 320
76	16 044 672
80	2 546 096
84	373 888

$k = 36$

i	$\lvert L_i \rvert$
32	10 668
36	16 256
40	2 048 256
44	35 551 872
48	353 494 848
52	2 028 114 816
56	7 216 135 936
60	14 981 968 512
64	19 484 132 736
68	14 981 968 512
72	7 216 127 808
76	2 028 114 816
80	348 203 520
84	35 551 872
88	2 048 256

$k = 43$

i	$\lvert L_i \rvert$
32	124 460
36	8 810 752
40	263 542 272
44	4 521 151 232
48	44 899 876 672
52	262 118 734 080
56	915 924 097 536
60	1 931 974 003 456
64	2 476 669 858 944
68	1 931 944 645 120
72	915 728 180 224
76	261 375 217 152
80	43 168 588 288
84	2 464 897 280

$k = 50$

i	$\lvert L_i \rvert$
28	186 944
32	19 412 204
36	113 839 296
40	33 723 852 288
44	579 267 441 920
48	5 744 521 082 944
52	33 558 415 333 632
56	117 224 645 074 752
60	247 311 270 037 888
64	316 973 812 770 944
68	247 074 613 401 728
72	115 408 474 548 096
76	25 844 517 328 896

Table 6.3: The local weight distributions of the $(127, k)$ primitive BCH codes.

$k = 36$		$k = 43$		$k = 50$	
i	$\lvert L_i \rvert$	i	$\lvert L_i \rvert$	i	$\lvert L_i \rvert$
31	2 667	31	31 115	27	40 894
32	8 001	32	93 345	28	146 050
35	4 572	35	2 478 024	31	4 853 051
36	11 684	36	6 332 728	32	14 559 153
39	640 080	39	82 356 960	35	310 454 802
40	1 408 176	40	181 185 312	36	793 384 494
43	12 220 956	43	1 554 145 736	39	10 538 703 840
44	23 330 916	44	2 967 005 496	40	23 185 148 448
47	132 560 568	47	16 837 453 752	43	199 123 183 160
48	220 934 280	48	28 062 422 920	44	380 144 258 760
51	823 921 644	51	106 485 735 720	47	2 154 195 406 104
52	1 204 193 172	52	155 632 998 360	48	3 590 325 676 840
55	3 157 059 472	55	400 716 792 672	51	13 633 106 229 288
56	4 059 076 464	56	515 207 304 864	52	19 925 309 104 344
59	7 022 797 740	59	905 612 814 120	55	51 285 782 220 204
60	7 959 170 772	60	1 026 361 189 336	56	65 938 862 854 548
63	9 742 066 368	63	1 238 334 929 472	59	115 927 157 830 260
64	9 742 066 368	64	1 238 334 929 472	60	131 384 112 207 628
67	7 959 170 772	67	1 026 345 592 720	63	158 486 906 385 472
68	7 022 797 740	68	905 599 052 400	64	158 486 906 385 472
71	4 059 071 892	71	515 097 101 376	67	131 258 388 369 668
72	3 157 055 916	72	400 631 078 848	68	115 816 225 032 060
75	1 204 193 172	75	155 191 535 184	71	64 917 266 933 304
76	823 921 644	76	106 183 681 968	72	50 491 207 614 792
79	217 627 200	79	26 980 367 680	75	15 345 182 164 032
80	130 576 320	80	16 188 220 608	76	10 499 335 164 864
83	23 330 916	83	1 617 588 840		
84	12 220 956	84	847 308 440		
87	1 408 176				
88	640 080				

Table 6.4: The local weight distributions of the (n, k) third-order Reed-Muller code.

$(128, 64)$	
i	$\|L_i\|$
16	94 488
24	74 078 592
28	3 128 434 688
32	311 574 557 952
36	18 125 860 315 136
40	551 965 599 940 608
44	9 482 818 340 782 080
48	93 680 095 610 142 720
52	538 097 941 223 571 456
56	1 752 914 038 641 131 520
60	2 787 780 190 808 309 760
64	517 329 044 342 046 720

$(256, 93)$	
i	$\|L_i\|$
32	777 240
48	2 698 577 280
56	304 296 714 240
64	74 957 481 580 800
68	707 415 842 488 320
72	28 055 013 884 190 720
76	764 244 915 168 215 040
80	20 661 780 862 988 697 600
84	414 411 510 493 363 568 640
88	6 266 129 424 660 312 883 200
92	71 773 299 826 457 585 909 760
96	627 671 368 441 418 233 282 560
100	4 208 996 769 021 096 823 357 440
104	21 729 928 024 588 603 285 831 680
108	86 666 048 822 136 825 068 912 640
112	267 785 773 787 841 625 294 110 720
116	642 456 218 534 940 726 012 149 760
120	1 198 819 482 820 829 207 341 301 760
124	1 741 767 435 501 050 021 239 848 960
128	1 971 038 877 022 035 145 182 412 800
132	1 735 627 864 909 747 949 509 017 600
136	1 184 951 930 170 762 649 130 762 240
140	620 824 077 435 771 999 611 781 120
144	242 710 219 348 184 804 622 336 000
148	65 293 324 137 047 881 521 561 600
152	8 982 921 659 842 430 396 006 400

Chapter 7

Conclusion

7.1 Summary of the Work

In this dissertation, the error correction capabilities of binary linear codes are investigated. Chapter 1 introduces the problems tackled in the dissertation. The basic definitions and properties of linear codes are provided in Chapter 2.

In Chapter 3, the monotone error structure and its related concept larger halves are used for the analysis of the error correctabilities beyond half the minimum distance. For the first-order Reed-Muller codes, the number of correctable and uncorrectable errors of weight half the minimum distance and half the minimum distance plus one is determined. Also the weight distribution of the minimal uncorrectable errors is derived.

In Chapter 4, the monotone error structure and trial sets are used for the analysis of the error correctabilities. For general linear codes, sufficient conditions under which any trial set contains all the minimum weight codewords are given and are actually satisfied for long Reed-Muller codes and some BCH codes. In addition, for the codes satisfying the sufficient conditions, the lower bounds on the number of uncorrectable errors of weight half the minimum distance is derived. It is shown that the lower bound asymptotically coincides with the corresponding upper bound for Reed-Muller codes and random linear codes.

Chapters 3 and 4 show the usefulness of the monotone error structure and its related notions, larger halves and trial sets, for the error performance analysis of the code. In particular, for the first-order Reed-Muller codes, a simple analysis is given for the error correctability on half the minimum distance. For general linear codes, a nontrivial lower bound on the error correctability on half the minimum distance is derived via trial sets for codes.

In Chapter 5, the relations between the local weight distributions of a code, its extended code, and its even weight subcode is revealed.

In Chapter 6, an algorithm for computing the local weight distribution is proposed. The main ideas of the algorithm are the invariance property of minimal codewords and the coset partitioning. By considering the code tree structure and the invariance property in cosets, the time-complexity of the algorithm is improved. By using the proposed algorithm, the local weight distributions are determined for the third-order $(256, 93)$ and $(128, 64)$ Reed-Muller codes and the $(128, 36), (128, 43)$, and $(128, 50)$ extended primitive BCH codes. From the local weight distributions obtained by the proposed algorithm and the relations between a code, its extended code, and its even weight subcode from Chapter 5, the local weight distributions are determined for the $(255, 93)$ and $(127, 64)$ punctured Reed-Muller code, the $(127, 36), (127, 43)$, and $(127, 50)$ extended primitive BCH codes, and the even weight subcodes of them.

Chapters 5 and 6 develop some methods of determining the local weight distributions for basic codes. However, the distributions obtained in this work are not effective for the improvements of the error probabilities over AWGNC as reported in [42, 41]. The improvements of the probabilities are effective for high-rate codes. Although the codes whose local weight distributions obtained in this work are not high-rate codes, the relations presented in Chapter 5 are not confined to high-rate codes and hence general. If the local weight distribution of some high-rate code is determined, the local weight distribution of the corresponding extended code and even weight subcode can be determined using the relations in Chapter 5.

7.2 Future Directions

In Section 4.5, a lower bound on the uncorrectable errors of weight half the minimum distance for the codes satisfying some condition. That condition is general and simple, which is composed of the minimum distance and the number of minimum weight (and plus one) codewords. Therefore, a generalization of the results to the weights greater than half the minimum distance without too restrictive conditions is a good problem.

A trial set for a code can be used for the minimum distance decoding. However, no nontrivial upper bound on the complexity of the trial set decoding is given so far. The trial set decoding is a type of gradient-like decoding, which includes the minimal codeword decoding. From the results of the size of minimum trial sets presented in Section 4.3, the size of a minimum trial set is smaller than that of minimal codewords. Thus, the decoding complexity of the trial set decoding seems to be less than that of the minimal codeword decoding. Any asymptotic analysis or simulation result is desirable.

For a method of determining the local weight distribution, as noted in Section 6.7, an algorithm for computing the local weight distribution of high-rate codes is desirable, since the bounds on the error probabilities given by the local weight distribution are tight for high-rate codes.

Bibliography

[1] E. Agrell, "Voronoi regions for binary linear block codes," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 310–316, January 1996.

[2] E. Agrell, "On the Voronoi neighbor ratio for binary linear block codes," *IEEE Transactions Information Theory*, vol. 44, no. 7, pp. 3064–3072, November 1998.

[3] A. Ashikhmin and A. Barg, "Minimal vectors in linear codes," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 2010–2017, September 1998.

[4] A. Barg, "Complexity issues in coding theory," in V. Pless and W.C. Huffman, Eds. *Handbook of Coding Theory*, North-Holland, vol. 1, pp. 649–754, 1998.

[5] A. Barg and I. Dumer, "On computing the weight spectrum of cyclic codes," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1382–1386, July 1992.

[6] E.R. Berlekamp, "The technology of error-correcting codes," in *Proceedings of IEEE*, vol. 68, no. 5, pp. 564-593, May 1980.

[7] E.R. Berlekamp and L.R. Welch, "Weight distributions of the cosets of the (32,6) Reed-Muller code," *IEEE Transactions on Information Theory*, vol. IT-18, no. 1, pp. 203–207, January 1972.

[8] Y. Borissov and N. Manev, "Minimal codewords in linear codes," *Serdica Mathematical Journal*, vol. 30, no. 2–3, pp. 303–324, 2004.

[9] Y. Borissov, N. Manev, and S. Nikova, "On the non-minimal codewords in binary Reed-Muller codes," *Discrete Applied Mathematics*, vol. 128, no. 1, pp. 65–74, May 2003.

[10] R.C. Bose, and D.K. Ray Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, pp. 68–79, March, 1960.

[11] R.C. Bose, and D.K. Ray Chaudhuri, "Further results on error correcting binary group codes," *Information and Control*, vol. 3, pp. 279–290, September, 1960.

[12] A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine, "On cryptographic properties of the cosets of $R(1, m)$," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1949–1513, May 2001.

[13] C. Carlet, "Boolean functions for cryptography and error correcting codes," to appear in Y. Crama and P. Hammer, Eds. *Boolean Methods and Models*, Cambridge University Press.

[14] G.C. Clark, Jr. and J.B. Cain, *Error-Correction Coding for Digital Communications*, New York: Plenum, 1981.

[15] Y. Desaki, T. Fujiwara, and T. Kasami, "The weight distributions of extended binary primitive BCH codes of length 128," *IEEE Transactions on Information Theory*, vol. 43, no. 4, pp. 1364–1371, July 1997.

[16] D. Divsalar and E. Biglieri, "Upper bounds to error probabilities of coded systems beyond the cutoff rate," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 2011–2018, December 2003.

[17] G.D. Forney, Jr., "Geometrically uniform codes," *IEEE Transactions on Information Theory*, vol. 37, no. 5, pp. 1241–1260, September 1991.

[18] T. Fujiwara and T. Kasami, "The weight distribution of $(256, k)$ extended binary primitive BCH codes with $k \leq 63$ and $k \geq 207$," *IEICE Technical Report*, IT97-46, September 1997.

[19] T. Helleseth, T. Kløve, and V. Levenshtein, "Error-correction capability of binary linear codes," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1408–1423, April 2005.

[20] A. Hocquenghem, "Codes correcteurs d'erreurs," *Chiffres*, vol. 2, pp. 147–156, 1959.

[21] X. Hou, "$GL(m, 2)$ acting on $R(r, m)/R(r-1, m)$," *Discrete Mathematics*, vol. 149, pp. 99–122, 1996.

[22] X. Hou, "Classification of $R(3, 8)/R(2, 8)$," unpublished.

[23] T.-Y. Hwang, "Decoding linear block codes for minimizing word error rate," *IEEE Transactions on Information Theory*, vol. IT-25, no. 6, pp. 733-737, November 1979.

[24] T. Kasami, T. Tanaka, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 1057–1064, May 1993.

[25] F.J. MacWilliams, "A theorem on the distribution of weights in a systematic code," *Bell Systems Technical Journal*, vol 40, pp. 281–308, 1961.

[26] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, North-Holland, 1977.

[27] J.L. Massey, "Minimal codewords and secret sharing," in *Proceedings of the 6th Joint Swedish-Russian Workshop of Information Theory*, pp. 246-249, 1993.

[28] M. Mohri, Y. Honda, and M. Morii,"A method for computing the local distance profile of binary cyclic codes," *IEICE Transactions on Fundamentals (Japanese Edition)*, vol. J86–A, no. 1, pp. 60–74, January 2003.

[29] M. Mohri, and M. Morii, "On computing the local distance profile of binary cyclic codes," in *Proceedings of International Symposium on Information Theory and Its Applications (ISITA2002)*, pp. 415–418, October 2002.

[30] D.E. Muller, "Application of Boolean algebra to switching circuit design and to error detection," *IRE Transactions on Electronic Computers*, vol. EC-3, pp. 6-12, 1954.

[31] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, Second Edition, MIT Press, 1977.

[32] G. Poltyrev, "Bounds on decoding error probability of binary linear codes via their spectra," *IEEE Transactions on Information Theory*, vol. 40, pp.1284–1292, July 1994.

[33] I.S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Transactions on Information Theory*, vol. IT-4, pp. 38–49, September, 1954.

[34] R.M. Roth, *Introduction to Coding Theory*, Cambridge University Press, 2006.

[35] I. Sason and S. Shamai, "Performance analysis of linear codes under maximum-likelihood decoding: a tutorial," *Foundations and Trends in Communications and*

Information Theory, vol. 3, no. 1-2, pp. 1–222, Now Publishers, Delft, the Netherlands, July 2006.

[36] G.E. Séguin, "A lower bound on the error probability for signals in white Gaussian noise," *IEEE Transactions on Information Theory*, vol. 44, pp. 3168–3175, November 1998.

[37] M. Sudan, *Algorithmic introduction to coding theory*, Lecture 4.5, available from http://people.csail.mit.edu/madhu/FT01/, 2001.

[38] T. Sugita, T. Kasami, and T. Fujiwara, "The weight distribution of the third-order Reed-Muller codes of length 512," *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1622–1625, September 1996.

[39] J.H. van Lint, *Introduction to Coding Theory*, Third Edition, Springer-Verlag, 1999.

[40] C.K. Wu,"On distribution of Boolean functions with nonlinearity $\leq 2^{n-2}$", *Australasian Journal of Combinatorics*, vol. 17, pp. 51–59, March 1998.

[41] T. Yasuda, "Improved lower and upper bounds on the decoding error probability for linear block codes using the local weight distribution," Master thesis, Graduate School of Information Science and Technology, Osaka University, 2006 (*in Japanese*).

[42] T. Yasuda, K. Yasunaga, and T. Fujiwara, "Improvement of the Séguin lower bound using the local weight distribution," in *Proceedings of Symposium on Information Theory and Its Applications (SITA2005)*, pp. 435–438, November 2005 (*in Japanese*).

[43] K. Yasunaga and T. Fujiwara, "Determination of the local weight distribution of binary linear block codes," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4444–4454, October 2006.

[44] K. Yasunaga and T. Fujiwara, "Correctable errors of weight half the minimum distance for the first-order Reed-Muller codes," in *Proceedings of the 29th Symposium on Information Theory and Its Applications (SITA2006)*, pp. 5–8, November 2006.

[45] K. Yasunaga and T. Fujiwara, "On trial set and uncorrectable errors for the first-order Reed-Muller codes," in *Proceedings of 2007 Hawaii and SITA Joint Conference on Information Theory (HISC2007)*, pp. 67–72, May 2007.

[46] K. Yasunaga and T. Fujiwara, "Minimum weight codewords in trial sets," in *Proceedings of the 30th Symposium on Information Theory and Its Applications (SITA2007)*, pp. 562–564, November 2007.

[47] K. Yasunaga and T. Fujiwara, "Correctable errors of weight half the minimum distance plus one for the first-order Reed-Muller codes," in *Proceedings of the 17th Symposium on Applied Algebra, Algebraic Algorithms, and Error Correcting Codes (AAECC-17), Lecture Notes in Computer Science*, vol. 4851, Springer, pp. 110–119, December 2007.

[48] K. Yasunaga, T. Fujiwara, and T. Kasami, "Local weight distribution of the (256, 93) third-order binary Reed-Muller code," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 3, pp. 698–701, March 2007.

[49] S. Yousefi, *Bounds on the performance of maximum-likelihood decoded binary block codes in AWGN interference*, Ph.D. dissertation, the University of Waterloo, 2002.

[50] G. Zémor, "Threshold effects in codes," in *Proceedings of the First French-Israeli Workshop on Algebraic Coding*, Paris, France, 1993, *Lecture Notes in Computer Science*, vol. 781, Springer, pp. 278–286, 1994.