5-2012

Self-dual codes, subcode structures, and applications.

Finley James Freibert
University of Louisville

SELF-DUAL CODES, SUBCODE STRUCTURES, AND APPLICATIONS

By

Finley James Freibert
B.A., DePauw University, 2006
M.A., University of Louisville, 2008

A Dissertation
Submitted to the Faculty of the
College of Arts and Sciences of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Mathematics
University of Louisville
Louisville, KY

May 2012

SELF-DUAL CODES, SUBCODE STRUCTURES, AND APPLICATIONS

Submitted by

Finley James Freibert

A Dissertation Approved on

4/2/2012
(Date)

by the Following Reading and Examination Committee:

Jon-Lark Kim, Dissertation Director

Aly A. Farag

André E. Kézdy

Hamid Kulosman

Robert C. Powers

ACKNOWLEDGMENTS

I thank my advisor, Dr. Jon-Lark Kim, for his great amount of support and inspiration. The first course I attended, on the first day of graduate school, was taught by Dr. Kim. During this course sequence my interest in Coding Theory originated with Dr. Kim's suggestion to give an in-class presentation on the subject. Since that time Dr. Kim has been both a teacher and a mentor to me every semester of my graduate career. I truly value Dr. Kim's patience and the advice he has given me over the years. I thank Dr. André Kézdy for all the advice he has given me over the years. Dr. Kézdy inspired me to begin looking at open problems in discrete mathematics early on in my graduate career; and this was a main influence in my decision to pursue a terminal degree in mathematics. I also thank Dr. Aly Farag, Dr. Hamid Kulosman, and Dr. Robert Powers for their support and the time spent reading and understanding my work. I am thankful to Tim, Chris, Adam, Max, and Kim for all the interesting and insightful discussions we have had and for all the mutual support. I am thankful to Patrick, Joni, Tommy, and my parents for all the encouragement and helpful advice. I thank my brother, Noel, for his constant support. His passionate work ethic and excellence in attaining his goals has inspired me greatly. Finally, I thank my wife, Lauren. Her generosity and support has helped me aspire to follow my dreams.

ABSTRACT

SELF-DUAL CODES, SUBCODE STRUCTURES, AND APPLICATIONS

Finley James Freibert

May 11, 2012

The classification of self-dual codes has been an extremely active area in coding theory since 1972 [33]. A particularly interesting class of self-dual codes is those of Type II which have high minimum distance (called extremal or near-extremal). It is notable that this class of codes contains famous unique codes: the extended Hamming $[8, 4, 4]$ code, the extended Golay $[24, 12, 8]$ code, and the extended quadratic residue $[48, 24, 12]$ code. We examine the subcode structures of Type II codes for lengths up to 24, extremal Type II codes of length 32, and give partial results on the extended quadratic residue $[48, 24, 12]$ code. We also develop a generalization of self-dual codes to Network Coding Theory and give some results on existence of self-dual network codes with largest minimum distance for lengths up to 10. Complementary Information Set (CIS for short) codes, a class of classical codes recently developed in [7], have important applications to Cryptography. CIS codes contain self-dual codes as a subclass. We give a new classification result for CIS codes of length 14 and a partial result for length 16.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Brief History and Introduction

The areas of *Coding Theory* and *Information Theory* date back to 1948 and Claude Shannon's influential paper "A mathematical theory of communication." In 1950, Richard Hamming's paper "Error detecting and error correcting codes" was published. Hamming introduced an important class of codes, the *Hamming Codes*, which were some of the most useful codes at the time due to their error-correcting and detecting capabilities. Since that time many other important codes and classes of codes have been discovered. In 1954 the Reed-Muller codes were described in the papers "Application of Boolean algebra to switching circuit design and to error detection" by Muller and "A class of multiple-error-correcting codes and the decoding scheme" by Reed. The study of *cyclic codes* was begun by Prange in the 1957 report "Cyclic error-correcting codes in two symbols." Generalizations to other fields and other coding schemes have been made as well, such as convolutional codes, turbo codes, and algebraic-geometry codes.

We will focus on the study of an important class of codes, the *self-dual* codes. The study of self-dual codes began with Vera Pless' paper "A classification of self-orthogonal codes over $GF(2)$" in 1972 [33]. Since that time self-dual codes have been one of the most active topics in algebraic coding theory [34, 9, 10, 4]. Self-dual codes are a particularly interesting class of codes due to the fact that

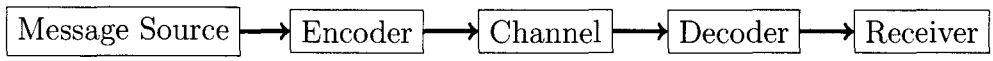
theorems such as those describing the weight distribution and divisibility of these codes follow from their strictly defined structure. These codes have interesting connections to groups, t-designs, lattices, and theta series [18, 26, 36]. Furthermore, many extremal self-dual codes often turn out to be the best among the linear codes with the same parameters. Nevertheless, little attention has been paid to the subcodes of self-dual codes, therefore since the structure and theory behind self-dual codes is so rich we investigate subcodes of self-dual codes and our results are found in Chapter 3. In Chapter 4, we consider the subcode structures of a class of codes called *formally self-dual*. Formally self-dual codes have also been a quite active topic in coding theory [2, 3, 13, 16, 40]. Formally self-dual codes are a generalization of self-dual codes.

In another direction, *Network Coding Theory* is a recent coding scheme which generalizes many concepts from classical Coding Theory. *Network Coding Theory* was introduced by Yeung and Zhang [46] and later expanded upon in other directions [1, 22, 23]. A generalization of the classical concept of duality is what we are interested in for Network Coding. In Chapter 5, we describe some results in this direction.

A final direction we take is to examine a new class of classical codes, called *Complementary Information Set* codes (abbreviated CIS codes), which have applications to cryptography. CIS codes were described in [7], and they contain self-dual codes as a subclass. In Chapter 6 we give some classification results for CIS codes of length 14 and 16. All computations are accomplished using the computer algebra system Magma [6]. As a supplement, in the Appendix, we give the Magma code for computing the equivalency classes of $GL(n, \mathbb{F}_2)$ which is used to classify CIS codes.

1.2 Channel Communication

Consider a non-empty set S of objects. There is a given *source*, given *receiver*, and a *channel* is positioned to connect the source for communication to the receiver. The source constructs a message $\mathbf{x}$ of positive integer length k using the alphabet S. The message is sent over the channel to the receiver. In a perfect world the received message $\hat{\mathbf{x}}$ would be the same as the injected message $\mathbf{x}$. However, if there is any interferance, noise, erasure, or error plaguing the channel, then the message will not be properly transmitted to the receiver. A practical solution to this problem is for the source to *encode* the message $\mathbf{x}$ of length k by incorperating a *redundancy*, so that if there is a resonable amount of damage during transmission the original message may still be recovered. A visualization of this scheme is given in the following diagram adapted from [18, p. 2]:



A more precise mathematical definition is necessary to further meaning and analysis of this system of source-channel-receiver communication. Basic coding theoretical notations and definitions will be derived from [18], while general definitions involving abstract algebra and vector spaces will be based on [14].

1.3 Vector Spaces

Let R be a non-empty set closed under two binary operations addition and multiplication. Given two elements $a, b \in R$ addition will be denoted $a + b$ and multiplication will be denoted ab. R is a *ring* if for any $a, b, c \in R$ the following six properties hold:

(1) $a + b = b + a$

(2) $(a + b) + c = a + (b + c)$

(3) $\exists 0 \in R$ so that $a + 0 = a$ (0 is called an *additive identity*)

(4) $\exists -a \in R$ so that $a + (-a) = 0$ ($-a$ is the *additive inverse* of a)

(5) $a(bc) = (ab)c$

(6) $a(b + c) = ab + ac$ and $(b + c)a = ba + ca$.

A *field* $\mathbb{F}$ is a ring such that for any $a, b \in \mathbb{F}$ the following three properties hold:

(1) $ab = ba$

(2) $\exists 1 \neq 0$ so that $a1 = a$ (1 is a *multiplicative identity*)

(3) if $a \neq 0$ then there exists $a^{-1} \in \mathbb{F}$ so that $aa^{-1} = 1$.

A *vector space* V over a field $\mathbb{F}$ is a non-empty set closed under addition and *scalar multiplication* ($av \in V$ if $a \in \mathbb{F}$ and $v \in V$) such that for any $u, v, w \in V$ and any $a, b \in \mathbb{F}$ the following eight properties hold:

(1) $u + v = v + u$

(2) $(u + v) + w = u + (v + w)$

(3) there exists an *additive identity*) 0 in V

(4) there exists an *additive inverse* $-v$ for all v)

(5) $a(u + v) = au + av$

(6) $(a + b)v = av + bv$

(7) $a(bv) = (ab)v$

(8) $1v = v$.

The elements of a vector space are called *vectors*. A subset U of a vector space V is called a *subspace* of V if U is a vector space over $\mathbb{F}$ under the same operations as V. A *linear combination* of vectors is a sum $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 \cdots + a_n\mathbf{v}_n$ where $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are vectors over $\mathbb{F}$ with coefficients $a_1, a_2, \ldots, a_n \in \mathbb{F}$. Let S be a set of vectors over a field $\mathbb{F}$; S is said to be *linearly dependent* if there exist $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in S$ and $a_1, a_2, \ldots, a_n \in \mathbb{F}$ so that $a_1, a_2, \ldots, a_n$ are not all zero and $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 \cdots + a_n\mathbf{v}_n = 0$. If S is not linearly dependent, then S is said to be *linearly independent*. Given a vector space V over $\mathbb{F}$, a subset B of V is said to be a *basis* for V if B is linearly independent and every element of V may be generated

by a linear combination of vectors from B. A non-trivial vector space with a basis of size k has *dimension* k.

1.4 Linear Codes

Let $\mathbb{F}_q$ denote the finite field with q elements. Let $\mathbb{F}_q^n = \{(a_1, a_2, \ldots, a_n) | a_i \in \mathbb{F}_q\}$; this is the vector space of n-tuples over $\mathbb{F}_q$. For convenience denote a vector $(a_1, a_2, \ldots, a_n)$ from $\mathbb{F}_q^n$ by $a_1 a_2 \ldots a_n$. Any subset C of a vector space is called a *code* and in particular C is an (n, M) *code* over $\mathbb{F}_q$ if C is a subset of size M from $\mathbb{F}_q^n$. Any element of a code C is called a *codeword.* Codes over $\mathbb{F}_2$, $\mathbb{F}_3$, and $\mathbb{F}_4$ are respectively called *binary*, *ternary*, and *quaternary codes.*

A code C is a *linear code* if it is a subspace of $\mathbb{F}_q^n$; otherwise, C is a *non-linear code.* C is an $[n, k]$ *code* if C is a linear code of dimension$k$. An $[n, k]$ linear code has q^k codewords. A $k \times n$ matrix G is called a *generator matrix* for an $[n, k]$ code C if the rows of G form a basis for C. Given a code C with generator matrix G and any set of k linearly independent columns of G, the positions corresponding to the independent columns form an *information set* for C, and the remaining $r = n - k$ positions are called a *redundancy set* and r is the *redundancy* of C.

Thus, the channel communication, discussed in the first section, may be implemented with linear codes. With the alphabet $\mathbb{F}_q$, construct a message $\mathbf{x} = x_1 x_2 \ldots x_k \in \mathbb{F}_q^k$. Using a linear $[n, k]$ code C with generator matrix G, encode the message as the codeword $\mathbf{c} = \mathbf{x}G = c_1 c_2 \ldots c_n \in C$. Transmit the codeword over the channel and error may be accumulated, this is modeled by an *error* vector $\mathbf{e} = e_1 e_2 \ldots e_n \in \mathbb{F}_q^n$. The vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$ is received and *decoded* to an estimate $\hat{\mathbf{x}}$ of the original message.

The *Hamming Weight* of a vector $\mathbf{x} \in \mathbb{F}_q^n$, denoted $wt(\mathbf{x})$ is the number of nonzero coordinates of $\mathbf{x}$. The *Hamming Distance* between two vectors $\mathbf{x}, \mathbf{y} \in$

$\mathbb{F}_q^n$, denoted $d(\mathbf{x}, \mathbf{y})$ is the number of coordinates in $\mathbf{x}$ and $\mathbf{y}$ which are different. The distance function satisfied the following properties of a *metric* on $\mathbb{F}_q^n$: for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_q^n$,

(1) $d(\mathbf{x}, \mathbf{y}) \geq 0$

(2) $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$

(3) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$

(4) $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$.

Given a code C, $d(C) := min\{wt(\mathbf{x}) : \mathbf{x} \in C$ and $\mathbf{x}$ is non-zero $\}$ denotes the *minimum weight* of C; if C is a linear code then $d(C)$ is also the *minimum distance* of C. A linear code of length n, dimension k, and minimum distance d will be called an $[n, k, d]$ code. If a code has minimum distance greater than or equal to d it will be called an $[n, k, \geq d]$ code

Two binary codes are said to be *equivalent* if there exists a permutation of coordinates mapping one code onto the other code. An $[n, k]$ binary code is said to be *unique* if it is the only code of length n and dimension k up to equivalence.

1.5 Self-Dual Codes

Given two vectors in $\mathbb{F}_q^n$, $v = a_1 a_2 \ldots a_n$ and $u = b_1 b_2 \ldots b_n$, the usual *dot product* of $\mathbf{u}$ with $\mathbf{v}$ is the sum $\mathbf{u} \cdot \mathbf{v} := a_1 b_1 + a_2 b_2 \cdots + a_n b_n$. This dot product is an *inner product* on a vector space V as it satisfies the following three properties for any $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and any $a, b \in \mathbb{F}$:

(1) $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$

(2) $(a\mathbf{u} + b\mathbf{v}) \cdot \mathbf{w} = a(\mathbf{u} \cdot \mathbf{w}) + b(\mathbf{v} \cdot \mathbf{w})$

(3) for fixed $\mathbf{u} \in V$ if $\mathbf{u} \cdot \mathbf{v} = 0$ for all $\mathbf{v} \in V$, then $\mathbf{u} = 0$.

Two vectors $\mathbf{u}$ and $\mathbf{v}$ are *orthogonal* if $\mathbf{u} \cdot \mathbf{v} = 0$. If C is a linear code, then the set $C^{\perp} := \{\mathbf{x} \in V : \mathbf{x} \cdot \mathbf{v} = 0 \ \forall \mathbf{v} \in C\}$. In particular, if C is contained in $C^{\perp}$,

then C is called *self-orthogonal*. Note that $C^{\perp}$ is always a subspace of V. In fact, $dim(C) + dim(C^{\perp}) = dim(V)$ (which follows from the kernel extension theorem of linear transformations). Hence if C is an $[n, k]$ code, then $C^{\perp}$ is an $[n, n-k]$ code. Further, if C is an $[n, k]$ code and $C = C^{\perp}$, then C is called *self-dual* and $k = \frac{n}{2}$; this implies that the length of any self-dual code must be even.

A self-dual code is called *Type I* (or *singly-even*) if it contains a codeword c such that $wt(c) = 2 \pmod 4$; otherwise, the self-dual code is called *Type II* (or *doubly-even*) as all codewords are divisible by 4. Type II codes of length n exist if and only if n is a multiple of 8 (this fact follows from Gleason Polynomials, p.344 in [18]). If C is an $[n, k, d]$ self-dual binary code then the minimum distance has the following upper bound [36].

$$d \leq \begin{cases} 4\lceil \frac{n}{24} \rceil + 4 & : n \neq 22 \pmod{24}, \\ 4\lceil \frac{n}{24} \rceil + 6 & : n = 22 \pmod{24}. \end{cases}$$

A self-dual code meeting this bound is called *extremal*.

1.6 Weight Distribution

Let A_w be the set of all vectors in a code C with weight w. The *weight distribution* of C is the list $A_0, A_1, ..., A_n$; however, if $A_i = 0$ then we omit A_i from the list. A code which has the same weight distribution as its dual is called *formally self-dual* .

A classical theory by MacWilliams relates the weight distribution of a code and its dual.

Lemma 1.1. *([26, the MacWilliams Identities, p. 129]) Let C be an $[n, k]$ code and denote A_w and $A_w^{\perp}$ by the number of codewords of weight w in the code C and $C^{\perp}$ respectively. Then*

$$\sum_{i=0}^{n} A_i P_w(n, i) = 2^k A_w^{\perp}, \quad for \ \ 0 \leq w \leq n,$$

where $P_w(n,i) = \sum_{j=0}^{w}(-1)^j \begin{pmatrix} i \\ j \end{pmatrix} \begin{pmatrix} n-i \\ w-j \end{pmatrix}$ *is a Krawtchouk polynomial.*

Lemma 1.1 is especially useful in determining the non-existence of a code. Non-existence is determined when a possible weight distribution has negative or non-integer values. In Section 3.5 this lemma is invoked to prove the non-existence of particular subcodes of the extended quadratic residue code. Lemma 1.1 is also applied to determine the possible weight distribution of a subcode, if it exists.

1.7 Obtaining New Codes from Previous Codes

Let C be an $[n,k,d]$ code over $\mathbb{F}_q$. Let T be a set of t coordinates. The code C^T, obtained by deleting the coordinate positions in T, is called the *code punctured on* T Let $C(T)$ be the set of codewords of C which are $\mathbf{0}$ on T. We puncture $C(T)$ on T to get a linear code of length $n-t$ called the *code shortened on* T and denoted by C_T [18].

Lemma 1.2. *([18, Theorem 1.5.7]) Let* C *be an* $[n,k,d]$ *code over* $\mathbb{F}_q$. *Let* T *be a set of* t *coordinates. Then the following hold:*
(a) $(C^\perp)_T = (C^T)^\perp$ *and* $(C^\perp)^T = (C_T)^\perp$).
(b) If $t < d$, *then* C^T *and* $(C^\perp)_T$ *have dimensions* k *and* $n-t-k$, *respectively.*
(c) If $t = d$ *and* T *is the set of coordinates where a minimum weight codeword is nonzero, then* C^T *and* $(C^\perp)_T$ *have dimensions* $k-1$ *and* $n-d-k+1$, *respectively.*

Lemma 1.2 is useful in examining a code with particular parameters with known dual distance. It can be used in conjunction with Lemma 1.1 to prove the non-existence of a code as in Section 3.5.

CHAPTER 2
MAXIMAL SUBCODES AND OPTIMUM DISTANCE PROFILES

2.1 Motivations

One of the main problems that has arisen in Coding Theory is the search for optimal codes with the largest size given a minimum distance or optimal codes with the largest minimum distance given a size [18, 26]. There has been extensive work in this direction [15]. Some well-known families of codes, such as the Reed-Muller codes or the cyclic codes, contain notable subcodes. However, comparatively little attention has been paid to the subcodes of an optimal linear code in general. It is a natural concern to determine which linear codes contain optimal (or near-optimal) subcodes. Among linear codes, we suggest self-dual, self-orthogonal, or formally self-dual even codes since their possible non-zero weights jump by 2 or 4. Thus there is a possibility to get subcodes with a large minimum distance.

We show that in many cases optimal subcodes can be obtained by computing optimum distance profiles (ODPs), a concept introduced by Luo, Han Vinck, and Chen [25]. The authors [25] considered how to construct and then exclude (or include, respectively) the basis codewords one by one while keeping a distance profile as large as possible in a dictionary order (or in an inverse dictionary order, respectively). Thus fault-tolerant capability is improved by selecting subcodes in communications and storage systems. The practical applications are found in WCDMA [17], [41] and address retrieval on optical media [42].

In [8] and [25], the authors give results on the ODPs of the binary Hamming $[7, 4, 3]$ code, the binary and ternary Golay codes, Reed-Solomon codes, the first-order and second order Reed-Muller codes. Since self-dual codes and formally self-dual codes are an interesting class of linear codes whose subcode structure and ODPs are not known yet, we propose to examine these codes in the following chapters.

Recently, Yan, et. al. [45] considered the optimum distance profiles of some quasi-cyclic codes and proposed two algorithms, called the "subcodes traversing algorithm" and "supercodes traversing algorithm." These algorithms enumerate all subcodes of a given code. Hence they are rather inefficient in finding ODPs of linear codes with a relatively large dimension. Their examples have dimension 10 only. Therefore we propose two full algorithms based on cosets, called the Chain Algorithms, and two random algorithms to find ODPs of the codes. These algorithms look at a chain of subcodes of a given code and consider the equivalence of the codes with the same dimension. Hence they are more efficient than the subcodes and supercodes traversing algorithm [45].

The following concept of optimal codes is proposed in [18, p. 53].

Definition 2.1. Let n and k be positive integers so that $k \leq n$. A linear $[n, k, d]$ code is *minimum distance optimal* if d is the maximum possible minimum distance among all $[n, k]$ codes. Given n and d, a linear $[n, k, d]$ code is *dimension optimal* if k is the largest possible.

Grassl's online table [15] is a good source for optimal code parameters given reasonable lengths and dimensions for finite fields of order up to 9.

Definition 2.2. Let C be a linear code. A subcode C_1 of C is *maximal* if there is no subcode $C_2 \neq C$ of C such that $C_1 \subsetneq C_2$ and $d(C_2) = d(C_1)$. Given $d' > d(C)$

the maximum dimension in the set

$$\{dim(C_i) : d' = d(C_i) \text{ and } C_i \text{ is a maximal subcode of } C\}$$

is called the *maximum dimension corresponding to* d'.

2.2 Optimum Distance Profiles of Codes

The concept of the optimum distance profile of a linear code was introduced in [8], [25] for details. We use the same basic definitions as these authors, although we will use a slightly different notation for the ODP entries which is more intuitive. Let C be a binary $[n, k]$ code and let $C_0 = C$. A sequence of linear subcodes of C, $C_0 \supset C_1 \supset \cdots \supset C_{k-1}$ is called a *subcode chain* , where the dimension of C_i is $k - i$ for $i = 0, \ldots k - 1$. Let $d_i := d(C_i)$ be the minimum distance of C_i. Then the sequence $d_0 \leq d_1 \leq \cdots \leq d_{k-1}$ is called a *distance profile* of C. For the given code C a *generator matrix with respect to the distance profile* is a generator matrix of C where the top $k - i$ rows generate C_i for $0 \leq i \leq k - 1$ (i.e., deleting the bottom i rows forms a generator matrix for C_i).

For any two integer sequences of length k, $a = a_0, \ldots, a_{k-1}$ and $b = b_0, \ldots, b_{k-1}$, a is called an *upper bound on* b *in the dictionary order* if a is equal to b or there is an integer t such that

$$a_i = b_i \text{ for } 0 \leq i \leq t - 1, \text{ and } a_t > b_t.$$

Similarly, a is called an *upper bound on* b *in the inverse dictionary order* if a is equal to b or there is an integer t such that

$$a_i = b_i \text{ for } t + 1 \leq i \leq k - 1, \text{ and } a_t > b_t.$$

It is noted that dictionary and inverse dictionary orders are analogous to the concepts of lexicographical order and reverse lexicographical order.

Definition 2.3. A distance profile of the linear code is called the *optimum distance profile* (or abbreviated *ODP*) *in the dictionary order* , denoted by $\mathrm{ODP}^{dic}[C](0)$, $\mathrm{ODP}^{dic}[C](1), \ldots, \mathrm{ODP}^{dic}[C](k-1)$ if it is an upper bound on any distance profile of C in the dictionary order. Similarly, a distance profile of the linear block code is called the *optimum distance profile in the inverse dictionary order* , denoted by $\mathrm{ODP}^{inv}[C](0)$, $\mathrm{ODP}^{inv}[C](1), \ldots, \mathrm{ODP}^{inv}[C](k-1)$ if it is an upper bound on any distance profile of C in the inverse dictionary order. We also use $\mathrm{ODP}[C]$ to denote the optimum minimum distance profile in both orders.

The ODP of a code and the maximum dimension with respect to a minimum distance are related concepts. Note that the first minimum distance d' to appear in the ODP in dictionary order corresponds to a maximal subcode with maximum dimension corresponding to d'. However, after this term, maximal subcodes in the subcode chain do not necessarily imply the maximum dimension. This is an observation which follows from the definition of a maximal subcode and the definition of ODP; we formalize the theory in the following lemmas. However, note that given a dimension $k' \leq k$ there may be multiple minimum distances d' with respect to which k' is the maximum dimension. Therefore for the first lemma we define $d_{k'}$ to be the maximum of such minimum distances.

Lemma 2.4. *Let C be an $[n, k]$ code. Let $k' \leq k$ be given. Define $d_{k'} = \max(\{d' : k'$ is the maximum dimension in C with respect to $d'\})$ and define d_{opt} to be the optimal minimum distance attained among all $[n, k']$ codes (many values available at [15]), then*

$$d_{opt} \geq d_{k'} \geq \max(\{ODP^{dic}[C]_{k'}, ODP^{inv}[C]_{k'}\}).$$

Proof. The claim $d_{opt} \geq d_{k'}$ is clear since d_{opt} is the maximum minimum distance possible among all $[n, k']$ codes. By the definition of $d_{k'}$, if C contains an

$[n, k', d']$ subcode, then $d_{k'} \geq d'$. Since $\mathrm{ODP}^{dic}[C]_{k_i}$ (respectively $\mathrm{ODP}^{inv}[C]_{k_i}$) corresponds to a dimension k_i subcode in the subcode chain having minimum distance $\mathrm{ODP}^{dic}[C]_{k_i}$ (respectively $\mathrm{ODP}^{inv}[C]_{k_i}$), the preceding claim proves the lemma. □

Corollary 2.5. *Let C be an $[n, k]$ code. Let $k' \leq k$ be given. Define $d_{k'}$ and d_{opt} as above. If $ODP^{dic}[C]_{k'} = d_{opt}$ or $ODP^{inv}[C]_{k'} = d_{opt}$, then equality is implied in the above lemma: $d_{opt} = d_{k'} = \max(\{ODP^{dic}[C]_{k'}, ODP^{inv}[C]_{k'}\})$.*

The necessity of defining $d_{k'}$, in Lemma 2.4, as a maximum is due to the fact that there may be multiple minimum distances yielding the same maximum dimension. An example where this occurs is the following:

Example 2.6. Let C be the [6,3,1] code with the following generator matrix:

$$G = \begin{bmatrix} 11 & 11 & 00 \\ 11 & 00 & 11 \\ 10 & 00 & 00 \end{bmatrix}$$

The maximum dimension with respect to $d_1 = 4$ is 2, due to the fact that the first two rows of G generate a [6,2,4] subcode of C with the following generator matrix:

$$G_1 = \begin{bmatrix} 11 & 11 & 00 \\ 11 & 00 & 11 \end{bmatrix}.$$

Similarly, the maximum dimension with respect to $d_2 = 3$ is 2; this is obtained by adding the third row of G to each row in G_1 which yields a $[6, 2, 3]$ subcode of C with the following generator matrix:

$$G_2 = \begin{bmatrix} 01 & 11 & 00 \\ 01 & 00 & 11 \end{bmatrix}$$

Notice that in Lemma 2.4 we fix the dimension k'; a dual statement where we instead fix the minimum distance is the following.

Lemma 2.7. *Let C be an $[n,k]$ code and let $0 \leq j \leq k-1$. Suppose d_j is a minimum distance appearing as $ODP^{dic}[C]_j$ or $ODP^{inv}[C]_j$. Define k_j to be the maximum dimension with respect to d_j, then $k_j \geq j$.*

Proof. The proof follows directly from the definition of maximal dimension with respect to d_j, since a subcode with this maximal dimension will have dimension k_j which is an upper bound on the dimension of any $[n, j, d_j]$ subcode. □

The following lemma is a special case of Lemma 2.7; this lemma states that in fact the first minimum distance in the dictionary order ODP corresponds to a maximal subcode with respect to that minimum distance.

Lemma 2.8. *Let C be an $[n,k,d]$ code. Suppose that for some j, $ODP^{dic}[C]_j$ is the first term in ODP greater than d. Then j is the maximum dimension with respect to $ODP^{dic}[C]_j$.*

Proof. If $\mathrm{ODP}^{dic}[C]_j$ is the first term in ODP greater than d, then $\mathrm{ODP}^{dic}[C]_{j+1} = d$ where $0 < j < k$. Suppose to the contrary that j is greater than the maximum dimension with respect to $\mathrm{ODP}^{dic}[C]_j$, then there must exist an $[n, j+1]$ subcode with minimum distance $\mathrm{ODP}^{dic}[C]_j$. This implies $\mathrm{ODP}^{dic}[C]_{j+1} = \mathrm{ODP}^{dic}[C]_j$ by definition of the dictionary order. Compiling this information we obtain the contradiction: $d = \mathrm{ODP}^{dic}[C]_{j+1} = \mathrm{ODP}^{dic}[C]_j > d$. □

If a code contains an optimal subcode (minimum distance optimal, dimension optimal, or both) there are many cases where this subcode appears in the subcode chain involved in an optimum distance profile. However, this is not always the case as in the following example:

Example 2.9. Let C be the [6,5,1] code with the following generator matrix:

$$G = \begin{bmatrix} 11 & 11 & 00 \\ 11 & 00 & 11 \\ 10 & 10 & 10 \\ 10 & 10 & 00 \\ 10 & 00 & 00 \end{bmatrix}$$

By expurgating weight 1 vectors from C we may obtain $[6, 4, 2]$ subcodes of C. Since there does not exist a $[6, 4, 3]$ code (see [15]), we may conclude that $\mathrm{ODP}^{dic}[C]_4 = 2$. By examining all $[6, 4, 2]$ subcodes of C it can be determined that none contain a $[6, 3, 3]$ subcode, and since no $[6, 3, 4]$ code exists we obtain $\mathrm{ODP}^{dic}[C]_3 = 2$. Finally, there is a unique $[6, 2, 4]$ code (which has a single non-zero weight of 4); as this code is a subcode of at least one $[6, 4, 2]$ subcode of C, and since there does not exist a $[6, 2, 5]$ code we may conclude $\mathrm{ODP}^{dic}[C]_2 = 4$ and $\mathrm{ODP}^{dic}[C]_1 = 4$. Therefore the optimum distance profile in dictionary order is $\mathrm{ODP}^{dic}[C] = [1, 2, 2, 4, 4]$.

Using similar arguments the ODP in inverse dictionary order is obtained as $\mathrm{ODP}^{inv}[C] = [1, 2, 2, 3, 5]$. Notice that the first three rows of G generate an optimal [6,3,3] code (both minimum distance optimal and dimension optimal). Therefore the maximum dimension with respect to minimum distance $d' = 3$ is $k' = 3$. However, the subcodes of dimension 3 appearing in both ODP orders have minimum distance 2. An explanation for this phenomenon is that all supercodes of the [6,3,3] code in C have minimum distance 1. This is an example where equality is not possible in Lemma 2.4 and in Lemma 2.7.

2.3 Equivalent Subcodes and Cosets in Binary Self-Dual Codes

The following theory of equivalent codes is adapted from section 1.6 in [18]. These theories motivate the algorithms presented in the next section.

Lemma 2.10. *Let C be a code and P be a permutation matrix. $(CP)^{\perp} = C^{\perp}P$.*

Proof. The proof follows from a series of equivalent statements which imply containment. First, to show $(CP)^{\perp} \subset C^{\perp}P$, let $y \in (CP)^{\perp}$ which is equivalent to $y \cdot x = 0 \ \forall x \in CP$ by definition of the dual. Since each codeword in CP may be represented as x_0P for some $x_0 \in C$ we have $y \cdot (x_0P) = 0 \ \forall x_0 \in C$. Now applying permutations to y and x_0P will not change the dot product value, hence $yP^{-1} \cdot (x_0P)P^{-1} = 0 \ \forall x_0 \in C$. Simplifying we obtain $yP^{-1} \cdot x_0 = 0$ which implies $yP^{-1} \in C^{\perp}$ adn equivalently $y \in C^{\perp}P$. Therefore $(CP)^{\perp} \subset C^{\perp}P$. $C^{\perp}P \subset (CP)^{\perp}$ is shown in an equivalent manner. □

Corollary 2.11. *Given two binary codes C_1 and C_2; $C_1P = C_2$ for some permutation matrix P if and only if ${C_1}^{\perp}P = {C_2}^{\perp}$.*

Proof. $C_1P = C_2$ is equivalent to $(C_1P)^{\perp} = C_2^{\perp}$ by definition of the dual. $(C_1P)^{\perp} = C_2^{\perp}$ is equivalent to${C_1}^{\perp}P = C_2^{\perp}$ by Lemma 2.10. □

Corollary 2.12. *Given two binary codes C_1 and C_2 such that $C_1P = C_2$ for some permutation matrix P; if C_1 is self-orthogonal, then C_2 is self-orthogonal.*

Proof. C_1 is self orthogonal implies that $C_1 \subset {C_1}^{\perp}$. Hence $C_1P \subset {C_1}^{\perp}P$, which implies $C_1P \subset (C_1P)^{\perp}$ by Lemma 2.10. Therefore $C_2 \subset {C_2}^{\perp}$, so C_2 is self-orthogonal. □

Given a linear code C and a subcode C' the *codimension* of C' in C is the difference $dim(C) - dim(C')$.

Lemma 2.13. *Given a binary $[n, k]$ code C there is a one-to-one correspondence between codimension 1 subcodes of C and $[n, n-k+1]$ supercodes of $C^{\perp}$.*

Proof. Let S_1 be the set of all $[n, k-1]$ inequivalent subcodes in C and let S_2 be the set of all $[n, n-k+1]$ inequivalent supercodes of $C^{\perp}$. Consider the map

$\perp$: $S_1 \to S_2$. Clearly $\perp^{-1} \equiv \perp$. Clearly $\perp$ is surjective since for any $D \in S_2$, $C^\perp \subset D$ implies $D^\perp \subset C$ and $D^\perp$ is $[n, k-1]$ and hence $D^\perp \in S_1$. The fact that $\perp$ is injective follows from Corollary 2.11 as for any $D \not\equiv D' \in S_1$ implies $D^\perp \not\equiv D'^\perp$. $\square$

2.4 Algorithms to Output Maximal Subcodes

Given an $[n, k, d]$ code C which has small length and dimension it may be relatively easy to examine its subcode structure by a brute force generation of all possible subcodes. However, as length and dimension increase this method becomes very time consuming; this is why we propose four algorithms which are relatively efficient in comparison to the brute force search. The notions of equivalence outlined in Lemmas 2.10, 2.11, 2.12 imply that it is redundant to consider equivalent subcodes. The first two algorithms are exhaustive in the sense that when applying them we obtain a complete list of inequivalent subcodes (respectively supercodes), with prescribed minimum distance, contained in (respectively containing) the given code C; in this way, the redundant cases considered in a brute force search are eliminated. The two "Random" Algorithms are especially useful for very large length and dimension, where the exhaustive search is infeasible. The Random Algorithms can also give results much faster than the Chain Algorithms since all cases are not considered. Our first algorithm, (Subcodes) Chain Algorithm I, directly uses Lemma 2.13 to search for subcodes.

(Subcodes) Chain Algorithm I: An algorithm to produce all maximal subcodes with maximum dimension k' and minimum distance $d' \geq d$.

1. Input: Begin with a binary $[n, k, d]$ code D and a positive integer $d' \geq d$ (such that there exists a codeword of weight d' in D).

2. Output: Produce the maximum dimension k' among all maximal subcodes with minimum distance d' and a list of inequivalent maximal subcodes of this dimension and minimum distance d'.

   (a) Initialize the set $\mathbf{B}_1 = \{D^\perp\}$. Begin with $i = 1$.

   (b) Build a set $\mathbf{B}_{i+1}$ of all inequivalent supercodes of dimension 1 higher of C for all $C \in \mathbf{B}_i$. In order to do this we add coset representatives from $\mathbb{F}_q^n/C$ to each code C in $\mathbf{B}_i$.

   (c) Check if $d(C^\perp) = d'$ for any code $C \in \mathbf{B}_{i+1}$. If "No" for any $C \in \mathbf{B}_{i+1}$, then repeat step (ii) by increasing i to $i + 1$. If "Yes", then output the maximum dimension $k' = k - i + 1$ and the set of $[n, k - i + 1, d']$ subcodes of D.

(Supercodes) Chain Algorithm II: An algorithm to find all $[n, k, d]$ supercodes containing an $[n, k', d']$ code with $d' \geq d$ and $k \geq k'$

1. Input: Begin with a set $\mathbf{C}_{k',d'}$ of inequivalent $[n, k', d']$ codes (respectively self-orthogonal codes) with $k \geq k'$ and $d' \geq d$.

2. Output: For each code C in $\mathbf{C}_{k',d'}$, produce all $[n, k, d]$ codes (respectively self-orthogonal codes) containing C.

   (a) Begin by building a set of all inequivalent supercodes (respectively self-orthogonal supercodes) of dimension 1 higher of each code C in $\mathbf{C}_{k',d'}$ with minimum distance greater than or equal to d. In order to do this we add coset representatives from $\mathbb{F}_q^n/C$ (respectively $C^\perp/C$ if C is self-orthogonal) to each code C in $\mathbf{C}_{k',d'}$ and keep a set of inequivalent supercodes $\mathbf{C}_{k'+1}$ generated in this way.

(b) Repeat the first step, by replacing $\mathbf{C}_{k',d'}$ with $\mathbf{C}_{k'+1}$ until the set of inequivalent codes which are generated have dimension k.

(c) Stop once dimension k is reached. For each code C in $\mathbf{C}_{k',d'}$ output all $[n,k,d]$ supercodes of C.

Example 2.14. As an example, we determine the ODPs for the four optimal $[28,7,12]$ self-complementary codes classified in [11]. These codes are doubly-even with non-zero weights 12,16,28. We begin with a $[28,3,16]$ constant weight code (meaning the only non-zero weight is 16). There is only one such code due to the fact that all non-zero codewords must intersect in exactly 8 positions; if the first two basis vectors are fixed, then there is only one possibility (up to coordinate permutation) for the third basis vector. By adding the all-one vector to the constant weight code we obtain a $[28,4,12]$ code with the following generator matrix:

$$G_{[28,4,16]} = \begin{bmatrix} 1111 & 0000 & 0000 & 1111 & 0000 & 1111 & 1111 \\ 0000 & 1111 & 0000 & 1111 & 1111 & 0000 & 1111 \\ 0000 & 0000 & 1111 & 1111 & 1111 & 1111 & 0000 \\ 1111 & 1111 & 1111 & 1111 & 1111 & 1111 & 1111 \end{bmatrix}$$

Applying (Supercodes) Chain Algorithm II to this generator matrix (and only keeping doubly-even supercodes) we obtain all four self-complementary $[28,7,12]$ codes with the following generator matrices:

$$\begin{bmatrix} G_{[28,4,16]} \\ 0100010001011010010010111001 \\ 0010011101110111001111001100 \\ 0001000100011110010101010011 \end{bmatrix}, \begin{bmatrix} G_{[28,4,16]} \\ 0100010001001101001110101010 \\ 0010011101110111001111001100 \\ 0001000100011110010101010011 \end{bmatrix},$$

$$\begin{bmatrix} G_{[28,4,16]} \\ 0100010001001011010100110110 \\ 0010011101110111001111001100 \\ 0001000100011110010101010011 \end{bmatrix}, \begin{bmatrix} G_{[28,4,16]} \\ 0101001100111010000010011010 \\ 0011000000110011001100110011 \\ 0000011001011010010110101100 \end{bmatrix}.$$

Let C be any $[28, 7, 12]$ self-complementary code. Since the $[28, 3, 16]$ subcode is optimal, in light of Lemma 2.8, we determine $\mathrm{ODP}^{dic}[C]_3 = 16$. As a [28,3,16] subcode cannot contain the all-one vector, we determine the ODP in dictionary order:

$$\mathrm{ODP}^{dic}[C] = [12, 12, 12, 12, 16, 16, 16].$$

The ODP in inverse order is clear since any supercode of the repetition code, containing a weight 16 vector, must also contain a weight 12 vector. Hence

$$\mathrm{ODP}^{inv}[C] = [12, 12, 12, 12, 12, 12, 28].$$

We now introduce the random algorithms:

Random (Subcodes) Algorithm I: An algorithm to search for maximal subcodes

1. Input: A linear code C with parameters $[n, k, d]$ and $d' > d$ where $A_{d'}$ is non-zero.

2. Output: A maximal subcode C' of C with d'.

   (a) Take any codeword x from C such that $\mathrm{wt}(x) \geq d'$. Let $C_1 =< x >$.

   (b) Choose any coset representative y of C/C_1. Let $C_1 :=< y > +C_1$. Repeat this until $d(C_1) = d'$.

(c) Repeat (2) until there is no coset representative such that $d(C_1) = d'$. Let $C' := C_1$.

The below algorithm is somewhat opposite to Random Algorithm I.

Random (Supercodes) Algorithm II: An algorithm to search for codes containing good codes

1. Input: A (best known) linear code C_1 with parameters $[n, k, d]$ and $d' < d$.
2. Output: A code C' containing C_1 with d' and $k' > k$.

   (a) Let C:=$C_1{}^{\perp}$.

   (b) Choose any coset representative y of C/C_1. Let $C_1 :=< y > +C_1$. Repeat this until $d(C_1) = d'$.

   (c) Repeat (2) until there is no coset representative such that $d(C_1) = d'$. Let $C' := C_1$.

Analysis and comparison of our algorithms: Our Chain Algorithms reduce the complexity of calculation by checking in each round the equivalence of all the codes of the same dimension in *chains of codes* obtained from a set of given codes. This is one of the two time consuming steps. Another time consuming step is to consider all coset representatives from $\mathbb{F}_q^n/C$. On the other hand, the algorithms given in Yan, et. al. [45] construct all subcodes of the same dimension not necessarily in chains of codes. Hence their algorithms are computing more than needed (hence less efficient) in calculating ODPs of linear codes. For example, a brute-force search of the subcodes of dimension k' for an $[n, k]$ code has complexity given by the Gaussian binomial coefficient $\begin{bmatrix} k \\ k' \end{bmatrix}_2$. In Section 3.4 for some $[32, 16, 8]$ codes we determine the maximum dimension subcode with respect to $d = 12$ to have dimension 11. A brute-force subcode search (such as the subcodes traversing

algorithm in [45]) would have to enumerate $\begin{bmatrix} 16 \\ 11 \end{bmatrix}_2 = 120,843,139,740,969,555$ subcodes; this task is not feasible.

Example 2.15. Using their traversing algorithms, the authors [45] have determined ODPs of a quasi-cyclic $[48, 10, 20]$ code C_{48} by finding all k-dimensional subcodes of C which is extensive work. Using the above Random Algorithms, we have also computed ODPs of C_{48} in the *dictionary* and *inverse dictionary* orders *in a minute* as follows:

$$\mathrm{ODP}^{dic}[C_{48}] = [20, 20, 20, 20, 24, 24, 24, 24, 32, 32],$$

$$\mathrm{ODP}^{inv}[C_{48}] = [20, 20, 20, 20, 20, 20, 20, 24, 28, 36].$$

CHAPTER 3
SUBCODES AND OPTIMUM DISTANCE PROFILES OF SELF-DUAL CODES

3.1 Motivations

We plan to construct optimal (self-orthogonal) subcodes of a given linear (self-dual) code. In order to construct finite-state codes, Pollara, Cheung and McEliece [35] constructed the first $[24, 5, 12]$ subcode of the binary Golay $[24, 12, 8]$ code, improving a previously known $[24, 5, 8]$ subcode. Maks and Simonis [30] have shown that there are exactly two inequivalent $[32, 11, 12]$ codes in the binary Reed-Muller code $R(2, 5)$ which contain $R(1, 5)$ and have the weight set $\{0, 12, 16, 20, 32\}$.

In this section, we give the ODPs of Type II self-dual codes of lengths up to 24 and the five extremal Type II codes of length 32, give a partial result of the ODP of the extended quadratic residue code q_{48} of length 48, and give some directions towards finding optimal self-orthogonal codes of length 72.

3.2 Optimal Subcodes of Type II Codes for $n \leq 16$

In this section, we begin with an example.

Example 3.1. There exists a unique Type II code of length 8 (p.29 of [18]) . This $[8, 4, 4]$ code is the Extended Hamming $[7, 4, 3]$ code denoted e_8 in [18]. Since this code is doubly-even and self-dual it contains weights 0,4, and 8. For $d' = 8$ it is clear that there exists a unique subcode $\langle \mathbf{1} \rangle$ of e_8. It is also clear to see that $\langle \mathbf{1} \rangle$ is

a maximal subcode and an optimal code.

There are two Type II [16, 8, 4] codes denoted by d_{16} and $2e_8$ (in [18]).

Lemma 3.2. *d_{16} contains a unique optimal $[16, 5, 8]$ code which is a maximal subcode with respect to minimum distance 8. Also, 5 is the maximum dimension corresponding to $d' = 8$.*

Proof. We construct a maximal subcode C_1, of d_{16}, with minimum distance 8. The maximum dimension possible for C_1 is $k = 5$ as there is no [16, 6, 8] code by [15]. Now consider the following form of the generator matrix of d_{16} with row vectors labeled as $\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, y\}$ and where A is a matrix with rows $\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$:

$$\left[\begin{array}{cccccccc} 11 & 11 & & & & & & \\ 11 & & 11 & & & & & \\ 11 & & & 11 & & & & \\ 11 & & & & 11 & & & \\ 11 & & & & & 11 & & \\ 11 & & & & & & 11 & \\ 11 & & & & & & & 11 \\ \hline 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right] = \left[\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ \hline y \end{array}\right] = \left[\begin{array}{c} A \\ \hline y \end{array}\right].$$

Define the map $\phi_1 : \{00, 11\} \to \{0, 1\}$ where $00 \mapsto 0$ and $11 \mapsto 1$, and define a second map $\phi_2 : \langle A \rangle \to \mathbb{F}_2^8$ where

$$\phi_2([b_1b_2 \quad b_3b_4 \quad \cdots \quad b_{15}b_{16}]) = [\phi_1(b_1b_2) \quad \phi_1(b_3b_4) \quad \cdots \quad \phi_1(b_{15}b_{16})] \cdot$$

Then the image of $\langle A \rangle$ under ϕ_2 is the even space of $\mathbb{F}_2^8$.

$$[\phi_2(A)] = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & & 1 & & & & & \\ 1 & & & 1 & & & & \\ 1 & & & & 1 & & & \\ 1 & & & & & 1 & & \\ 1 & & & & & & 1 & \\ 1 & & & & & & & 1 \end{bmatrix}$$

Let G_{C_1} be the generator matrix of C_1. Take the top four rows of G_{C_1} (denoted G_H) to be generated in $\langle A \rangle$ as the unique optimal code in $\phi_2(\langle A \rangle)$ with minimum distance 4, the extended Hamming $[8, 4, 4]$ code (e_8). Hence by adding the vector y to the generator matrix it is clear to see that y is independent from the rows of G_H and C_1 has minimum distance 8.

$$[G_{C_1}] = \left[\begin{array}{c} G_H \\ \hline y \end{array}\right] = \left[\begin{array}{cccccccc} 11 & 11 & 11 & 11 & & & & \\ & & 11 & 11 & 11 & 11 & & \\ & & & & 11 & 11 & 11 & 11 \\ 11 & & 11 & & 11 & & 11 & \\ \hline 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right]$$

Therefore the maximal dimension of C_1 is 5. The uniqueness of C_1 is due to the uniqueness of the [16,5,8] Reed-Muller code $R(1, 4)$ (p. 81 of [18]). □

Lemma 3.3. *$2e_8$ contains a unique optimal $[16, 5, 8]$ code which is a maximal subcode with respect to minimum distance 8. Also, 5 is the maximum dimension corresponding to $d' = 8$.*

Proof. To construct C_1, a maximal subcode of $2e_8$ with minimum distance 8 similar arguments to above lemma are used. As above, the maximum dimension possible for C_1 is $k = 5$ as there exists no $[16, 6, 8]$ code by [15]. The maximum dimension

possible for C_1 is $k = 5$. Consider the following form of the generator matrix of $2e_8$ with row vectors labeled as $\{a_1, a_2, a_3, a_4, a_5, a_6, z_1, z_2\}$ and where A is a matrix with rows $\{a_1, a_2, a_3, a_4, a_5, a_6\}$:

$$\left[\begin{array}{cccccccc} 11 & 11 & & & & & & \\ & 11 & 11 & & & & & \\ & & 11 & 11 & & & & \\ & & & & 11 & 11 & & \\ & & & & & 11 & 11 & \\ & & & & & & 11 & 11 \\ \hline 10 & 10 & 10 & 10 & & & & \\ & & & & 10 & 10 & 10 & 10 \end{array}\right] = \left[\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ \hline z_1 \\ z_2 \end{array}\right] = \left[\begin{array}{c} A \\ \hline z_1 \\ z_2 \end{array}\right].$$

By applying the shrinking maps defined in the above proof the following length 8 generator matrix is obtained from the matrix A:

$$[\phi_2(A)] = \left[\begin{array}{cccccccc} 1 & 1 & & & & & & \\ & 1 & 1 & & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & 1 & 1 & \\ & & & & & & 1 & 1 \end{array}\right]$$

This code contains an [8,4,4] code equivalent to the unique extended Hamming code. The pre-image of this [8,4,4] code is a [16,4,8] subcode of A. Applying a similar argument to the above proof, by appending the weight eight vector $z_1 + z_2$ (or $z_1 + z_2 + b_i$ for some $b_i \in \langle A \rangle$) to the generator matrix of the $[16, 4, 8]$ code, a $[16, 5, 8]$ code C_1 is obtained. The uniqueness of C_1 follows from the uniqueness of the [16,5,8] Reed-Muller code $R(1, 4)$. □

The information from this section is summarized in the following table.

TABLE 3.1

Maximum Dimension Subcodes of All Type II codes of $n = 16$

Codes	max. dim. with $d = 8$	max. dim. with $d = 16$
d_{16}	5	1
$2e_8$	5	1

3.3 Classification of Optimal Subcodes and ODP for Type II Codes of Length 24

Consider $n = 24$. There are exactly nine Type II self-dual codes of length 24. These are denoted by $A24(2d_{12})$, $B24(d_{10} + 2e_7)$, $C24(3d_8)$, $D24(4d_6)$, $E24(d_{24})$, $F24(6d_4)$, $G24(g_{24})$, $d_{16} + e_8$, and $3e_8$ in the notations of [9], [34]. The first seven codes are indecomposable and the rest are decomposable. Note that $G24(g_{24})$ represents the binary Golay $[24, 12, 8]$ code.

Pollara, et. al. [35] constructed the first $[24, 5, 12]$ subcode $C_{24}^{5,12}$ of g_{24}, improving a previously known $[24, 5, 8]$ subcode. Note that $C_{24}^{5,12}$ is unique [43], has only two non-zero weights 12 and 16, and has a $[24, 2, 16]$ subcode $C_{24}^{2,16}$. As $C_{24}^{2,16}$ satisfies the Griesmer bound, it has a generator matrix of which each row has weight 16 [43], [18]. Hence it is easy to see that $C_{24}^{2,16}$ is unique.

Using this information, Luo, et. al. [25] have determined

$$\begin{aligned} \mathrm{ODP}^{dic}[g_{24}] &= [8, 8, 8, 8, 8, 8, 8, 8, 12, 12, 12, 16, 16] \\ \mathrm{ODP}^{inv}[g_{24}] &= [8, 8, 8, 8, 8, 8, 8, 8, 12, 12, 12, 24]. \end{aligned}$$

However, less is known of the subcodes of the other Type II self-dual codes of length 24. We have checked that the unique $[24, 5, 12]$ code is contained in any of the nine Type II codes of length 24.

Using (Subcodes) Chain Algorithm I we obtain inequivalent maximal $[24, k', 8]$ subcodes of each Type II code of length 24 (with minimum distance 4). Then applying (Supercodes) Chain Algorithm II to the unique $[24, 5, 12]$ code for each Type II code of length 24 (with minimum distance 4) we obtain a $[24, k', 8]$ code equivalent to one of the maximal subcodes. Therefore we determine the ODP in the dictionary order of the Type II $[24, 12, 4]$ codes as follows.

Theorem 3.4.

$$\begin{aligned}
ODP^{dic}[2d_{12}] &= [4,4,4,8,8,8,8,12,12,12,16,16] \\
ODP^{dic}[d_{10}+2e_7] &= [4,4,4,8,8,8,8,12,12,12,16,16] \\
ODP^{dic}[3d_8] &= [4,4,8,8,8,8,8,12,12,12,16,16] \\
ODP^{dic}[4d_6] &= [4,4,8,8,8,8,8,12,12,12,16,16] \\
ODP^{dic}[d_{24}] &= [4,4,4,4,8,8,8,12,12,12,16,16] \\
ODP^{dic}[6d_4] &= [4,8,8,8,8,8,8,12,12,12,16,16] \\
ODP^{dic}[d_{16}+e_8] &= [4,4,4,8,8,8,8,12,12,12,16,16] \\
ODP^{dic}[3e_8] &= [4,4,4,8,8,8,8,12,12,12,16,16]
\end{aligned}$$

For each Type II $[24, 12, 4]$ code we apply (Subcodes) Chain Algorithm I to the maximal $[24, k', 8]$ subcodes (containing the all one vector) to obtain a $[24, 4, 12]$ subcode (containing the all one vector). Therefore we may determine the ODP in the inverse dictionary order of the Type II $[24, 12, 4]$ codes as follows.

Theorem 3.5.

$$
\begin{aligned}
ODP^{inv}[2d_{12}] &= [4,4,4,8,8,8,8,8,12,12,12,24] \\
ODP^{inv}[d_{10}+2e_7] &= [4,4,4,8,8,8,8,8,12,12,12,24] \\
ODP^{inv}[3d_8] &= [4,4,8,8,8,8,8,8,12,12,12,24] \\
ODP^{inv}[4d_6] &= [4,4,8,8,8,8,8,8,12,12,12,24] \\
ODP^{inv}[d_{24}] &= [4,4,4,4,8,8,8,8,12,12,12,24] \\
ODP^{inv}[6d_4] &= [4,8,8,8,8,8,8,8,12,12,12,24] \\
ODP^{inv}[d_{16}+e_8] &= [4,4,4,8,8,8,8,8,12,12,12,24] \\
ODP^{inv}[3e_8] &= [4,4,4,8,8,8,8,8,12,12,12,24]
\end{aligned}
$$

Table 3.2 gives the maximum dimension with respect to minimum distance d for the Type II length 24 codes.

Corollary 3.6. *For each Type II length 24 code, there are maximum dimension subcodes with respect to* $d = 8, 12, 16, 24$ *(except* 20*) that are involved in the subcode chain for the ODP in dictionary order or the inverse order. Furthermore, each Type II length 24 code contains dimension optimal (and minimum distance optimal) subcodes with parameters* $[24,5,12], [24,2,16], [24,1,24]$.

TABLE 3.2

Maximum Dimension Subcodes of All Type II codes of $n = 24$

Codes	max. dim. with $d = 8$	max. dim. with $d = 12$
$2d_{12}$	9	5
$d_{10} + 2e_7$	9	5
$3d_8$	10	5
$4d_6$	10	5
d_{24}	8	5
$6d_4$	11	5
$d_{16} + e_8$	9	5
$3e_8$	9	5
g_{24}	12	5

3.4 Classification of Optimal Subcodes and ODP for Extremal Type II Codes of Length 32

As there are 85 Type II self-dual codes of length 32, we focus on extremal Type II self-dual $[32, 16, 8]$ codes. There are exactly five Type II self-dual $[32, 16, 8]$ codes, denoted by $C81$ (or q_{32}), $C82$ (or r_{32}, $R(2, 5)$), $C83$ (or $2g_{16}$), $C84$ (or $8f_4$), $C85$ ($16f_2$) in the notation of [9], [10]. Using symplectic geometric approach, Maks and Simonis [30] show that the second order Reed-Muller code r_{32} contains exactly two inequivalent $[32, 11, 12]$ codes, each of which further contains the first order Reed-Muller $[32, 6, 16]$ code $R(1, 5)$. Note that any $[32, 6, 16]$ code is equivalent to $R(1, 5)$. Furthermore, Jaffe [20] proved using his language `Split` that there exist exactly two $[32, 11, 12]$ codes. These subcodes have optimal dimensions for each minimum distance. Hence Chen and Han Vinck [8] have determined the ODP in the dictionary order for r_{32} as follows:

$$\mathrm{ODP}[r_{32}] = [8, 8, 8, 8, 8, 12, 12, 12, 12, 12, 16, 16, 16, 16, 16, 32].$$

On the other hand, little was known of the subcodes of the other four extremal Type II $[32, 16, 8]$ codes. We show that they also have the same optimum distance profiles as r_{32} does.

Using (Supercodes) Chain Algorithm II with $C_{k',d'} = \{R(1, 5)\}$, we independently construct two inequivalent $[32, 11, 12]$ codes in r_{32} containing $R(1, 5)$, denoted by RC_1 and RC_2. We note that $\dim(RC_1 \cap RC_2) = 10$. Using (Supercodes) Chain Algorithm II, we have checked that each of RC_1 and RC_2 is a subcode of any of the five Type II $[32, 16, 8]$ codes. We denote the five codes based on RC_1 (RC_2, respectively) by $C81^1, \ldots, C85^1$ ($C81^2, \ldots, C85^2$, respectively).

Hence we obtain:

Theorem 3.7. *Each code C of the five Type II $[32, 16, 8]$ codes has*

$$ODP[C] = [8, 8, 8, 8, 8, 12, 12, 12, 12, 12, 16, 16, 16, 16, 16, 32].$$

One generator matrix for each Type II $[32, 16, 8]$ code with respect to the ODP in the dictionary order is given at the end of this section.

$$RC_1 = \left[\begin{array}{c} 11111111111111111111111111111111 \\ \hline 00000000000000001111111111111111 \\ 00000000111111110000000011111111 \\ 00001111000011110000111100001111 \\ 00110011001100110011001100110011 \\ 01010101010101010101010101010101 \\ \hline 10000001000101110100110100100100 \\ 01000001000101000010011110001101 \\ 00100001010001110111010000010010 \\ 00001001000010010101110010100011 \\ 00100001000100100001110111010001 \end{array}\right]$$

$$RC_2 = \left[\begin{array}{c} 11111111111111111111111111111111 \\ \hline 00000000000000001111111111111111 \\ 00000000111111110000000011111111 \\ 00001111000011110000111100001111 \\ 00110011001100110011001100110011 \\ 01010101010101010101010101010101 \\ \hline 10000001000101110100110100100100 \\ 01000001000101000010011110001101 \\ 00100001010001110111010000010010 \\ 00001001000010010101110010100011 \\ 00100001000100100111101101001000 \end{array}\right]$$

Corollary 3.8. *For each extremal Type II length 32 code, there are maximum dimension subcodes with respect to* $d = 12, 16, 32$ *that are involved in the subcode*

chain for the ODP in dictionary order or the inverse order. Furthermore, each extremal Type II length 32 code contains dimension optimal (and minimum distance optimal) subcodes with parameters $[32, 11, 12], [32, 6, 16], [32, 1, 32]$.

$$C81^1 = \left[\begin{array}{c} RC_1 \\ \hline 10000000000000010001011000001110 \\ 01000000000000100001010100110001 \\ 00100000000000100011000101001001 \\ 00010000000000010001000011001101 \\ 00001000000100000010010100110010 \end{array}\right]$$

$$C82^1 = \left[\begin{array}{c} RC_1 \\ \hline 10000000000000010110111000011111 \\ 01000000000000010010000000111011 \\ 00100000000000010110110111010101 \\ 00010000000000010000111001000101 \\ 00001000000000010001101000010011 \end{array}\right]$$

$$C83^1 = \left[\begin{array}{c} RC_1 \\ \hline 10000001000100010011010101100110 \\ 01000001000100100011011001011001 \\ 00100001000100100001001000100001 \\ 00010001000100010010001000100010 \\ 00001001000000000001011100100010 \end{array}\right]$$

$$C84^1 = \left[\begin{array}{c} RC_1 \\ \hline 1000000000000001000101100000 1110 \\ 0100000000000010000101010011 0001 \\ 0010000000000010001100010100 1001 \\ 0001000000000001000000011011 0101 \\ 0000100000010000001101000100 1010 \end{array}\right]$$

$$C85^1 = \left[\begin{array}{c} RC_1 \\ \hline 1000000000000001000101100000 1110 \\ 0100000000000001010101111101 1010 \\ 0010000000000001000110100011 0100 \\ 0001000000000001000000011011 0101 \\ 0000100000000001000001001001 1011 \end{array}\right]$$

$$C81^2 = \left[\begin{array}{c} RC_2 \\ \hline 1000000000010000001001010010 1100 \\ 0100000000010011001001101110 1100 \\ 0010000000010011000000101001 0100 \\ 0001000000010000001000111110 1111 \\ 0000100000000001000101101110 1111 \end{array}\right]$$

$$C82^2 = \left[\begin{array}{c} RC_2 \\ \hline 1000001000101110001011101111 0110 \\ 0100001000101000001010001000 0001 \\ 0010000100010010000100100010 0001 \\ 0001000100010001000100010001 0001 \\ 0000100100000110000001100000 1001 \end{array}\right]$$

$$C83^2 = \left[\begin{array}{c} RC_2 \\ \hline 10000001000100010011010101100110 \\ 01000001000100100011011001011001 \\ 00100001000100100001001000100001 \\ 00010001000100010010001000100010 \\ 00001001000000000001011100100010 \end{array}\right]$$

$$C84^2 = \left[\begin{array}{c} RC_2 \\ \hline 10000000000100000010010100101100 \\ 01000000000100110010011011101100 \\ 00100000000100110000001010010100 \\ 00010000000100000011001001101000 \\ 00001000000000010000011101101000 \end{array}\right]$$

$$C85^2 = \left[\begin{array}{c} RC_2 \\ \hline 10000000000100000010010100101100 \\ 01000000000100000110011111110100 \\ 00100000000100000100001110001100 \\ 00010000000100000010001111101111 \\ 00001000000000010001011011101111 \end{array}\right]$$

3.5 Results Towards the ODP for the Unique $[48, 24, 12]$ Code

The extended QR code q_{48} is a unique $[48, 24, 12]$ self-dual code. Using Random (Subcodes) Algorithm I, we find that for $d' = 16$, there is a maximal $[48, 14, 16]$ subcode of q_{48}. The best known minimum distance optimal $[48, 14]$ code has $d = 16$. (Note that 17 is the upper bound.) One code is given in Magma. We have checked that our code is not equivalent to this code. Similarly, for $d' = 20$, there is a maximal $[48, 9, 20]$ subcode of q_{48}. This is minimum distance optimal. One $[48, 9, 20]$ code is given in Magma. We have checked that our $[48, 9, 20]$ code is not equivalent to this code. For $d' = 24$, there is a maximal $[48, 6, 24]$ subcode of q_{48}, which is in fact a unique code by [43]. This is minimum distance optimal. One code is given in Magma. We have checked that our code is equivalent to this code.

With respect to the inverse dictionary order we have examined some self-complementary subcodes of q_{48}. There is a $[48, 5, 24]$ self-complementary subcode (note that $k = 5$ is the maximum dimension of a $[48, k, 24]$ self-complementary subcode since the unique $[48, 6, 24]$ code does not contain the all-one vector). There is a maximal $[48, 9, 20]$ self-complementary subcode containing the $[48, 5, 24]$ code (note that $k = 10$ is the maximum dimension of a $[48, k, 20]$ self-complementary subcode). In what follows, we classify all possible weight distributions of a supposed $[48, 10, 20]$ self-complementary subcode of q_{48}.

Lemma 3.9. *If C is a self-complementary [48,10,20] subcode of q_{48}, then the non-zero codewords of C have weights 20,24,28,48.*

Proof. Suppose to the contrary that C has non-zero weights 20,28,48. Then clearly $A_{20} := 2^9 - 1$. Using the MacWilliams Identities (Lemma 1.1) we obtain the equation $2256 + 16A_{20} = 2^{10}A_2^{\perp}$. Hence $A_2^{\perp} = \frac{163}{16}$, a contradiction. □

Lemma 3.10. *If C is a self-complementary [48,10,20] subcode of q_{48}, then $d^{\perp}(C) \neq$*

2.

Proof. Suppose to the contrary that $d^{\perp}(C) = 2$. Shortening C on a minimum weight codeword x_2 of $C^{\perp}$ yields a [46,9,20] code C_{46} with possible non-zero weights 20,24,28 by Lemma 1.2 (here we switched the role of C and $C^{\perp}$).

Define the following matrices:

$$\begin{aligned} B &= [A_0^{\perp}(C_{46}) \quad A_1^{\perp}(C_{46}) \quad A_2^{\perp}(C_{46}) \quad A_3^{\perp}(C_{46})]^T, \\ A &= [A_0(C_{46}) \quad A_{20}(C_{46}) \quad A_{24}(C_{46}) \quad A_{28}(C_{46})]^T. \end{aligned}$$

Then the MacWilliams Identities yield the matrix equation $2^9 B = PA$, where

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 46 & 6 & -2 & -10 \\ 1035 & -5 & -21 & 27 \\ 15180 & -100 & 44 & 60 \end{bmatrix}.$$

By Grassl's table [15] there (respectively) does not exist a [45,9,20] linear code and there does not exist a [44,8,20] linear code, therefore respectively we have $A_1^{\perp}(C_{46}) = 0$ and $A_2^{\perp}(C_{46}) = 0$. Combined with the fact that $A_0^{\perp}(C_{46}) = 1$ the above matrix equation yields a unique solution of:

$$A = [1 \quad 243 \quad 147 \quad 121]^T. \tag{3.1}$$

The possible weight distribution of C_{46} and $C_{46}^{\perp}$ follows from (3.1). In particular, $d(C_{46}^{\perp}) = 3$ which by shortening C_{46} on a minimum weight codeword of $C_{46}^{\perp}$ using Lemma 1.2 implies the existence of a $[43, 7, 20]$ code with non-zero weights 20,24,28. This is a contradiction to the classification of [43,7,20] due to Bouyuklieva and Jaffe [5]. □

Lemma 3.11. *If C is a self-complementary [48,10,20] subcode of q_{48}, then there is one possible weight distribution of C:*

$$A_0 = 1 \quad A_{20} = 348 \quad A_{24} = 326 \quad A_{28} = 348 \quad A_{48} = 1.$$

Proof. Define the following matrices:

$$\begin{aligned} B &= [A_0^\perp \quad A_2^\perp \quad A_4^\perp]^T, \\ A &= [A_0 \quad A_{20} \quad A_{24}]^T. \end{aligned}$$

Then the MacWilliams Identities along with the fact that C is self-complementary yield the matrix equation $2^{10}B = PA$, where

$$P = \begin{bmatrix} 2 & 2 & 1 \\ 2256 & 16 & -24 \\ 389160 & -600 & 276 \end{bmatrix}.$$

By the previous lemma $A_2^\perp = 0$, combined with the fact that $A_0 = A_0^\perp = 1$ the above matrix equation yields a unique solution of:

$$A = [1 \quad 348 \quad 326]^T.$$

□

Lemma 3.12. *There does not exist a self-complementary [48,k,16] subcode C of q_{48} for $k \geq 17$.*

Proof. Suppose a [48,17,16] self-complementary subcode C exists. The possible non-zero weights of C are 16,20,24,28,32,48. Define the following matrices:

$$\begin{aligned} B &= [A_0^\perp \quad A_2^\perp \quad A_4^\perp \quad A_6^\perp]^T, \\ A &= [A_0 \quad A_{16} \quad A_{20} \quad A_{24}]^T. \end{aligned}$$

Then the MacWilliams Identities along with the fact that C is self-complementary yield the matrix equation $2^{17}B = PA$, where

$$P = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2256 & 208 & 16 & -24 \\ 389160 & 40 & -600 & 276 \\ 24543024 & -14544 & 5616 & -2024 \end{bmatrix}.$$

Isolating the matrix A yields the matrix equation $2^{17}P^{-1}B = A$ where

$$2^{17}P^{-1} = \begin{bmatrix} 17/14 & 65/224 & 3/56 & 1/224 \\ 9729/2 & 17457/32 & 211/8 & -15/32 \\ 207552/7 & -1012/7 & -752/7 & 12/7 \\ 62040 & -1605/2 & 162 & -5/2 \end{bmatrix}.$$

The first row of $2^{17}P^{-1}$ implies

$$\frac{65}{224}A_2^{\perp} + \frac{3}{56}A_4^{\perp} + \frac{1}{224}A_6^{\perp} = -\frac{3}{14},$$

which is impossible as $A_i^{\perp} \geq 0$ for all i. Hence no such code C can exist. □

The previous lemmas and example from this section yield the following theorem towards the inverse dictionary order ODP for q_{48}.

Theorem 3.13.

$$ODP^{inv}[q_{48}] = [12, 12, 12, 12, 12, 12, 12, 12, a_1, a_2,$$

$$a_3, a_4, a_5, a_6, b, 20, 20, 20, 20, 24, 24, 24, 24, 48]$$

where $a_i \in \{12, 16\}$ *and* $b \in \{12, 16, 20\}$.

Proof. Since q_{48} contains the all-one vector, the repetition code $[48, 1, 48]$ must be the one dimensional subcode first appearing in the subcode chain. By [43] there is a unique $[48, 6, 24]$ code with non-zero weights 24, 32; since this code does not contain the all-one vector it cannot be involved in the inverse dictionary order subcode chain. Hence $k \leq 5$ for a $[48, k, 24]$ code involved in the subcode chain. Applying Random (Supercodes) Algorithm II to the $[48, 1, 48]$ subcode of q_{48} we obtained a subcode chain involving a $[48, 5, 24]$ code contained in a $[48, 9, 20]$ subcode of q_{48}. Therefore $\mathrm{ODP}^{inv}[q_{48}]_i = 24$ for $2 \leq i \leq 5$, and $\mathrm{ODP}^{inv}[q_{48}]_j = 20$ for $6 \leq i \leq 9$. The maximum dimension for a $[48, k, 20]$ code is $k = 10$ by Grassl's table [15], hence $\mathrm{ODP}^{inv}[q_{48}]_{10} = b$ for $b \in \{12, 16, 20\}$ and also $\mathrm{ODP}^{inv}[q_{48}]_j = a_i$ for $11 \leq j \leq 16$ and $a_i \in \{12, 16\}$. Finally, $\mathrm{ODP}^{inv}[q_{48}]_i = 12$ for $17 \leq i \leq 24$ by Lemma 3.12. □

Lemma 3.14. *There does not exist a [48,k,16] subcode C of q_{48} for $k \geq 17$.*

Proof. Suppose a [48,17,16] subcode C of q_{48} exists. Since the self-complementary case is already considered in Lemma 3.12, we only need to examine the case where the maximum weight in C is 36 since the non-zero weights in q_{48} are $12, 16, 20, 24, 28, 32, 36, 48$. Hence the possible non-zero weights of C are 16, 20, 24, 28, 32, 36. Define the following matrices:

$$\begin{aligned} B &= [A_0^\perp \quad A_1^\perp \quad A_2^\perp \quad A_3^\perp \quad A_4^\perp \quad A_5^\perp \quad A_6^\perp]^T, \\ A &= [A_0 \quad A_{16} \quad A_{20} \quad A_{24} \quad A_{28} \quad A_{32} \quad A_{36}]^T. \end{aligned}$$

Then the MacWilliams Identities yield the matrix equation $2^{17}B = PA$, where

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 48 & 16 & 8 & 0 & -8 & -16 & -24 \\ 1128 & 104 & 8 & -24 & 8 & 104 & 264 \\ 17296 & 304 & -104 & 0 & 104 & -304 & -1736 \\ 194580 & 20 & -300 & 276 & -300 & 20 & 7380 \\ 1712304 & -2672 & 456 & 0 & -456 & 2672 & -19800 \\ 12271512 & -7272 & 2808 & -2024 & 2808 & -7272 & 25080 \end{bmatrix}.$$

Isolating the matrix A yields the matrix equation $2^{17}P^{-1}B = A$ where

$$2^{17}P^{-1} = \begin{bmatrix} 34/21 & 17/21 & 65/168 & 1/6 & 1/14 & 1/42 & 1/168 \\ 4788 & 1698 & 2109/4 & 135 & 23 & 1 & -3/4 \\ 30000 & 4592 & -61 & -312 & -92 & -8 & 3 \\ 61360 & 680 & -965 & 140 & 132 & 20 & -5 \\ 212448/7 & -39488/7 & 158/7 & 96 & -536/7 & -160/7 & 30/7 \\ 4482 & -1239 & 3633/8 & -81/2 & 19/2 & 25/2 & -15/8 \\ 272/3 & -272/3 & 65/3 & -56/3 & 4 & -8/3 & 1/3 \end{bmatrix}.$$

The first row of $2^{17}P^{-1}$ implies

$$\frac{17}{21}A_1^\perp + \frac{65}{168}A_2^\perp + \frac{1}{6}A_3^\perp + \frac{1}{14}A_4^\perp + \frac{1}{42}A_5^\perp + \frac{1}{168}A_6^\perp = -\frac{13}{21},$$

which is impossible as $A_i^{\perp} \geq 0$ for all i. Hence no such code C can exist.

□

Theorem 3.15.

$$ODP^{dic}[q_{48}] = [12, 12, 12, 12, 12, 12, 12, 12, a_1, a_2,$$

$$16, 16, 16, 16, b_1, b_2, b_3, b_3, c_1, c_2, c_3, c_4, d, e]$$

where $a_i \in \{12, 16\}$, $b_k \in \{16, 20\}$, $c_l \in \{16, 20, 24\}$, $d \in \{16, 20, 24, 28, 32\}$, *and* $e \in \{20, 24, 28, 32, 36, 48\}$.

Proof. Note that the non-zero weights in q_{48} are 12,16,20,24,28,32,36,48. Therefore by Grassl's table [15] we may deduce the following:

$\text{ODP}^{dic}[q_{48}]_j = 12$ for $17 \leq j \leq 24$, by the previous lemma.

$\text{ODP}^{dic}[q_{48}]_j = a_i$ for $15 \leq j \leq 16$ and $a_i \in \{12, 16\}$.

$\text{ODP}^{dic}[q_{48}]_j = 16$ for $11 \leq j \leq 14$, because as mentioned at the beginning of this section, there exists a maximal $[48, 14, 16]$ subcode of q_{48}.

$\text{ODP}^{dic}[q_{48}]_j = b_i$ for $7 \leq j \leq 10$ and $b_i \in \{16, 20\}$.

$\text{ODP}^{dic}[q_{48}]_j = c_i$ for $3 \leq j \leq 6$ and $c_i \in \{16, 20, 24\}$.

$\text{ODP}^{dic}[q_{48}]_2 = d$ for $d \in \{16, 20, 24, 28, 32\}$.

$\text{ODP}^{dic}[q_{48}]_1 = e$ for $e \in \{20, 24, 28, 32, 36, 48\}$.

Note that 16 is not present for values of e because if so then the $[48, 14, 16]$ code involved in the subcode chain would have to be a constant weight code. There does not exist a constant weight code (with weight 16) of dimension greater than 5 by the following reasoning.

Suppose there exists a $[48, k, 16]$ constant weight code. Define the following matrices:

$$\begin{aligned} B &= [A_0^\perp \quad A_1^\perp]^T, \\ A &= [A_0 \quad A_{16}]^T. \end{aligned}$$

Then the MacWilliams Identities yield the matrix equation $2^k B = PA$, where

$$P = \begin{bmatrix} 1 & 1 \\ 48 & 16 \end{bmatrix}.$$

Since $A_0^\perp = 1 = A_0$, then the matrix equation yields the following system:

$$\begin{aligned} 2^k &= 1 + A_{16} \\ 2^k A_1^\perp &= 48 + 16A_{16}. \end{aligned}$$

Solving for A_{16} in the first equation and substituting into the second equation yields:

$$2^k A_1^\perp = 48 + 16(2^k - 1).$$

Solving for $A_1^\perp$ we obtain:

$$A_1^\perp = 2^{5-k} + 16.$$

And finally since $A_1^\perp$ is an integer, then $k \leq 5$. □

From the previous ODPs that have been found for Type II codes, dimension optimal subcodes are involved in subcode chains. Therefore we have the following:

Conjecture: A $[48, 6, 24]$ code is involved in a subcode chain for the ODP in dictionary order.

Thus we have the following corollary.

Corollary 3.16. *If a* $[48, 6, 24]$ *code is involved in a subcode chain for the ODP in dictionary order, then*

$$ODP^{dic}[q_{48}] = [12, 12, 12, 12, 12, 12, 12, 12, a_1, a_2,$$

$$16, 16, 16, 16, b_1, b_2, b_3, b_3, 24, 24, 24, 24, 32, 32]$$

where $a_i \in \{12, 16\}$ *and* $b_j \in \{16, 20\}$.

We were able to find a doubly-even self-complementary $[48, 16, 16]$ code with generator matrix $G_{[48,16,16]}$. Such a code was previously not known to exist. Only one singly-even self-complementary $[48, 16, 16]$ code was found by A. Kohnert [24].

The dual code has minimum distance $d = 4$. The generator matrix for this doubly-even self-complementary $[48, 16, 16]$ code is the following:

$$G_{[48,16,16]} = \begin{bmatrix}
100100000000001000110001001011100011100100010100 \\
010100000000000001111000001001001100110010110100 \\
001100000000001000101111000010101001010111011110 \\
000010000000001000101000111100101110101111000111 \\
000001000000000000011110010000101010101010110101 \\
000000100000001001001110101100000101100100001101 \\
000000010000000001111111110010100000011110000000 \\
000000001000000001100001101101000111111110000000 \\
000000000100000001100111001111001001101111111100 \\
000000000010000001001101100111100011001111111010 \\
000000000001001001110011111110000010100000101000 \\
000000000000101000010100011110000011000111001110 \\
000000000000011001111001100000000110000110011110 \\
000000000000000100001101011001101101011010011000 \\
000000000000000010010010111001101101000011100110 \\
000000000000000000000000000000011111111111111110
\end{bmatrix}$$

3.6 Examination of the Length 72

Note that q_{72} (the extended quadratic residue code of length 72) is a Type II $[72, 36, 12]$ code. Due to the complexity, we use Random (Subcodes) Algorithm I. For $d' = 16$, there is a maximal $[72, 29, 16]$ subcode of q_{72} with $A_{16} = 2160$. The best known minimum distance optimal $[72, 29]$ code has $d = 16$ (and at most $d \leq 21$) with $A_{16} = 28417$, given in Magma. Hence our code is not equivalent to this code. For $d' = 20$, there is a maximal $[72, 23, 20]$ subcode with $A_{20} = 3046$. The best known minimum distance optimal $[72, 23]$ code has $d = 20$ (and at most ≤ 24) with $A_{20} = 7120$ given in Magma. Hence our code is not equivalent to this

code.

We start from a best known linear $[72, 31, 20]$ code, given in Magma. Let C_1 be this code and let $d' = 16 < d = 20$. Using Random (Supercodes) Algorithm II, we have constructed in a few seconds a *doubly-even self-orthogonal* $[72, 35, 16]$ *code* C' containing C_1 with $A_{16} = 129972$. It is known from Magma that there is a best known minimum distance code with parameters $[72, 35, 16]$. This is a doubly-even self-orthogonal code with $A_{16} = 136116$. Hence our code *is not equivalent* to the known code. We do not know how many doubly-even self-orthogonal $[72, 35, 16]$ codes exist.

CHAPTER 4
OPTIMUM DISTANCE PROFILES OF NEAR EXTREMAL FORMALLY SELF-DUAL CODES

4.1 Optimum Distance Profiles for Near Optimal and Optimal FSD Codes of Length 16

In this section, we determine the optimum distance profiles of some interesting formally self-dual codes of lengths 16–22. In [3], Betsumiya and Harada classified the formally self-dual even codes of length 16. We examined the optimum distance profiles of all near optimal formally self-dual even [16,8,4] codes; the results may be found in the tables at the end of this chapter. The tables arrange the codes first by their weight distribution in column 1, and second by their ODPs in column 2. Columns 3-4 total the number of codes with the given Weight Distribution and ODP which are respectively Self-Dual, Iso-Dual, and Formally Self-Dual.

In another paper, Betsumiya and Harada have shown that there is a unique optimal [16,8,5] code, and in fact this code is formally self-dual [2]. We examined the optimum distance profiles of this code and found the following:

Theorem 4.1. *If C is the unique $[16, 8, 5]$ code, then*

$$ODP^{dic}[C] = [5, 6, 6, 6, 8, 8, 8, 10],$$

and

$$ODP^{inv}[C] = [5, 5, 5, 6, 7, 8, 10, 12]$$

with generator matrix for each profile respectively:

$$G(C) = \left[\begin{array}{c} 0000111111011101 \\ \hline 0011001110101010 \\ 0101011001101100 \\ 1001010100011011 \\ \hline 0101010111111111 \\ 1000010001111101 \\ 1000000111001001 \\ \hline 1000000010111000 \end{array}\right], \quad G(C) = \left[\begin{array}{c} 0101010111111111 \\ \hline 1011101110011001 \\ \hline 1001110011100001 \\ \hline 1001001001001101 \\ \hline 1000011010011111 \\ \hline 1000010001111101 \\ 0100000001011100 \\ 1000000010111000 \end{array}\right].$$

Proof. (Subcodes) Chain Algorithm I can be applied recursively to give the ODP in dictionary order for C. The weight distribution of C is

$$A_0 = 1 \quad A_5 = 24 \quad A_6 = 44 \quad A_7 = 40 \quad A_8 = 45$$

$$A_9 = 40 \quad A_{10} = 28 \quad A_{11} = 24 \quad A_{12} = 10.$$

To find the ODP in inverse dictionary order, one must consider some cases. Clearly, $\mathrm{ODP}^{inv}[C]_1 = 12$. By easy case analysis of length 16 binary vectors the sum of a weight 12 vector and a weight 11 vector has weight less than or equal to 9. Therefore $\mathrm{ODP}^{inv}[C]_2 \leq 10$. By inspection of the weight 12 and weight 10 vectors in C it is clear that there exists a subcode with weight distribution $A_0 = 1 \quad A_{10} = 2 \quad A_{12} = 1,$ hence $\mathrm{ODP}^{inv}[C]_2 = 10$ Let $C_{[16,2,10]}$ be any $[16, 2, 10]$ code with weight distribution $A_0 = 1 \quad A_{10} = 2 \quad A_{12} = 1,$ then there is only one code with this weight distribution up to equivalence (since fixing the support of any weight 12 vector c_1 forces $|c_1 \cap c_2| = 6$ for any weight 10 vector c_2 in $C_{[16,2,10]}$. By applying (Supercodes) Chain Algorithm II to $C_{[16,2,10]}$ there is exactly one supercode of $C_{[16,2,10]}$ with maximal minimum distance; this is a $[16, 3, 8]$ code. Hence $\mathrm{ODP}^{inv}[C]_3 \leq 8$. In a similar manner we obtain $\mathrm{ODP}^{inv}[C]_4 \leq 7$ as two inequivalent $[16, 4, 7]$ codes were generated from the $[16, 3, 8]$ code. Continuing to apply

the algorithm to increase the dimensions of the subcode chains containing either of the $[16, 4, 7]$ codes we notice that for dimension 6 there are exactly three inequivalent $[16, 6, 6]$ codes in the subcode chain; however, continuing with the algorithm none of these three $[16, 6, 6]$ codes are contained in a $[16, 8, 5]$ code. Therefore, we must proceed instead from all inequivalent $[16, 5, 6]$ codes containing either of the $[16, 4, 7]$ codes we generated (there are 12 such codes). Proceeding with the algorithm we obtain a $[16, 8, 5]$ code in the final step (this code is equivalent to C as C is unique). Therefore we may conclude $\mathrm{ODP}^{inv}[C]_3 = 8$,$\mathrm{ODP}^{inv}[C]_4 = 7$, $\mathrm{ODP}^{inv}[C]_5 = 6$, and $\mathrm{ODP}^{inv}[C]_i = 5$ for $6 \leq i \leq 8$. □

Corollary 4.2. *The* $[16, 8, 5]$ *code contains dimension optimal and minimum distance optimal subcodes with parameters* $[16, 7, 6]$ *and* $[16, 2, 10]$ *(these are maximum dimension subcodes with respect to* $d = 6$ *and* $d = 10$*).*

4.2 Optimum Distance Profiles for Near Optimal and Optimal FSD Codes of Length 18-22

In 1992, Simonis showed that there is a unique [18,9,6] code [40]; it turns out that this code is also formally self-dual even. We examined the optimum distance profiles of this code and found the following:

Theorem 4.3. *If* C *is the unique* $[18, 9, 6]$ *code, then*

$$ODP^{dic}[C] = [6, 6, 6, 6, 8, 8, 10, 10, 12],$$

and

$$ODP^{inv}[C] = [6, 6, 6, 6, 8, 8, 8, 8, 18]$$

with generator matrix for each profile respectively:

$$G(C) = \left[\begin{array}{c} 111111001011101001 \\ \hline 111010001100010111 \\ 100011111111000100 \\ \hline 100011100010011001 \\ 010010000111110001 \\ \hline 100011001011010010 \\ 100011000101111101 \\ 100010000100011010 \\ 100000000111010111 \end{array}\right], \quad G(C) = \left[\begin{array}{c} 111111111111111111 \\ \hline 111111100010100010 \\ 110001101011000111 \\ 100011101100110110 \\ 001011001110011011 \\ \hline 100011100010011001 \\ 100011000101111101 \\ 100010000100011010 \\ 100000000111010111 \end{array}\right].$$

Proof. The ODP in dictionary order is found directly using (Subcodes) Chain Algorithm I. The ODP in inverse dictionary order is also found directly using (Subcodes) Chain Algorithm I, but replacing $\mathbb{F}_q^n$ by the unique even weight $[18, 17, 2]$. This is sufficient since the subcodes in inverse dictionary order must be self-complementary, hence the only possible minimum weights are 6 and 8. □

Corollary 4.4. *The* $[18, 9, 6]$ *code contains dimension optimal and minimum distance optimal subcodes with parameters* $[18, 3, 10]$ *and* $[18, 1, 18]$ *(these are maximum dimension subcodes with respect to* $d = 10$ *and* $d = 18$*).*

Fields et. al. classified the even [20,10,6] formally self-dual codes; there are exactly seven codes [13]. We examined the optimum distance profiles of these codes and found the following:

Theorem 4.5. *If* C *is one of the seven formally self-dual even* $[20, 10, 6]$ *codes, then*

$$ODP^{dic}[C] = [6, 6, 6, 6, 8, 8, 10, 10, 12, 12],$$

and

$$ODP^{inv}[C] = [6, 6, 6, 6, 8, 8, 8, 10, 10, 20].$$

Proof. For the seven even formally self-dual $[20, 10, 6]$ codes, (Subcodes) Chain Algorithm I is difficult to implement (this is due to the fact that the first maximal subcode with optimum minimum distance occurs at dimension 6). However, for each $[20, 10, 6]$ code C we can obtain a set $C_{sc:[20,8,6]}$ of all self-complementary [20,8,6] subcodes of C using this algorithm. Jaffe has a classification for all even [20,7,8] codes [20] (generator matrices for these codes and the unique [20,8,8] code may be found at [19]). By applying (Supercodes) Chain Algorithm II to all [20,7,8] codes we determine that there are no even supercodes with parameters [20,10,6]; therefore $\mathrm{ODP}^{dic}[C]_i = 6$ for $7 \leq i \leq 10$. Since all formally self-dual even $[20, 10, 6]$ codes have non-zero weights $6, 8, 10, 12, 14, 20$ we may deduce from information about optimal codes at [15] that an upper bound on the ODP in dictionary order is given by $[6, 6, 6, 6, 8, 8, 10, 10, 12, 20]$. Next we examine the three $[20, 4, 10]$ codes (also classified by Jaffe [20]). We may disregard one of these codes since it contains a weight 16 vector. The ODP in dictionary order for these codes is $[10, 10, 12, 12]$, hence a better upper bound on the ODP in dictionary order for each formally self-dual even $[20, 10, 6]$ code is given by $[6, 6, 6, 6, 8, 8, 10, 10, 12, 12]$. By applying (Supercodes) Chain Algorithm II we verify that this is in fact the ODP in dictionary order for each code; this is accomplished by generating all $[20, 6, 8]$ supercodes of the $[20, 4, 10]$ codes, then finding a single self-complementary supercode of dimension 8 equivalent to a code in $C_{sc:[20,8,6]}$.

$ODP^{inv}[C]$ is obtained in a similar manner applied to the unique self-complementary $[20, 3, 10]$ code (this code is unique because when adding a vector to the unique $[20, 2, 10]$ self-complementary code there is only one choice up to permutation to preserve the minimum weight). □

Corollary 4.6. *If C is a formally self-dual even $[20, 10, 6]$ code, then C contains dimension optimal and minimum distance optimal subcodes with parameters $[20, 4, 10]$ and $[20, 1, 20]$ (these are maximum dimension subcodes with respect to*

$d = 10$ *and* $d = 20$*).*

For example, the first formally self-dual even $[20, 10, 6]$ code in [13] has the following generator matrices in the dictionary and the inverse dictionary order respectively.

$$G(C) = \left[\begin{array}{c} 01001011110001111101 \\ 10111001101101000111 \\ \hline 10011111011111110100 \\ 10001010100110101110 \\ \hline 10001010100101010001 \\ 10001010011001011110 \\ \hline 10000111000101101011 \\ 01000110001001110100 \\ 10000100000101100100 \\ 10000000001101111010 \end{array}\right], \quad G(C) = \left[\begin{array}{c} 11111111111111111111 \\ \hline 11111000001011100001 \\ 11101010000101011010 \\ \hline 11100011010010001110 \\ 11000010010100100011 \\ 10000011101001001001 \\ \hline 10000011001101110101 \\ 10000000001101111010 \\ 00100001000101010010 \\ 01000001000001100101 \end{array}\right].$$

Gulliver and Östergard have shown that there is a unique formally self-dual odd [20,10,6] code [16]. We examined the optimum distance profiles of this code and found the following:

Theorem 4.7. *If* C *is the formally self-dual odd* $[20, 10, 6]$ *code, then*

$$ODP^{dic}[C] = [6, 7, 8, 8, 8, 8, 8, 8, 12, 16],$$

$$ODP^{inv}[C] = [6, 6, 6, 7, 8, 8, 8, 10, 12, 16]$$

with generator matrix for each profile respectively:

$$G(C) = \left[\begin{array}{c} 01111011110111101111 \\ \hline 10010111011101110100 \\ \hline 01001010010100101001 \\ 10010110000001101001 \\ 10010100101001010010 \\ 10001100011000110001 \\ 10001000000101011011 \\ 10000000101110001110 \\ \hline 01000000000100111110 \\ \hline 10000000001001111100 \end{array}\right], \quad G(C) = \left[\begin{array}{c} 11110111101111011110 \\ \hline 00001011011111110011 \\ \hline 11110010011010100010 \\ \hline 11100001110100110010 \\ 11000011001101101111 \\ 10000010000110110101 \\ \hline 00100010101010100100 \\ \hline 00100010001101010110 \\ 01000000000100111110 \\ 10000000001001111100 \end{array}\right].$$

Proof. The ODP in both orders for this code may be found using the methods of the previous proof, and the classifications due to Jaffe [20] [19]. □

Corollary 4.8. *The* $[20, 10, 6]$ *formally self-dual odd code contains dimension optimal and minimum distance optimal subcodes with parameters* $[20, 9, 7]$ *and* $[20, 8, 8]$ *(these are maximum dimension subcodes with respect to* $d = 7$ *and* $d = 8$*).*

Betsumiya and Harada have also shown that there is a unique optimal [22,11,7] code, and in fact this code is formally self-dual [2]. We examined the optimum distance profiles of this code and found the following:

Theorem 4.9. *If* C *is the unique* $[22, 11, 7]$ *code, then*

$$ODP^{dic}[C] = [7, 8, 8, 8, 8, 8, 8, 8, 12, 12, 16],$$

$$ODP^{inv}[C] = [7, 7, 7, 7, 7, 7, 8, 11, 12, 12, 16]$$

with generator matrix for each profile respectively:

$$G(C) = \left[\begin{array}{c} 01101000111111110111111 \\ \hline 00010101010011111110011 \\ 10100110000101111101101 \\ \hline 01101000001100100110000 \\ 10100100010100000110001 \\ 01100000000011101100010 \\ 00010100000110100110100 \\ 10100100001001100010100 \\ 10100000000100110101100 \\ 10000000001110110010010 \\ \hline 10000000000101101110000 \end{array}\right], \quad G(C) = \left[\begin{array}{c} 11010110111110101111100 \\ \hline 11001111000001100011111 \\ 10101000001011111111000 \\ \hline 01100001011101101010100 \\ \hline 00011001010001100001010 \\ \hline 00000100011011101111100 \\ 10101000000000100011010 \\ 01100000010101010100000 \\ 01100000000011101100010 \\ 10100000000100110101100 \\ 10000000000101101110000 \end{array}\right].$$

Proof. The ODP in dictionary order may be obtained directly using (Subcodes) Chain Algorithm I. The ODP in inverse order may be found using the methods of the previous proof, for the [16,8,5] code, by beginning with a [22,1,16] code and checking that supercodes with high minimum distance eventually generate the [22,11,7] code. □

Corollary 4.10. *The* $[22, 11, 7]$ *formally self-dual odd code contains dimension optimal and minimum distance optimal subcodes with parameters* $[22, 10, 8]$, $[22, 4, 11]$, *and* $[22, 3, 12]$ *(these are maximum dimension subcodes with respect to* $d = 8, 11, 12$*).*

TABLE 4.1

ODP for Near Extremal FSD $[16, 8, 4]$ codes (Part 1)

Weight Distribution	ODP	SD	ID	FSD	Total
$A_{0,16} = 1, A_{4,12} = 4,$ $A_{6,10} = 96, A_8 = 54$	$ODP = [4, 6, 6, 8, 8, 8, 8, 16]$	0	1	0	1
$A_{0,16} = 1, A_{4,12} = 8,$ $A_{6,10} = 80, A_8 = 78$	$ODP^{dic} = [4, 4, 6, 6, 6, 6, 10, 12]$ $ODP^{inv} = [4, 4, 6, 6, 6, 8, 8, 16]$	0	1	0	
	$ODP = [4, 4, 6, 6, 8, 8, 8, 16]$	0	3	0	
	$ODP = [4, 6, 6, 8, 8, 8, 8, 16]$	0	1	0	5
$A_{0,16} = 1, A_{4,12} = 10,$ $A_{6,10} = 72, A_8 = 90$	$ODP^{dic} = [4, 4, 4, 6, 6, 8, 10, 12]$ $ODP^{inv} = [4, 4, 4, 6, 6, 8, 8, 16]$	0	0	1	
	$ODP^{dic} = [4, 4, 6, 6, 6, 8, 10, 12]$ $ODP^{inv} = [4, 4, 6, 6, 6, 8, 8, 16]$	0	0	7	
	$ODP = [4, 4, 6, 6, 8, 8, 8, 16]$	0	6	2	
	$ODP^{dic} = [4, 6, 6, 6, 8, 8, 8, 10]$ $ODP^{inv} = [4, 4, 6, 6, 8, 8, 8, 16]$	0	1	0	17
$A_{0,16} = 1, A_{4,12} = 12,$ $A_{6,10} = 64, A_8 = 102$	$ODP^{dic} = [4, 4, 4, 6, 6, 8, 10, 12]$ $ODP^{inv} = [4, 4, 4, 6, 6, 8, 8, 16]$	0	0	1	
	$ODP = [4, 4, 4, 6, 8, 8, 8, 16]$	0	2	2	
	$ODP^{dic} = [4, 4, 6, 6, 6, 8, 10, 12]$ $ODP^{inv} = [4, 4, 4, 6, 8, 8, 8, 16]$	0	2	1	
	$ODP^{dic} = [4, 4, 6, 6, 6, 8, 10, 12]$ $ODP^{inv} = [4, 4, 6, 6, 6, 8, 8, 16]$	0	1	1	
	$ODP = [4, 4, 6, 6, 8, 8, 8, 16]$	0	14	25	
	$ODP = [4, 4, 6, 8, 8, 8, 8, 16]$	1	5	2	
	$ODP = [4, 6, 6, 8, 8, 8, 8, 16]$	0	1	0	58

TABLE 4.2

ODP for Near Extremal FSD $[16, 8, 4]$ codes (Part 2)

Weight Distribution	ODP	SD	ID	FSD	Total
$A_{0,16} = 1, A_{4,12} = 14,$ $A_{6,10} = 56, A_8 = 114$	$ODP = [4, 4, 4, 6, 8, 8, 8, 16]$	0	0	4	
	$ODP = [4, 4, 4, 8, 8, 8, 8, 16]$	0	2	2	
	$ODP^{dic} = [4, 4, 6, 6, 6, 8, 10, 12]$ $ODP^{inv} = [4, 4, 4, 6, 8, 8, 8, 16]$	0	0	2	
	$ODP^{dic} = [4, 4, 6, 6, 8, 8, 8, 10]$ $ODP^{inv} = [4, 4, 4, 6, 8, 8, 8, 16]$	0	0	8	
	$ODP^{dic} = [4, 4, 6, 6, 8, 8, 8, 10]$ $ODP^{inv} = [4, 4, 4, 8, 8, 8, 8, 16]$	0	1	0	
	$ODP^{dic} = [4, 4, 6, 6, 8, 8, 8, 12]$ $ODP^{inv} = [4, 4, 4, 6, 8, 8, 8, 16]$	0	1	1	
	$ODP^{dic} = [4, 4, 6, 6, 8, 8, 8, 16]$ $ODP^{inv} = [4, 4, 4, 8, 8, 8, 8, 16]$	0	6	3	
	$ODP = [4, 4, 6, 6, 8, 8, 8, 16]$	0	2	4	36

TABLE 4.3

ODP for Near Extremal FSD [16, 8, 4] codes (Part 3)

Weight Distribution	ODP	SD	ID	FSD	Total
$A_{0,16} = 1, A_{4,12} = 16,$ $A_{6,10} = 48, A_8 = 126$	$ODP^{dic} = [4, 4, 4, 6, 6, 8, 10, 12]$ $ODP^{inv} = [4, 4, 4, 6, 6, 8, 8, 16]$	0	1	0	
	$ODP = [4, 4, 4, 6, 8, 8, 8, 16]$	0	2	2	
	$ODP = [4, 4, 4, 8, 8, 8, 8, 16]$	0	3	3	
	$ODP^{dic} = [4, 4, 6, 6, 8, 8, 8, 12]$ $ODP^{inv} = [4, 4, 4, 6, 8, 8, 8, 16]$	0	0	1	
	$ODP^{dic} = [4, 4, 6, 6, 8, 8, 8, 12]$ $ODP^{inv} = [4, 4, 4, 8, 8, 8, 8, 16]$	0	0	1	
	$ODP^{dic} = [4, 4, 6, 6, 8, 8, 8, 16]$ $ODP^{inv} = [4, 4, 4, 8, 8, 8, 8, 16]$	0	2	1	
	$ODP = [4, 4, 6, 6, 8, 8, 8, 16]$	0	1	2	
	$ODP = [4, 4, 6, 8, 8, 8, 8, 16]$	0	2	0	21
$A_{0,16} = 1, A_{4,12} = 20,$ $A_{6,10} = 32, A_8 = 150$	$ODP = [4, 4, 4, 8, 8, 8, 8, 16]$	0	3	0	
	$ODP = [4, 4, 6, 8, 8, 8, 8, 16]$	0	1	0	4
$A_{0,16} = 1, A_{4,12} = 28,$ $A_8 = 198$	$ODP = [4, 4, 4, 8, 8, 8, 8, 16]$	2	0	0	2
Total		3	65	76	144

CHAPTER 5
NETWORK CODING THEORY

5.1 Random Network Coding Notations and Formulation

The area of *Network Coding Theory* was introduced by Yeung and Zhang in 1999 [46]. Ahlswede, Cai, Li, and Yeung expanded the concept in their paper "Network Information Flow" [1]. Since that time, Network Coding Theory has become an active research area. A *communication network* is a finite directed graph $G = (V, E)$ where V is a set of vertices (or nodes) and E is a set of edges (or channels). Symbols may be sent through the network from *source nodes* to*sink nodes*. In 2008, Koetter and Kschischang introduced network coding based on subspaces. We begin with a discussion of their formulation of *random network coding*.

A general formulation of coding on a network is *random network coding* (based on formulation in [22]). Let N be a positive integer and $\mathbb{F}_q$ be a finite field. Given a single source and single sink network G with input (row) vectors $p_1, p_2, \ldots, p_M$ in ${\mathbb{F}_q}^N$ and error (row) vectors $e_1, e_2, \ldots, e_T$ in ${\mathbb{F}_q}^N$ where M and T are non-negative integers. The sink receives the *packets* $y_1, y_2, \ldots, y_L$ for which

$$y_j = \sum_{i=1}^{M} h_{j,i} p_i + \sum_{t=1}^{T} g_{j,t} e_t \tag{5.1}$$

where $h_{j,i}, g_{j,t} \in \mathbb{F}_q$ are unknown random coefficients. In matrix form $y = Hp + Ge$

where H and G are random L by M and L by T matrices and

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_L \end{pmatrix}, p = \begin{pmatrix} p_1 \\ \vdots \\ p_M \end{pmatrix}, e = \begin{pmatrix} e_1 \\ \vdots \\ e_T \end{pmatrix}.$$

Thus, given a network, a primary concern is to obtain the row space of y and the row space of p. In order to model the relationship between these spaces, notation is needed. Let W be an N-dimensional vector space over $\mathbb{F}_q$ and let V be a subspace of W. Let $k \geq 0$, if $\dim(V) > k$, then $\mathcal{H}_k(V)$ returns a randomly chosen k-dimensional subspace of V, otherwise $\mathcal{H}_k(V) = V$. Let $\mathcal{P}(W)$ denote the set of all subspaces of W (also known as the projective geometry of W).

Given a network and an N-dimensional vector space W, the input and output (spaces) are elements of $\mathcal{P}(W)$. Let V denote the channel input (i.e., the row space of p) and U denote the channel output (i.e., the row space of y). If $k = \dim(U \cap V)$ and E is an error space, then we define U and V to always be related as

$$U = \mathcal{H}_k(V) \oplus E \tag{5.2}$$

with $\rho = \dim(V) - k$ erasures and $t = \dim(E)$ errors.

In this way, vector spaces are the "codewords" being sent so it is natural to define a metric on $\mathcal{P}(W)$ to determine the "distance" between two spaces. Define $d : \mathcal{P}(W) \times \mathcal{P}(W) \to \mathbb{Z}_{\geq 0}$ by

$$d(A, B) = \dim(A + B) - \dim(A \cap B) = \dim(A) + \dim(B) - 2\dim(A \cap B) \tag{5.3}$$

where $A + B$ is the smallest subspace containing both A and B (i.e., $A + B = \{a + b : a \in A, b \in B\}$).

5.2 Network Codes

In this section we see the relation between network codes and classical codes. Many classical theorems have been extended to network coding, primary examples being Sphere Packing Bound and Singleton Bound (in [22]), and Johnson Bounds (in [44]).

Given a network and an N-dimensional vector space W over $\mathbb{F}_q$, a *(network) code* $\mathbf{C}$ is a nonempty subset of $\mathcal{P}(W)$. The *minimum distance* of $\mathbf{C}$ is denoted by

$$D(\mathbf{C}) = \min_{X,Y \in \mathbf{C} : X \neq Y} d(X,Y). \tag{5.4}$$

The *maximal dimension* of the codewords of $\mathbf{C}$ is denoted by $\ell(\mathbf{C}) = \max_{X \in \mathbf{C}} \dim(X)$. So a code will be referred to as having *type* $[N, \ell(\mathbf{C}), \log_q |\mathbf{C}|, D]$.

The dimension concept for network codes corresponds to the weight concept for classical codes. For a network code $\mathbf{C}$, if the dimension of each codeword is the same, then $\mathbf{C}$ is a *constant-dimension* code. Hence a constant-dimension code is of type $[N, \ell, \log_q |\mathbf{C}|, D]$. Constant-dimension codes are related to a Johnson scheme, a Grassmann graph, and rank metric codes.

Koetter and Kschischang introduced many important theorems which are analogous to theorems in classical coding theory. The following four theorems are introduced in [22]. For a code $\mathbf{C}$ with given output U the *minimum distance decoder* returns a nearest codeword V from $\mathbf{C}$ (i.e., V is such that for all $V' \in \mathbf{C}$, $d(U,V) \leq d(U,V')$).

Theorem 5.1. *Given a network and a code* $\mathbf{C}$, *if* $V \in \mathbf{C}$ *is the input and* $U = \mathcal{H}_k(V) \oplus E$ *is the received space, with* $\dim(E) = t$. *Then the maximum number of erasures is*

$$\rho = \begin{cases} \ell(\mathbf{C}) - k & \text{if } \ell(\mathbf{C}) - k > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

and the minimum distance decoder decodes V *from* U *as long as* $2(t+p) < D(\mathbf{C})$.

Given an N-dimensional vector space W and let $\mathcal{P}(W,\ell)$ be the set of all ℓ-dimensional subspaces of W. Define the sphere of radius t centered at $V \in \mathcal{P}(W,\ell)$ as

$S(V,\ell,t) = \{U \in \mathcal{P}(W,\ell) : d(U,V) \leq 2t\}$.

Theorem 5.2. *For $t \leq \ell$*

$$|S(V,\ell,t)| = \sum_{i=0}^{t} q^{i^2} \begin{bmatrix} \ell \\ i \end{bmatrix}_q \begin{bmatrix} N-\ell \\ i \end{bmatrix}_q \tag{5.6}$$

where $\begin{bmatrix} n \\ i \end{bmatrix}_q = \prod_{j=0}^{n-1} \frac{q^{n-j}-1}{q^{i-j}-1}$ *is the q-ary Gaussian coefficient which counts the number of i-dimensional subspaces of* $\mathbb{F}_q{}^n$.

Theorem 5.3. *(Sphere-Packing Bound)*

Let $\mathbf{C}$ *be a non-empty subset of* $\mathcal{P}(W,\ell)$ *with* $D(\mathbf{C}) \geq 2t$ *and let* $s = \lfloor \frac{t-1}{2} \rfloor$. *Then*

$$|\mathbf{C}| \leq \frac{\mathcal{P}(W,\ell)}{S(V,\ell,s)} < 4q^{(\ell-s)(N-s-\ell)}. \tag{5.7}$$

Also, there exists a constant-dimension code $\mathbf{C}'$ *with* $D(\mathbf{C}') \geq 2t$ *so that*

$$|\mathbf{C}| \geq \frac{\mathcal{P}(W,\ell)}{S(V,\ell,t-1)} > \frac{1}{16t} q^{(\ell-t+1)(N-t-\ell+1)}. \tag{5.8}$$

Theorem 5.4. *(Singleton Bound)*

Let $\mathbf{C}$ *be any q-ary code in* $\mathcal{P}(W,\ell)$ *of type* $[N, \ell(\mathbf{C}), \log_q |\mathbf{C}|, D]$ *then*

$$|\mathbf{C}| \leq \begin{bmatrix} N-(D-2)/2 \\ max\{\ell, N-\ell\} \end{bmatrix}_q \tag{5.9}$$

Inspired by the previous four theorems (from [22]), Fu and Xia extended the classical Johnson bounds for constant weight codes to the following two Johnson-Type Bounds on constant-dimension network codes (in [44]). Let $A_q[N, 2D, \ell]$ be the maximum size of a $[N, \ell(\mathbf{C}), \log_q |\mathbf{C}|, D]$ constant-dimension code.

Theorem 5.5. *(Johnson Type Bound 1)*

If $(q^\ell - 1)^2 > (q^n - 1)(q^{\ell-D} - 1)$, *then*

$$A_q[N, 2D, \ell] \leq \left\lfloor \frac{(q^\ell - q^{\ell-D})(q^n - 1)}{(q^\ell - 1)^2 - (q^n - 1)(q^{\ell-D} - 1)} \right\rfloor \tag{5.10}$$

Theorem 5.6. *(Johnson Type Bound 2)*

$A_q[N, 2D, \ell] \leq \left\lfloor \frac{(q^n-1)}{(q^\ell-1)} A_q[N-1, 2D, \ell-1] \right\rfloor$

or $A_q[N, 2D, \ell] \leq \left\lfloor \frac{(q^n-1)}{(q^\ell-1)} \left\lfloor \frac{(q^{n-1}-1)}{(q^{\ell-1}-1)} \left\lfloor \cdots \left\lfloor \frac{(q^{n-\ell+D}-1)}{(q^D-1)} \right\rfloor \cdots \right\rfloor \right\rfloor \right\rfloor$

5.3 Results on Self-Complementary and Self-Dual Network Codes

In [22] the idea of a *complementary code* is introduced as an analog to the classical dual of a linear code. Let $\mathbf{C}$ be a constant-dimension code of type $[N, \ell, \log_q |\mathbf{C}|, D]$ then the *complementary code* corresponding to $\mathbf{C}$ is $\mathbf{C}^\perp = \{V^\perp : V \in \mathbf{C}\}$ and $\mathbf{C}^\perp$ is of type $[N, N - \ell, \log_q |\mathbf{C}|, D]$. Let $\mathbf{C}$ be a constant-dimension code and $\mathbf{C} = \mathbf{C}^\perp$, then $\mathbf{C}$ is called a *self-complementary code* and $\mathbf{C}$ is of type $[N, \frac{N}{2}, \log_q |\mathbf{C}|, D]$.

We develop the following theory as an analog to classical self-dual codes. If $\mathbf{C}$ is a set of classical self-dual codes of length N, then $\mathbf{C}$ is called a *self-dual network code* (we will just say *self-dual code* when the context is clear). Note that $\mathbf{C}$ is also a constant-dimension, self-complementary code. For positive integers i and j where $j \leq i$ let $m_{i,j}$ denote the maximum number of self-dual codes in ${\mathbb{F}_q}^{2i}$ such that each pair intersect in exactly j dimensions (i.e., for A and B two classical self-dual codes in $\mathbf{C}$, $d(A, B) = 2i - 2j$). We will now discuss some results on $m_{i,j}$.

Proposition 5.7. $m_{i,1} \leq 2^{i-1} + 1$ *for all positive integers* i.

Proof. The total number of even vectors in ${\mathbb{F}_q}^{2i}$ is 2^{2i-1}. Each self-dual code in this enumeration $m_{i,1}$ only intersects in the all-one and all-zero vectors. So each

self-dual code contains $2^i - 2$ vectors besides these two. Hence if $\mathbf{C}$ is a network code meeting this maximum, then $|C| \leq \frac{(2^{2i-1}-2)}{(2^i-2)} = \frac{2(2^{i-1}+1)(2^{i-1}-1)}{2(2^{i-1}-1)} = 2^{i-1}+1$. $\square$

Conjecture: $m_{i,1} = 2^{i-1} + 1$ for all positive integers i.

Lemma 5.8. *The conjecture is true for* $i \leq 5$.

Proof. For $i = 3$ we have a self-dual code network code meeting the bound. This code consists of the following classical self-dual codes:

$$\begin{bmatrix} 11 & 00 & 00 \\ 00 & 11 & 00 \\ 00 & 00 & 11 \end{bmatrix}, \begin{bmatrix} 10 & 00 & 10 \\ 01 & 10 & 00 \\ 00 & 01 & 01 \end{bmatrix}, \begin{bmatrix} 10 & 01 & 00 \\ 01 & 00 & 10 \\ 00 & 10 & 01 \end{bmatrix},$$
$$\begin{bmatrix} 10 & 00 & 01 \\ 01 & 01 & 00 \\ 00 & 10 & 10 \end{bmatrix}, \begin{bmatrix} 10 & 10 & 00 \\ 01 & 00 & 01 \\ 00 & 01 & 10 \end{bmatrix}$$

For $i = 4$ we have a self-dual code network code meeting the bound. This code consists of the following classical self-dual codes:

$$\begin{bmatrix} 1100 & 0000 \\ 0011 & 0000 \\ 0000 & 1100 \\ 0000 & 0011 \end{bmatrix}, \begin{bmatrix} 1010 & 0000 \\ 0100 & 1000 \\ 0001 & 0010 \\ 0000 & 0101 \end{bmatrix}, \begin{bmatrix} 1001 & 0000 \\ 0100 & 0100 \\ 0010 & 0001 \\ 0000 & 1010 \end{bmatrix}, \begin{bmatrix} 1000 & 1000 \\ 0100 & 0001 \\ 0010 & 0010 \\ 0001 & 0100 \end{bmatrix},$$
$$\begin{bmatrix} 1000 & 0100 \\ 0100 & 0010 \\ 0010 & 1000 \\ 0001 & 0001 \end{bmatrix}, \begin{bmatrix} 1000 & 0010 \\ 0101 & 0000 \\ 0010 & 0100 \\ 0000 & 1001 \end{bmatrix}, \begin{bmatrix} 1000 & 0001 \\ 0110 & 0000 \\ 0001 & 1000 \\ 0000 & 0110 \end{bmatrix},$$

$$\left[\begin{array}{cc} 1000 & 1101 \\ 0100 & 0111 \\ 0010 & 1110 \\ 0001 & 1011 \end{array}\right], \left[\begin{array}{cc} 1000 & 1110 \\ 0100 & 1011 \\ 0010 & 0111 \\ 0001 & 1101 \end{array}\right]$$

For $i = 5$ we have a self-dual code network code meeting the bound. This code consists of the following classical self-dual codes:

$$\left[\begin{array}{c} 1100000000 \\ 0011000000 \\ 0000110000 \\ 0000001100 \\ 0000000011 \end{array}\right], \left[\begin{array}{c} 1010000000 \\ 0100100000 \\ 0001001000 \\ 0000010010 \\ 0000000101 \end{array}\right], \left[\begin{array}{c} 1001000000 \\ 0100010000 \\ 0010000100 \\ 0000100001 \\ 0000001010 \end{array}\right], \left[\begin{array}{c} 1000100000 \\ 0101000000 \\ 0010000010 \\ 0000010100 \\ 0000001001 \end{array}\right],$$

$$\left[\begin{array}{c} 1000010000 \\ 0110000000 \\ 0001000001 \\ 0000101000 \\ 0000000110 \end{array}\right], \left[\begin{array}{c} 1000001000 \\ 0100000010 \\ 0010100000 \\ 0001000100 \\ 0000010001 \end{array}\right], \left[\begin{array}{c} 1000000100 \\ 0100000001 \\ 0010001000 \\ 0001010000 \\ 0000100010 \end{array}\right], \left[\begin{array}{c} 1000000010 \\ 0100011100 \\ 0010001101 \\ 0001011001 \\ 0000110101 \end{array}\right],$$

$$\left[\begin{array}{c} 1000000001 \\ 0100010110 \\ 0010011100 \\ 0001001110 \\ 0000111010 \end{array}\right], \left[\begin{array}{c} 1000010101 \\ 0100001000 \\ 0010010011 \\ 0001010110 \\ 0000100111 \end{array}\right], \left[\begin{array}{c} 1000101010 \\ 0100000100 \\ 0010101000 \\ 0001100011 \\ 0000011011 \end{array}\right], \left[\begin{array}{c} 1000001110 \\ 0100000111 \\ 0010010000 \\ 0001001011 \\ 0000101101 \end{array}\right],$$

$$\begin{bmatrix} 1000011010 \\ 0100001110 \\ 0010000001 \\ 0001011100 \\ 0000110110 \end{bmatrix}, \begin{bmatrix} 1000001101 \\ 0100001011 \\ 0010000111 \\ 0001100000 \\ 0000011110 \end{bmatrix}, \begin{bmatrix} 1000011100 \\ 0100001101 \\ 0010010101 \\ 0001000010 \\ 0000111001 \end{bmatrix},$$

$$\begin{bmatrix} 1000001011 \\ 0100011001 \\ 0010011010 \\ 0001010011 \\ 0000100100 \end{bmatrix}, \begin{bmatrix} 1000000111 \\ 0100100110 \\ 0010100011 \\ 0001100101 \\ 0000011000 \end{bmatrix}$$

□

Proposition 5.9. $m_{i,i-1} = 3$ *for all positive integers* i.

Proof. By Theorem 9.5.4 (pp. 360) in [18], there are exactly three self-dual codes D_1, D_2, D_3 containing a fixed $[2i, i-1]$ self-complementary, self-orthogonal code. Therefore $m_{i,i-1} \geq 3$. So we need to show it is impossible to have a set of self-dual codes of size larger than three, such that each pair shares $i-1$ dimensions (in pairwise intersection). Suppose there exists a self-dual code B where $D_1 \cap B \neq D_2 \cap B$ yet $d(D_1, B) = 2 = d(D_2, B)$. $\dim(D_1 \cap D_2 \cap B) = i-2$ is obtained by applying pigeonhole principle, since the pairwise intersection of any of D_1, D_2, or B has dimension $i-1$. Let $\{v_3, v_4, \ldots, v_i\}$ be basis for $D_1 \cap D_2 \cap B$. Fix v_1, v_2, u s.t. $\{v_1, v_2, v_3, \ldots, v_i\}$ is basis for D_1, and $\{u, v_2, v_3, \ldots, v_i\}$ is basis for A_2. This implies $\{v_1, u, v_3, \ldots, v_i\}$ must be a basis for B. But $v_1 \cdot v_j = 0$ for all $j \in \{2, \ldots, i\}$ as D_1 is self-dual and $u \cdot v_j = 0$ for all $j \in \{2, \ldots, i\}$ as D_2 is self-dual. Hence $v_1 \cdot u = 1$(if not then $u \in D_1$). This contradicts the self-duality of B. □

Our next aim is to show $m_{i,j} \geq m_{i-j+1,1}$ for all positive integers i, j. The proof of this claim requires some notations and lemmas involving the trivial self-dual code. Given any N positive even integer, let $C_N^{[11]}$ be the self-dual code

generated by weight 2 vectors in ${\mathbb{F}_2}^N$ that are the rows of the block-matrix with $\frac{N}{2}$ blocks of the 1 by 2 matrix [11]. In other words, the code is generated by the matrix

$$G_N^{[11]} = [g_{i,j} \in \mathbb{F}_2 : g_{i,j} = 1 \text{ iff } j \in \{2i-1, 2i\} \text{ and } g_{i,j} = 0 \text{ otherwise}]$$

For positive integers i and j where $j \leq i$ let $\alpha_{i,j}$ denote the maximum number of self-dual codes in ${\mathbb{F}_2}^{2i}$ such that each pair intersect in some j dimensional subspace S of $C_N^{[11]}$ (so for A and B two self-dual codes in the enumeration $A \cap B = S$).

Lemma 5.10. *$m_{i,1} = \alpha_{i,1}$ for all positive integers i.*

Proof. The single dimension shared is generated by the all-one vector. □

Lemma 5.11. *$m_{i,j} \geq \alpha_{i,j}$ for all positive integers i, j.*

Proof. The proof follows directly from definitions since the latter is a special case of the former. □

For a given set of vectors S of length N let $(S)_{00}^k$ denote the set of all vectors of length $N + 2k$ obtained from S by appending k pairs of zeros to the end of the vectors in S.

Lemma 5.12. *$\alpha_{i-k,1} \leq \alpha_{i,k+1}$ for all positive integers i and $k \in \{0, 1, 2, \ldots, i-1\}$.*

Proof. Fix any $k \in \{0, 1, 2, \ldots, i-1\}$. Recall $\alpha_{i-k,1}$ is the maximum number of self-dual codes in ${\mathbb{F}_2}^{2i-2k}$ intersecting in exactly $< \mathbf{1} >$ (where $\mathbf{1}$ is the all-one vector). Let $\mathbf{C}$ be a fixed set of such self-dual codes indexed by l. Let B_l denote a basis containing $\mathbf{1}$ for the corresponding code in $\mathbf{C}$. Then a set $\mathbf{C}'$ of $\alpha_{i,k+1}$ self-dual codes in ${\mathbb{F}_2}^{2i}$ intersecting in a $k+1$ dimensional subspace of $C_i^{[11]}$, say $< \mathbf{1}, g_i, \ldots, g_{i-(k-1)} : g_j$'s are rows of $G_{2i}^{[11]} >$, will be so that a subspace of $\mathbf{C}'$ can be generated by $B_l' := \{\mathbf{1}, g_i, \ldots, g_{i-(k-1)}\} \cup (B_l - \mathbf{1})_{00}^k$. Hence the indices are the same for the codes from $\mathbf{C}$ in $\mathbf{C}'$. Hence $\alpha_{i-k,1} \leq \alpha_{i,k+1}$. □

Example 5.13. This example is relavent to Lemma 5.12 Let $i = 5$ and $k = 2$

$\alpha_{3,1} \leq \alpha_{5,3}$

$$\left[\begin{array}{ccc} 11 & 00 & 00 \\ 00 & 11 & 00 \\ \hline 11 & 11 & 11 \end{array}\right] \rightarrow \left[\begin{array}{ccccc} 11 & 00 & 00 & \mathbf{00} & \mathbf{00} \\ 00 & 11 & 00 & \mathbf{00} & \mathbf{00} \\ \hline 00 & 00 & 00 & 11 & 00 \\ 00 & 00 & 00 & 00 & 11 \\ 11 & 11 & 11 & 11 & 11 \end{array}\right]$$

$$\left[\begin{array}{ccc} 10 & 01 & 00 \\ 01 & 00 & 10 \\ \hline 11 & 11 & 11 \end{array}\right] \rightarrow \left[\begin{array}{ccccc} 10 & 01 & 00 & \mathbf{00} & \mathbf{00} \\ 01 & 00 & 10 & \mathbf{00} & \mathbf{00} \\ \hline 00 & 00 & 00 & 11 & 00 \\ 00 & 00 & 00 & 00 & 11 \\ 11 & 11 & 11 & 11 & 11 \end{array}\right]$$

Now we may prove the proposition using the previous lemmas.

Proposition 5.14. *$m_{i,j} \geq m_{i-j+1,1}$ for all positive integers i, j.*

Proof. From the previous lemmas we have the following inequalities:

$$m_{i-j+1,1} = \alpha_{i-j+1,1} \tag{5.11}$$

$$\alpha_{i-j+1,1} \leq \alpha_{i,j} \tag{5.12}$$

$$\alpha_{i,j} \leq m_{i,j} \tag{5.13}$$

Where (5.11) follows by Lemma 5.10, (5.12) follows by Lemma 5.12, and (5.13) follows by Lemma 5.11. Therefore $m_{i,j} \geq m_{i-j+1,1}$ for all positive integers i, j. □

CHAPTER 6
COMPLEMENTARY INFORMATION SET CODES

6.1 Motivations

A generalization of self-dual codes was recently proposed by Carlet, Gaborit, Kim, and Solé in [7]. In the paper, a new class of codes, called *complementary information set* (or *CIS*) codes, is defined. Given an integer n, a binary linear code with parameters $[2n, n, d]$ which has two disjoint information sets is a *complementary information set* code. CIS codes have a variety of connections and applications; the authors (in [7]) note the direct applications found in Cryptography, with relations to Boolean S-Boxes, Boolean functions, and masking [27, 28, 29, 38]. In particular, coordinate permutations F on $\mathbb{F}_2^n$, such that given some maximal integer d, for every pair of vectors $a, b \in \mathbb{F}_2^n$ such that (a, b) is nonzero and has Hamming weight less than d, the value of the Walsh Hadamard transform of F at (a, b) is null. These functions are called *Graph Correlation Immune* of order d (or *d-GCI*). A d-GCI function protects against an attack of order d and if the function is linear, then it is equivalent to a $[2n, n, d]$ CIS code.

In the paper [7], CIS codes are classified for $2n = 2, 4, 6, 8, 10, 12$. In the proceeding sections we classify $[14, 7, d]$ CIS codes (i.e. the case where $n = 7$), and we give some results towards the classification for $2n = 16$. For length 14 we use a modified method involving equivalency classes of $GL(n, \mathbb{F}_2)$. Using this method we verified that all CIS codes for lengths 2,4,6,8,10, and 12 are the same as the

classification given in [7]. For length 16 we classify all $[16, 8, 4]$ codes and verify which ones are CIS.

6.2 A Classification Tool Using Graph Isomorphism

The classification of various classes of binary linear $[n, k, d]$ codes is a classical problem; in the previous chapters we have discussed researchers' work towards the classification of self-dual and formally self-dual codes. Thus, an interesting problem in the area of CIS codes is the classification problem. One main difficulty that arises when classifying codes is the equivalency test. When comparing a small set of codes the equivalency test can be implemented easily (in Magma [6]) by performing a pairwise comparison of all codes in the set. However, when comparing more than a few thousand codes the test becomes rather time consuming. A useful solution for this problem, proposed independently in 1978 by [12, 37], is to generate a list of inequivalent combinatorial objects (codes) by producing a "canonical representative" for each equivalency class. This method is described by Kaski and Östergard and it is called *Orderly Generation* ([21] pp.120-124). There is no equivalence test in this method, the only criterion is set membership.

The difficulty in applying the *Orderly Generation* method is finding a way to determine a "canonical representative" for each class. As suggested in [21], a clever navigation of this difficulty is to make use of Brendan McKay's graph isomorphism program *nauty* [31]. Two graphs G and G' with vertex sets V and V' are said to be *isomorphic* if there exists a bijection $\phi : V \to V'$ such that (u, v) is an adjacent pair of vertices in G if and only if $(\phi(u), \phi(v))$ is an adjacent pair of vertices in G'. Given a graph G with vertex set V and a fixed labeling on the vertices with the integers $1, 2, ..., |V|$, *nauty* can output a "canonical" labeling among all isomorphic graphs. In fact, if the graph is a colored graph, then *nauty* will give a canonical

labeling which preserves the color among labels. In [32], Östergard uses *nauty* functionality to classify binary linear codes of minimum distance greater than two for up to length 14. In [39], Schaathun implements a search which classifies all $[36, 8, 16]$ linear codes and uses *nauty*.

6.3 A Correspondence Between Codes and Graphs

Now we must describe how to transform a linear code to a colored graph. As per the formulations in [21, 32, 39], let a linear $[n, k, d]$ code C be given. Let S be the set of minimum weight in C. If S does not generate C, then include all codewords in C of weight 1 higher then the maximum weight in S. Repeat the last step until S generates C. Fix an ordering on S so that c_i represents a specific element of S for $i \in \{1, ..., |S|\}$. Construct a set of $|S| + n$ vertices labelled with the integers $1, 2, ..., |S| + n$ (denote v_i the vertex with label i). Construct a bipartite graph in the following way. Let $\{v_1, v_2, ..., v_{|S|}\}$ be one partite set, and let the other partite set be $\{v_{|S|+1}, v_{|S|+2}, ..., v_{|S|+n}\}$. Draw an edge $(v_i, v_{|S|+j}$ if and only if c_i has a 1 in coordinate j. Color vertices $\{v_1, v_2, ..., v_{|S|}\}$ black. Color vertices $\{v_{|S|+1}, v_{|S|+2}, ..., v_{|S|+n}\}$ red. The following lemma is adapted from the known formulations in [21, 32, 39].

Lemma 6.1. *A permutation α_1 of the labels on the black vertices corresponds to a permutation of the ordering on the codewords. A permutation α_2 of the labels of the red vertices corresponds to a permutation of columns of codewords. As a result, applying α_1 and α_2 to a graph G (constructed from a code C'), yields a graph G' (corresponding to a code C' equivalent to C).*

Proof. The first claim is clear from the construction since c_i corresponds to vertex v_i. The second claim follows from the fact that if $\alpha_2(v_{|S|+i}) = v_{|S|+j}$, then all codewords which had a 1 in column i, now have a 1 in column j after applying α_2.

Since α_1 and α_2 correspond to permuting generators and columns in the code C to obtain G', then G' must correspond to a code C' equivalent to C. $\square$

Because of the functionality in *nauty*, a canonically labelled graph (with the color restriction described above) corresponds to a canonical form of a linear $[n, k, d]$ code. Therefore we may apply *Orderly Generation.*

6.4 A Correspondence Between $GL(n, \mathbb{F}_2)$ and Graphs

Given any linear $[2n, n, d]$ code C, it is clear that if the coordinate set $\{1, 2, ..., n\}$ is an information set, then any generator matrix of C has the form $G = [I|A]$, after performing Gaussian Elimination, where I is the n by n identity matrix and A is an n by n matrix. In [7] this is called the *systematic form* of the generator matrix for a $[2n, n, d]$ code C. C is CIS if and only if C may be converted to *systematic form* where $A \in GL(n, \mathbb{F}_2)$, by Lemma IV.1 of [7]. Hence if the equivalency classes of $GL(n, \mathbb{F}_2)$ are classified, then the classification of CIS codes can be obtained using the ideas of Section 6.3. Therefore an interesting related classification problem is to find all equivalency classes of $GL(n, \mathbb{F}_2)$ (under row and column permutations).

We now describe how to transform an element of $GL(n, \mathbb{F}_2)$ to a colored bipartite graph. Similar to the method of Section 6.3, let $A \in GL(n, \mathbb{F}_2)$. Construct a set of $2n$ vertices labelled with the integers $1, 2, ..., 2n$ (denote v_i the vertex with label i). Construct a bipartite graph in the following way. Let $\{v_1, v_2, ..., v_n\}$ be one partite set, and let the other partite set be $\{v_{n+1}, v_{n+2}, ..., v_{2n}\}$. Draw an edge $(v_i, v_{n+j}$ if and only if row i has a 1 in column j. Color vertices $\{v_1, v_2, ..., v_n\}$ black. Color vertices $\{v_{n+1}, v_{n+2}, ..., v_{2n}\}$ red. The following lemma is adapted from the known combinatorial formulations in [21].

Lemma 6.2. *A permutation α_{row} (resp. α_{col}) of the labels on the black (resp.*

red) vertices corresponds to a permutation of rows (resp. columns). As a result, applying α_{row} and α_{col} to a graph G (constructed from $A \in GL(n, \mathbb{F}_2)$), yields a graph G' (corresponding to an equivalent matrix $A' \in GL(n, \mathbb{F}_2)$).

Proof. The first claim follows from the construction since a row corresponds to a vertex in $\{v_1, v_2, ..., v_n\}$ and a column position corresponds to a vertex in $\{v_{n+1}, v_{n+2}, ..., v_{2n}\}$.

Since α_{row} and α_{col} correspond to permuting rows and columns in the matrix A to obtain G', then G' must correspond to a matrix A' equivalent to A. □

6.5 Length 14 CIS Codes

In order to apply the theories developed in the previous section we need a construction method for the elements of $GL(n, \mathbb{F}_2)$. Our aim in this section is to first classify elements (up to equivalence) in $GL(n, \mathbb{F}_2)$ for $n \leq 7$, then we use these elements to classify all CIS codes of length 14.

We first discuss how to obtain matrices in $GL(n, \mathbb{F}_2)$ using inequivalent matrices from $GL(n-1, \mathbb{F}_2)$. The following two lemmas are adapted from Lemma VI.3 and Proposition VI.4 of [7].

Lemma 6.3. *Any matrix $A \in GL(n, \mathbb{F}_2)$ has a submatrix $A' \in GL(n-1, \mathbb{F}_2)$.*

Proof. Let a_i be the ith column of A and let r_i be the ith row of A where $1 \leq i \leq n$. Delete a_1 from A to obtain an n by $n-1$ matrix A_1. Let r'_i be the ith row of A_1. Since A_1 has rank $n-1$, there exists a j such that $\{r'_i : i \neq j\}$ are linearly independent and $r'_j = \sum_{i \neq j} c_i r'_i$ for uniquely determined c_i. Therefore by deleting r'_j from A_1 we obtain an $n-1$ by $n-1$ matrix A' having rank $n-1$. □

Lemma 6.4. *For any matrix $A' \in GL(n-1, \mathbb{F}_2)$, a matrix $A \in GL(n-1, \mathbb{F}_2)$ may be obtained by the following: For any $x, y \in \mathbb{F}_2^{n-1}$, fix $c := xA^{-1}$ and $z := [1] + cy^T$, then*

$$A = \begin{bmatrix} z & x \\ y^T & A' \end{bmatrix}$$

.

Proof. Since the rows of A' are linearly independent x must be a linear combination of the rows of A', which implies there exists a $c \in \mathbb{F}_2^{n-1}$ such that $cA = x$. Solving for c we obtain $c = xA^{-1}$. To ensure that the top row of A is linearly independent from the other rows the value of z must be such that $c[y^T A'] \neq [z\ x]$. Hence $cy^T \neq z$, and as the values are binary this is equivalent to $cy^T + [1] = z$. □

By applying this theory recursively to all representatives from equivalency classes of $GL(n-1, \mathbb{F}_2)$ along with the canonical selection method in Section 6.4 we may obtain all equivalency class representatives in $GL(n, \mathbb{F}_2)$. For $n = 1, 2, ..., 7$ we have obtained the number of equivalency classes given in Table 6.1. The Magma code for this computation is given in the Appendix.

For each representative A from equivalency classes of $GL(n, \mathbb{F}_2)$, appending the n by n identity matrix I, $[I\ A]$ is a generator matrix for a CIS code. By applying the method introduced in Section 6.3 we can then obtain a set of all inequivalent CIS codes of length $2n$. We have first verified that all CIS codes for lengths 2,4,6,8,10, and 12 are the same as the classification given in [7]. We then obtained a list of all inequivalent CIS codes for length 14. The number of $[14, 7]$ CIS codes is listed in Table 6.2, the rows give the possible minimum distances and the columns tell how many are self-dual, formally self-dual but not self-dual, and neither.

6.6 $[16, 8, 4]$ CIS Codes

Since the number of equivalency classes of $GL(7, \mathbb{F}_2)$ is very large, it is not feasible to determine the classes of $GL(8, \mathbb{F}_2)$. Therefore we consider another

method for considering the $[16, 8]$ CIS codes. We give the following lemma based on the theory of shortening codes in [18] to justify our method.

Lemma 6.5. *If C is a binary $[n, k]$ with generator matrix in standard form G, then shortening C on the first column yields an $[n, k-1]$ code.*

Proof. Since G is in standard form the only row of the generator matrix with a 1 in the first column is the first row. Therefore, shortening on the first column yields an $[n, k-1]$ code. □

Applying this lemma recursively to any $[n, k, d]$ code, a nested chain of subcodes is obtained, the smallest subcode having parameters $[n-k+1, 1, \geq d]$. Therefore, any $[16, 8, 4]$ code has a nested chain of subcodes ("subcode" meaning by adding a zero column it is a subcode):

$$[16, 8, 4] \supset [15, 7, \geq 4] \supset [14, 6, \geq 4] \supset [13, 5, \geq 4]$$

$$\supset [12, 4, \geq 4] \supset [11, 3, \geq 4] \supset [10, 2, \geq 4] \supset [9, 1, \geq 4]$$

If we have a list of all inequivalent $[n', k', \geq 4]$ codes L we construct all $[n'+1, k'+1, \geq 4]$ supercodes by adding a zero column onto each code C in L and then increasing the dimension by adding vectors from $\mathbb{F}_2^{n'+1}/C$. We apply this method recursively and keep only "canonical" representatives as in Section 6.3 to obtain a classification of 255,290 total inequivalent $[16, 8, 4]$ codes. In the Table 6.3 we have the totals for how many of these codes are self-dual, only even formally-self-dual, only odd formally self-dual, and neither self-dual nor formally self-dual. We also include a column which states how many have $d^{\perp} \neq 1$, which means there are no zero columns in the generator matrix. We conclude that there are a total of 267,442 $[16, 8, 4]$ CIS codes.

TABLE 6.1

Number of Equivalency Classes in $GL(n, \mathbb{F}_2)$ Under Row & Column Permutations

$n =$	1	2	3	4	5	6	7
Total	1	2	7	51	885	44,206	6,843,555

TABLE 6.2

Classification of Length 14 CIS codes

	Total CIS	SD	Only FSD	Not SD or FSD
$d = 2$	62015	3	4407	57605
$d = 3$	22561	0	2160	20401
$d = 4$	1476	1 [7]	121	1354
Total	86052	4	6688	79360

TABLE 6.3

Classification of $[16, 8, 4]$ codes and $[16, 8, 4]$ CIS codes

	Total	$d^{\perp} \neq 1$	SD	Only Even FSD	Odd FSD	Not SD or FSD
CIS $[16, 8, 4]$	267,442	267,442	3	141	12,827	254,471
All $[16, 8, 4]$	271,783	268,261	3 [33]	141 [3]	12,827	255,290

CHAPTER 7
CONCLUSION

The classification of self-dual codes continues to be an extremely active area in coding theory. A particularly interesting class of self-dual codes is those of Type II which have high minimum distance (called extremal or near-extremal). It is notable that this class of codes contains famous unique codes: the extended Hamming $[8, 4, 4]$ code, the extended Golay $[24, 12, 8]$ code, and the extended quadratic residue $[48, 24, 12]$ code. A long standing open problem in coding theory is to prove the existence or non-existence of a Type II $[72, 36, 16]$ code.

The aim of Chapters 3 is to shed light on the structure of this interesting class of codes. We examine the maximal subcodes and ODPs of Type II codes for lengths up to 32. Of recent significance is the classification of length 40 Type II codes [4]. The examination of these codes would be extensive work as there are 16470 Type II $[40, 20, 8]$ codes (the highest minimum distance in this case is 8 which is not minimum distance optimal by [15]). Therefore we examined a more interesting case, the unique Type II code of length 48: q_{48}. In the paper, we gave many partial results towards the ODPs of q_{48}. Thus we propose the open problem:

Open Problem: Determine completely the ODP in both orders for q_{48}.

In a similar direction to the Type II codes, we examine all optimal formally self-dual codes for lengths 16-22. We suggest that as the optimal formally self-dual codes become classified for larger lengths, their optimum distance profiles and optimal subcodes should be examined.

A new research area in Information Theory is the area of Network Cod-

ing Theory. Many concepts of classical Coding Theory have been generalized to network codes. In Chapter 5, we develop a generalization of self-dual codes to Network Coding Theory and give results on existence of self-dual network codes with the largest possible minimum distance for lengths up to 10.

A new application of Coding Theory to Cryptography has be formulated in [7]. Complementary Information Set (or CIS) codes were described and classified for lengths up to and including 12. In Chapter 6, we give classification results for length 14 CIS codes and give some partial results on the classification for length 16 CIS codes.

In conclusion, we have described subcode structures in the form of Optimum Distance Profiles and maximum dimension subcodes (with respect to given minimum distance) for notable Type II codes and formally self-dual codes. We have discussed applications of self-dual codes in the areas of Network Coding Theory and Cryptography. In these applications we give results on codes of high minimum distance for self-dual network codes and classifications of CIS codes. As future work, we hope to extend our results to larger lengths and give generalizations of other concepts to Network Coding Theory.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W.-H. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[2] K. Betsumiya and M. Harada. Binary optimal odd formally self-dual codes. *Designs, Codes and Cryptography*, 23(1):11–21, 2001.

[3] K. Betsumiya and M. Harada. Classification of formally self-dual even codes of lengths up to 16. *Designs, Codes and Cryptography*, 23(3):325–332, 2001.

[4] K. Betsumiya, M. Harada, and A. Munemasa. A complete classification of doubly-even self-dual codes of length 40. Online available at *http://arxiv.org/pdf/1104.3727v2.pdf*, 2011. Accessed on 2012-02-25.

[5] I. Bouyuklieva and D.B. Jaffe. Optimal binary linear codes of dimension at most seven. *Discrete Mathematics*, 226:51–70, 2001.

[6] J. Cannon and C. Playoust. *An Introduction to Magma*. University of Sydney, Sydney, Australia, 1994.

[7] C. Carlet, P. Gaborit, J.-L. Kim, and P. Solé. A new class of codes for boolean masking of cryptographic computations. Online available at *http://arxiv.org/pdf/1110.1193v1.pdf*, 2011. Accessed on 2012-03-11.

[8] Y. Chen and A.J. Han Vinck. A lower bound on the optimum distance profiles of the second-order reed-muller codes. *IEEE Transactions on Information Theory*, 56(9):4309–4320, 2010.

[9] J.H. Conway and V. Pless. On the enumeration of self-dual codes. *Journal of Combinatorial Theory, Series A*, 28(1):26–53, 1980.

[10] J.H. Conway, V. Pless, and N.J.A. Sloane. The binary self-dual codes of length up to 32:a revised enumeration. *Journal of Combinatorial Theory, Series A*, 60(2):183–195, 1992.

[11] S. M. Dodunekov, S. B. Encheva, and S. N. Kapralov. On the $[28, 7, 12]$ binary self-complementary codes and their residuals. *Designs, Codes and Cryptography*, 4(1):57–67, 1994.

[12] I. A. Faradzev. Constructive enumeration of combinatorial objects. *Problemes Combinatoires et Theorie des Graphes Colloque Internat, Colloq. Internat. CNRS, 260, CNRS, Paris*, pages 131–135, 1978.

[13] J.E. Fields, P. Gaborit, W.C. Huffman, and V. Pless. On the classification of extremal even formally self-dual codes of lengths 20 and 22. *Discrete Applied Mathematics*, 111(1-2):75–86, 2001.

[14] J. A. Gallian. *Contemporary Abstract Algebra.* Houghton Mifflin Company, Boston, second edition, 2006.

[15] M. Grassl. Bounds on the minimum distance of linear codes and quantum codes. Online available at *http://www.codetables.de*, 2007. Accessed on 2011-04-25.

[16] T.Aaron Gulliver and P.R.J. Östergard. Binary optimal linear rate 1/2 codes. *Discrete Mathematics*, 283(1-3):255–261, 2004.

[17] H. Holma and A. Toskala. *WCDMA for UMTS-HSPA Evolution and LTE, 4th ed.* Wiley, London, U.K., 2007.

[18] W. Huffman and V. Pless. *Fundamentals of error-correcting codes.* University Press, Cambridge, 2003.

[19] D.B. Jaffe. Information about binary linear codes. Online available at *http://www.math.unl.edu/ djaffe2/codes/webcodes/codeform.html*, 2000. Accessed on 2011-04-25.

[20] D.B. Jaffe. Optimal binary linear codes of length ≤ 30. *Discrete Mathematics*, 223(1-3):135–155, 2000.

[21] P. Kaski and P.R.J. Östergard. *Classification Algorithms for Codes and Designs.* Springer, Berlin, Germany, 2006.

[22] R. Koetter and F. Kschischang. Coding for errors and erasures in random network coding. *IEEE Trans. Inf. Theo.*, 54(8):3579–3591, 2008.

[23] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Trans. Net.*, 11(5):782–795, 2003.

[24] A. Kohnert. New [48, 16, 16] optimal linear binary block code. Online available at *http://arxiv.org/pdf/0912.4107.pdf*, 2009. Accessed on 2012-02-25.

[25] Y. Luo, A.J. Han Vinck, and Y. Chen. On the optimum distance profiles about linear block codes. *IEEE Transactions on Information Theory*, 56(3):1007–1014, 2010.

[26] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* North Holland, Amsterdam, 1977.

[27] H. Maghrebi, S. Guilley, C. Carlet, and J.-L. Danger. Classification of high-order boolean masking schemes and improvements of their efficiency. Online available at *http://eprint.iacr.org/2011/520.pdf*, 2011. Accessed on 2012-03-12.

[28] H. Maghrebi, S. Guilley, C. Carlet, and J.-L. Danger. Optimal first-order masking with linear and non-linear bijections. preprint, 2011.

[29] H. Maghrebi, S. Guilley, and J.-L. Danger. Leakage squeezing countermeasure against high-order attacks. *Proceedings of WISTP, LNCS 6633*, pages 208–223, 2011.

[30] J. Maks and J. Simonis. Optimal subcodes of second order reed-muller codes and maximal linear spaces of bivectors of maximal rank. *Designs, Codes and Cryptography*, 21(1-3):165–180, 2000.

[31] B. D. McKay. *nauty* user's guide (version 2.4). Online available at *http://cs.anu.edu.au/ bdm/nauty/nug.pdf*, 2009. Accessed on 2012-03-12.

[32] P.R.J. Östergard. Classifying subspaces of hamming spaces. *Designs, Codes and Cryptography*, 27(3):297–305, 2000.

[33] V. Pless. A classification of self-orthogonal codes over $gf(2)$. *Discrete Mathematics*, 3:215–228, 1972.

[34] V. Pless and N.J.A. Sloane. On the classification and enumeration of self-dual codes. *Journal of Combinatorial Theory*, 18(3):313–335, 1975.

[35] F. Pollara, K. M. Cheung, and R. J. Mceliece. Further results on finite-state codes. *Telecom. and Data Acq. Pro. Rep.*, 42-92 October-December 1987:56–62, 1988.

[36] E.M. Rains and N. J. A. Sloane. *Self-Dual Codes*, pages 177–294. Amsterdam: Elsevier, 1998.

[37] R. C. Read. Every one a winner; or, how to avoid isomorphism search when cataloguing combinatorial configurations. *Annals of Discrete Mathematics*, 2:107–120, 1978.

[38] M. Rivain and E. Prouff. Provably secure higher-order masking of aes. *Proceedings of CHES 2010, LNCS 6225*, pages 413–427, 2010.

[39] H. G. Schaathun. On higher weights and code existence. *Cryptography and Coding: Lecture Notes in Computer Science*, 5921/2009:56–64, 2009.

[40] J. Simonis. The $[18, 9, 6]$ code is unique. *Discrete Mathematics*, 106-107(1):439–448, 1992.

[41] R. Tanner and J. Woodard. *WCDMA-Requirements and Practical Design.* Wiley, London, U.K., 2004.

[42] M. van Dijk, S. Baggen, and L. Tolhuizen. Coding for informed decoders. In *Proceedings. 2001 IEEE International Symposium on Information Theory.*, page 202, 2001.

[43] H. van Tilborg. On the uniqueness resp. nonexistence of certain codes meeting the griesmer bound. *Information and Control*, 44(1):16–35, 1980.

[44] S.-T. Xia and F.-W. Fu. Johnson type bounds on constant dimension codes. *D. C. C.*, 50(2):1–11, 2009.

[45] J. Yan, Z. Zhuang, and Y. Luo. On the optimum distance profiles of some quasi cyclic codes. In *Proceedings. 2011 13th International Conference on Communication Technology.*, pages 979–983, 2011.

[46] R.W. Yeung and Z. Zhang. Distributed source coding for satellite communications. *IEEE Transactions on Information Theory*, 45(4):1111–1120, 1999.

APPENDIX

Program Input:

```
SetLogFile("GL_14.out");
"The following program determines equivalency classes of GL(n,F_2)
(up to row and column permutation) for lengths 2 through 7.";

K:=GF(2);
GL_Matrices:={ColumnSubmatrixRange(GeneratorMatrix(
RepetitionCode(K,2)),2,2)};
GL_Matrices_temp:={@@};

n:=2;

for R in GL_Matrices do
for x in KSpace(K,n-1) do
for y in KSpace(K,n-1) do
  c:=Matrix(x)*(R^-1);
  z:=Matrix(KSpace(K,1)![1])+Matrix(c)*Transpose(Matrix(y));
  A1:=VerticalJoin(HorizontalJoin(z,x),HorizontalJoin(
Transpose(Matrix(y)),R));
GA:=A1;

M1 := MatrixRing( K, NumberOfRows(GA) )!0;
```

```
M2 := MatrixRing( K, NumberOfColumns(GA) )!0;
A := VerticalJoin(HorizontalJoin(M2,Transpose(GA)),
HorizontalJoin(GA,M1));
P := Graph< NumberOfRows(GA)+NumberOfColumns(GA) | A>;
L:=[];

for i:=1 to  #VertexSet(P) do
vv:=Vertices(P)!i;
if "red" in {Label(x) : x in Neighbors(vv) | IsLabelled(x)} then
Append(~L,"blue");
else
Append(~L,"red");
end if;
AssignVertexLabels(~P, L);
end for;

G2:=ColumnSubmatrix(RowSubmatrix(AdjacencyMatrix(
CanonicalGraph(P)),1,NumberOfRows(GA)),NumberOfRows(GA)+1,
NumberOfColumns(GA));

G:=ChangeRing(G2,K);

Include(~GL_Matrices_temp,G);

end for;
end for;
end for;
```

```
"";"";
"Number of equivalency classes for n=2:";
#GL_Matrices_temp;



GL_Matrices:=GL_Matrices_temp;
GL_Matrices_temp:={@@};


n:=3;


for R in GL_Matrices do
for x in KSpace(K,n-1) do
for y in KSpace(K,n-1) do
  c:=Matrix(x)*(R^-1);
  z:=Matrix(KSpace(K,1)![1])+Matrix(c)*Transpose(Matrix(y));
  A1:=VerticalJoin(HorizontalJoin(z,x),HorizontalJoin(
Transpose(Matrix(y)),R));
GA:=A1;


M1 := MatrixRing( K, NumberOfRows(GA) )!0;
M2 := MatrixRing( K, NumberOfColumns(GA) )!0;
A := VerticalJoin(HorizontalJoin(M2,Transpose(GA)),
HorizontalJoin(GA,M1));
P := Graph< NumberOfRows(GA)+NumberOfColumns(GA) | A>;
L:=[];


for i:=1 to  #VertexSet(P) do
```

```
vv:=Vertices(P)!i;
if "red" in {Label(x) : x in Neighbors(vv) | IsLabelled(x)} then
Append(~L,"blue");
else
Append(~L,"red");
end if;
AssignVertexLabels(~P, L);
end for;

G2:=ColumnSubmatrix(RowSubmatrix(AdjacencyMatrix(
CanonicalGraph(P)),1,NumberOfRows(GA)),NumberOfRows(GA)+1,
NumberOfColumns(GA));

G:=ChangeRing(G2,K);

Include(~GL_Matrices_temp,G);

end for;
end for;
end for;
"";"";
"Number of equivalency classes for n=3:";
#GL_Matrices_temp;


GL_Matrices:=GL_Matrices_temp;
GL_Matrices_temp:={@@};
```

```
n:=4;

for R in GL_Matrices do
for x in KSpace(K,n-1) do
for y in KSpace(K,n-1) do
  c:=Matrix(x)*(R^-1);
  z:=Matrix(KSpace(K,1)![1])+Matrix(c)*Transpose(Matrix(y));
  A1:=VerticalJoin(HorizontalJoin(z,x),HorizontalJoin(
Transpose(Matrix(y)),R));
GA:=A1;

M1 := MatrixRing( K, NumberOfRows(GA) )!0;
M2 := MatrixRing( K, NumberOfColumns(GA) )!0;
A := VerticalJoin(HorizontalJoin(M2,Transpose(GA)),
HorizontalJoin(GA,M1));
P := Graph< NumberOfRows(GA)+NumberOfColumns(GA) | A>;
L:=[];

for i:=1 to  #VertexSet(P) do
vv:=Vertices(P)!i;
if "red" in {Label(x) : x in Neighbors(vv) | IsLabelled(x)} then
Append(~L,"blue");
else
Append(~L,"red");
end if;
AssignVertexLabels(~P, L);
```

```
end for;

G2:=ColumnSubmatrix(RowSubmatrix(AdjacencyMatrix(
CanonicalGraph(P)),1,NumberOfRows(GA)),NumberOfRows(GA)+1,
NumberOfColumns(GA));

G:=ChangeRing(G2,K);

Include(~GL_Matrices_temp,G);

end for;
end for;
end for;
"";"";
"Number of equivalency classes for n=4:";
#GL_Matrices_temp;


GL_Matrices:=GL_Matrices_temp;
GL_Matrices_temp:={@@};

n:=5;

for R in GL_Matrices do
for x in KSpace(K,n-1) do
for y in KSpace(K,n-1) do
  c:=Matrix(x)*(R^-1);
```

```
  z:=Matrix(KSpace(K,1)![1])+Matrix(c)*Transpose(Matrix(y));
  A1:=VerticalJoin(HorizontalJoin(z,x),HorizontalJoin(
Transpose(Matrix(y)),R));
GA:=A1;

M1 := MatrixRing( K, NumberOfRows(GA) )!0;
M2 := MatrixRing( K, NumberOfColumns(GA) )!0;
A := VerticalJoin(HorizontalJoin(M2,Transpose(GA)),
HorizontalJoin(GA,M1));
P := Graph< NumberOfRows(GA)+NumberOfColumns(GA) | A>;
L:=[];

for i:=1 to  #VertexSet(P) do
vv:=Vertices(P)!i;
if "red" in {Label(x) : x in Neighbors(vv) | IsLabelled(x)} then
Append(~L,"blue");
else
Append(~L,"red");
end if;
AssignVertexLabels(~P, L);
end for;

G2:=ColumnSubmatrix(RowSubmatrix(AdjacencyMatrix(
CanonicalGraph(P)),1,NumberOfRows(GA)),NumberOfRows(GA)+1,
NumberOfColumns(GA));

G:=ChangeRing(G2,K);
```

```
Include(~GL_Matrices_temp,G);


end for;

end for;

end for;

"";"";

"Number of equivalency classes for n=5:";

#GL_Matrices_temp;



GL_Matrices:=GL_Matrices_temp;

GL_Matrices_temp:={@@};


n:=6;


for R in GL_Matrices do

for x in KSpace(K,n-1) do

for y in KSpace(K,n-1) do

  c:=Matrix(x)*(R^-1);

  z:=Matrix(KSpace(K,1)![1])+Matrix(c)*Transpose(Matrix(y));

  A1:=VerticalJoin(HorizontalJoin(z,x),HorizontalJoin(

Transpose(Matrix(y)),R));

GA:=A1;


M1 := MatrixRing( K, NumberOfRows(GA) )!0;

M2 := MatrixRing( K, NumberOfColumns(GA) )!0;
```

```
A := VerticalJoin(HorizontalJoin(M2,Transpose(GA)),
HorizontalJoin(GA,M1));
P := Graph< NumberOfRows(GA)+NumberOfColumns(GA) | A>;
L:=[];

for i:=1 to  #VertexSet(P) do
vv:=Vertices(P)!i;
if "red" in {Label(x) : x in Neighbors(vv) | IsLabelled(x)} then
Append(~L,"blue");
else
Append(~L,"red");
end if;
AssignVertexLabels(~P, L);
end for;

G2:=ColumnSubmatrix(RowSubmatrix(AdjacencyMatrix(
CanonicalGraph(P)),1,NumberOfRows(GA)),NumberOfRows(GA)+1,
NumberOfColumns(GA));

G:=ChangeRing(G2,K);

Include(~GL_Matrices_temp,G);

end for;
end for;
end for;
"";"";
```

```
"Number of equivalency classes for n=6:";
#GL_Matrices_temp;
"";



GL_Matrices:=GL_Matrices_temp;
GL_Matrices_temp:={@@};

n:=7;

for R in GL_Matrices do
for x in KSpace(K,n-1) do
for y in KSpace(K,n-1) do
  c:=Matrix(x)*(R^-1);
  z:=Matrix(KSpace(K,1)![1])+Matrix(c)*Transpose(Matrix(y));
  A1:=VerticalJoin(HorizontalJoin(z,x),HorizontalJoin(
Transpose(Matrix(y)),R));
GA:=A1;

M1 := MatrixRing( K, NumberOfRows(GA) )!0;
M2 := MatrixRing( K, NumberOfColumns(GA) )!0;
A := VerticalJoin(HorizontalJoin(M2,Transpose(GA)),
HorizontalJoin(GA,M1));
P := Graph< NumberOfRows(GA)+NumberOfColumns(GA) | A>;
L:=[];

for i:=1 to  #VertexSet(P) do
```

```
vv:=Vertices(P)!i;
if "red" in {Label(x) : x in Neighbors(vv) | IsLabelled(x)} then
Append(~L,"blue");
else
Append(~L,"red");
end if;
AssignVertexLabels(~P, L);
end for;

G2:=ColumnSubmatrix(RowSubmatrix(AdjacencyMatrix(
CanonicalGraph(P)),1,NumberOfRows(GA)),NumberOfRows(GA)+1,
NumberOfColumns(GA));

G:=ChangeRing(G2,K);

Include(~GL_Matrices_temp,G);

end for;
end for;
end for;
"";"";
"Number of equivalency classes for n=7:";
#GL_Matrices_temp;
"";
```

Program Output:

```
The following program determines equivalency classes of GL(n,F_2)
(up to row and column permutation) for lengths 2 through 7.

Number of equivalency classes for n=2:
2

Number of equivalency classes for n=3:
7

Number of equivalency classes for n=4:
51

Number of equivalency classes for n=5:
885

Number of equivalency classes for n=6:
44206

Number of equivalency classes for n=7:
6843555
```

INDEX

CURRICULUM VITAE

Finley James Freibert

Education

University of Louisville, Louisville, Kentucky
Ph.D., Applied and Industrial Mathematics, May 2012
M.A., Mathematics, May 2008
DePauw University, Greencastle, Indiana
B.A., Mathematics May 2006
Minors: *Film Studies* and *Sociology*

Professional Experience

University of Louisville, Louisville, Kentucky
Graduate Teaching Assistant, Department of Mathematics, 2006-2012
E.ON U.S. Louisville, Kentucky
Generation Planning Intern, 2009

Publications

Optimum Distance Profiles of Self-Dual Codes and Formally Self-Dual Codes. (with J.-L. Kim) Submitted 2012.
Classification Results for CIS Codes. In Preparation.
On the Self-Duality of Network Codes. (with J.-L. Kim) In Preparation.

Achievements

Referee for journal *Designs, Codes and Cryptography.* 2011-2012
AMS Sectional Meeting Travel Grant. 2011
Passed Actuarial Exam P. 2010
Ken F. and Sandra S. Hohman Fellowship,
Dept. of Math., Univ. of Louisville, 2009-2010
Eugene C. Pulliam, Senior Scholarship, 2005-2006
DePauw University Dean's List, 2002-2006
DePauw University Academic Scholarship, 2002-2006

Talks and Presentations

Classification of Comp. Information Set Codes of Length 14. Mar. 2012
MAA 2012 Sectional Meeting,
Bellarmine University, Louisville, Kentucky.

O.D.P. and Opt. Subcodes of Bin. Self-Dual Type II Codes. Jan. 2012
AMS 2012 Joint Mathematics Meeting,
Special Session on Advances in Coding Theory,
Hynes Convention Center, Boston, Massachusetts.

Optimal Distance Profiles of Binary Self-Dual Type II Codes. Oct. 2011
AMS 2011 Fall Central Sectional Meeting,
Special Session on Coding Theory,
University of Nebraska-Lincoln, Lincoln, Nebraska.

An Introduction to Binary Self-Dual Codes. Oct. 2011
Graduate Student Seminar,
University of Louisville, Louisville, Kentucky.

Optimum Distance Profiles of Binary Self-Dual Codes Oct. 2011
Algebra, Combinatorics, and Number Theory Seminar
University of Louisville, Louisville, Kentucky.

Optimal Subcodes of Binary Self-Dual Type II Codes. May 2011
24th Cumberland Conference on Comb., Graph Theory, and Comp.,
University of Louisville, Louisville, Kentucky.

Codes in Random Network Coding. Mar. 2010
AMS 2010 Spring Southeastern Sectional Meeting,
Special Session on Advances in Algebraic Coding Theory,
University of Kentucky, Lexington, Kentucky.

Codes in Random Network Coding. Mar. 2010
Algebra, Combinatorics, and Number Theory Seminar
University of Louisville, Louisville, Kentucky.

Existence or Non-Existence of a Type II Code. June 2008
Discrete Mathematics Workshop,
University of Louisville, Louisville, Kentucky.