2005

ON THE PROPERTIES AND COMPLEXITY OF MULTICOVERING RADII

Andrew Eugene Mertz
University of Kentucky

ABSTRACT OF DISSERTATION

Andrew Eugene Mertz

The Graduate School
University of Kentucky
2005

ON THE PROPERTIES AND COMPLEXITY OF MULTICOVERING RADII

ABSTRACT OF DISSERTATION

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the College of Engineering at the University of Kentucky

By

Andrew Eugene Mertz
Lexington, Kentucky

Director: Dr. Andrew Klapper, Professor of Computer Science
Lexington, Kentucky
2005

ABSTRACT OF DISSERTATION

ON THE PROPERTIES AND COMPLEXITY OF MULTICOVERING RADII

People rely on the ability to transmit information over channels of communication that are subject to noise and interference. This makes the ability to detect and recover from errors extremely important. Coding theory addresses this need for reliability. A fundamental question of coding theory is whether and how we can correct the errors in a message that has been subjected to interference. One answer comes from structures known as error correcting codes.

A well studied parameter associated with a code is its covering radius. The covering radius of a code is the smallest radius such that every vector in the Hamming space of the code is contained in a ball of that radius centered around some codeword. Covering radius relates to an important decoding strategy known as nearest neighbor decoding.

The multicovering radius is a generalization of the covering radius that was proposed by Klapper [11] in the course of studying stream ciphers. In this work we develop techniques for finding the multicovering radius of specific codes. In particular, we study the even weight code, the 2-error correcting BCH code, and linear codes with covering radius one.

We also study questions involving the complexity of finding the multicovering radius of codes. We show: Lower bounding the m-covering radius of an arbitrary binary code is NP-complete when m is polynomial in the length of the code. Lower bounding the m-covering radius of a linear code is Σ_2^p-complete when m is polynomial in the length of the code. If P is not equal to NP, then the m-covering radius of an arbitrary binary code cannot be approximated within a constant factor or within a factor n^ϵ, where n is the length of the code and $\epsilon < 1$, in polynomial time. Note that the case when $m = 1$ was also previously unknown. If NP is not equal to Σ_2^p, then the m-covering radius of a linear code cannot be approximated within a constant factor or within a factor n^ϵ, where n is the length of the code and $\epsilon < 1$, in polynomial time.

KEYWORDS: Coding Theory, Complexity, Covering Radius, Multicovering Radius, Approximation Complexity.

Andrew Eugene Mertz

September 8, 2005

ON THE PROPERTIES AND COMPLEXITY OF MULTICOVERING RADII

By

Andrew Eugene Mertz

Director of Dissertation: Dr. Andrew Klapper

Director of Graduate Studies: Dr. Grzegorz Wasilkowski

September 8, 2005

RULES FOR THE USE OF DISSERTATIONS

DISSERTATION

Andrew Eugene Mertz

The Graduate School
University of Kentucky
2005

ON THE PROPERTIES AND COMPLEXITY OF MULTICOVERING RADII

DISSERTATION

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the College of Engineering at the University of Kentucky

By

Andrew Eugene Mertz
Lexington, Kentucky

Director: Dr. Andrew Klapper, Professor of Computer Science
Lexington, Kentucky
2005

To my parents, Mary Helen and Gene Mertz. Thank you for the incalculable support and encouragement.

Acknowledgments

Many people have contributed their time and talent to this work. I would like thank my advisor, Andrew Klapper for being a wonderful mentor, teacher, editor, and example; and to the rest of my committee, Kenneth Calvert, Paul Eakin, and Judy Goldsmith, for their time, comments, and evaluations.

I would also like to thank my family and friends for all of their support and being there for me when I really needed it. Especially my wife Jessica Mertz for putting up with many late nights and endless grammar questions. My parents, Mary Helen and Gene Mertz, for more than can be said is so small a space. Ben Vandgrift and Dan Miller for great times and a place to stay when I needed it.

There have been many people who have challenged and inspired me. In particular I would like to thank Clyde Dubbs who helped me see the beauty of mathematics, and an unnamed gentleman who talked to a young boy about planets, stars, and gravity and instilled a desire to learn more.

This research was partially supported by NSF grant #CCF-9706078.

Contents

List of Figures

Chapter 1
Introduction

More and more people have come to rely on the ability to transmit large volumes of information over long distances. The applications are varied: entertainment, finance, communication, and defense to name a few. Complicating the situation is the fact that the channels of communication are subject to noise and interference. This makes the ability to detect and recover from errors extremely important. Also, much of this information is sensitive in nature. Thus, there is a need for reliability and security.

Coding theory addresses this need for reliability. A fundamental question of coding theory is whether and how we can correct the errors in a message that has been subjected to interference. One answer comes from structures known as error correcting codes. Error correcting codes are embedded in almost every device or system that uses some form of communication or storage; from modems and cell phones to hard drives.

In this chapter we describe the basic properties and language of error correcting codes and covering radii. See MacWilliams and Sloane's or Pless and Huffman's book [3] [23] [24] for a more detailed treatment of error correcting codes. For an excellent survey of the covering radius of codes see Cohen et al.'s monograph [3].

1.1 Codes

One use of codes is to correct errors that occur when information is transmitted over a noisy channel. Imagine that we have a radio transmitter that we can use to send binary messages and that sometimes when we send a 0 a recipient receives a 1, or similarly a 1 becomes a 0. Suppose that the probability, independent of the location of a symbol in the string, of such a transposition is p. The communication model that we have just described is called a binary symmetric channel and is illustrated in Figure 1.1. We place the restriction that the

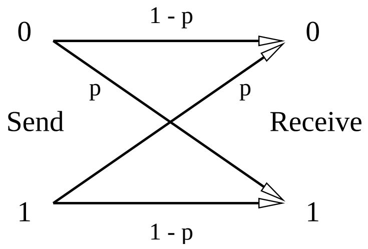

Figure 1.1: The binary symmetric channel with error probability $0 \leq p < 1/2$.

error probability p is less than 1/2. If p were greater than 1/2 we could simply reverse our interpretation of a received symbol; if p were equal to 1/2, then the received symbol would be random.

To protect a message that we wish to transmit we encode it into a larger sequence of symbols. That is, we add some redundancy. We encode a message $\mathbf{m} = m_1 m_2 \ldots m_k$ consisting of k symbols as a codeword $\mathbf{c} = c_1 c_2 \ldots c_n$ where $k \leq n$. This codeword $\mathbf{c}$ is then transmitted instead of $\mathbf{m}$.

Definition: In general, given a set S of q symbols, the set of all strings of length n over S is called the q-ary Hamming space and is denoted $\boldsymbol{F}_q^n$. An arbitrary element of $\boldsymbol{F}_q^n$ is called a vector. A nonempty subset of a Hamming space is a code and its elements are called codewords. If a code contains only one codeword then it is said to be trivial.

We will mostly be focusing on non-trivial codes and the binary Hamming space. For the binary Hamming space we will drop the q in the notation. Sometimes it is of interest to restrict our attention to a particular class of codes known as linear codes.

Definition: We define the componentwise sum, componentwise difference, componentwise product, scalar multiplication, and scalar product of vectors as follows: Suppose $\mathbf{x} = x_1 x_2 \ldots x_n$ and $\mathbf{y} = y_1 y_2 \ldots y_n$ in $\boldsymbol{F}_q^n$ and $\alpha \in \boldsymbol{F}_q^1$ then

$$\mathbf{x} + \mathbf{y} \triangleq (x_1 + y_1)(x_2 + y_3) \ldots (x_n + y_n),$$

$$\mathbf{x} - \mathbf{y} \triangleq (x_1 - y_1)(x_2 - y_3) \ldots (x_n - y_n),$$

$$\mathbf{x} * \mathbf{y} \triangleq (x_1 * y_1)(x_2 * y_3) \ldots (x_n * y_n),$$

$$\alpha \mathbf{x} \triangleq (\alpha * x_1)(\alpha * x_2) \ldots (\alpha * x_n),$$

and

$$\mathbf{x} \cdot \mathbf{y} \triangleq x_1 * y_1 + x_2 * y_2 + \cdots + x_n * y_n.$$

Definition: The vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be orthogonal if $\mathbf{x} \cdot \mathbf{y} = 0$.

Definition: A code $C \subseteq \boldsymbol{F}_q^n$ is said to be linear if all of the pairwise sums and scalar multiples of codewords are also in the code.

In other words a linear code is closed under scalar multiplication and componentwise addition. So a linear code is a linear subspace of $\boldsymbol{F}_q^n$. Therefore there exists a basis for C.

Definition: Let $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ be a set of linearly independent codewords where k is maximal. Then the $k \times n$ matrix

$$\begin{pmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_k \end{pmatrix}$$

is called a generator matrix of C. The codewords of C are the q^k linear combinations of the rows in the generator matrix.

Definition: Given a linear code $C \subseteq \boldsymbol{F}_q^n$, the linear code that consists of the vectors which are orthogonal to every codeword of C is called the dual of C and is denoted:

$$C^{\perp} \triangleq \{\mathbf{v} \in \boldsymbol{F}_q^n : \mathbf{v} \cdot \mathbf{c} = 0 \text{ for all } \mathbf{c} \in C\}.$$

Definition: As $C^{\perp}$ is linear it has a generator matrix. Any generator matrix of $C^{\perp}$, which will be an $(n-k) \times n$ matrix, is called a parity check matrix of C.

Often a linear code is specified by giving its parity check matrix. If $\mathbf{H}$ is any parity check matrix of C, the code C can be defined as

$$C = \left\{\mathbf{x} \in \boldsymbol{F}_q^n : \mathbf{H}\mathbf{x}^{\mathrm{T}} = \mathbf{0}^{n-k}\right\}.$$

Definition: Given a parity check matrix $\mathbf{H}$ and vector $\mathbf{x} \in \boldsymbol{F}_q^n$ the vector $\mathbf{H}\mathbf{x}^{\mathrm{T}} \in \boldsymbol{F}_q^{n-k}$ is called the syndrome of $\mathbf{x}$. Thus the code C consists of exactly the vectors with syndrome $\mathbf{0}^{n-k}$.

1.1.1 Maximum Likelihood Decoding

When a codeword $\mathbf{c}$ is sent through the channel it is subject to noise and may be altered. Thus the received vector $\mathbf{r}$ may be different from $\mathbf{c}$. The effect of the channel can be

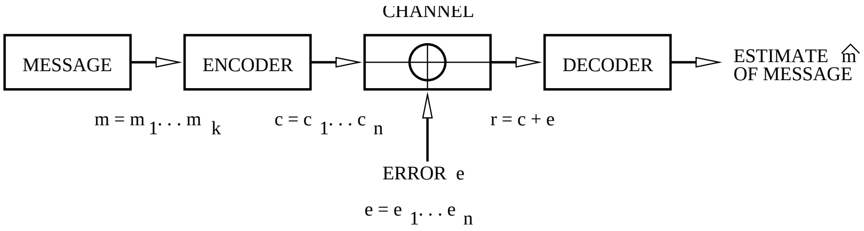

Figure 1.2: Communication model

represented by a vector $\mathbf{e} = \mathbf{r} - \mathbf{c}$ called the error vector. The channel can be thought of as adding the error vector to the transmitted codeword.

Upon receiving the vector $\mathbf{r}$ the decoder must determine what codeword was really sent and therefore what message was sent. One decoding strategy, maximum likelihood decoding, assumes that the most likely error vector was the one that occurred. We will now introduce some definitions to help examine this decoding strategy.

Definition: The Hamming distance between two vectors $\mathbf{x}, \mathbf{y} \in \boldsymbol{F}_q^n$ is the number of coordinates in which they differ.

$$\text{dist}(\mathbf{x}, \mathbf{y}) \triangleq |\{i : x_i \neq y_i\}|.$$

Definition: The Hamming weight of a vector $\mathbf{x} \in \boldsymbol{F}^n q$ is the number of nonzero coordinates of $\mathbf{x}$,

$$\text{wt}(\mathbf{x}) \triangleq \text{dist}(\mathbf{x}, \mathbf{0}^n).$$

Thus

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \text{wt}(\mathbf{x} - \mathbf{y}).$$

In the binary case $\mathbf{x} + \mathbf{y} = \mathbf{x} - \mathbf{y}$ so for binary vectors

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \text{wt}(\mathbf{x} + \mathbf{y}).$$

Definition: The support of a vector $x \in \boldsymbol{F}_q^n$ is the set

$$\text{supp}(x) \triangleq \{i : x_i \neq 0\}.$$

In the case of the binary alphabet we have

$$\begin{aligned} \text{wt}(\mathbf{x} + \mathbf{y}) &= \text{wt}(\mathbf{x}) + \text{wt}(\mathbf{y}) - 2\text{wt}(\mathbf{x} * \mathbf{y}) \\ &= \text{wt}(\mathbf{x}) + \text{wt}(\mathbf{y}) - 2|\text{supp}(\mathbf{x}) \cap \text{supp}(\mathbf{y})|. \end{aligned}$$

Definition: Also, we will consider the weight of a set of vectors to be the maximum of the weights of its elements

$$\mathrm{wt}(S) \triangleq \max_{\mathbf{x} \in S}(\mathrm{wt}(\mathbf{x})).$$

Definition: The distance from a vector to a set of vectors to be the minimum distance between the vector and the elements of the set

$$\mathrm{dist}(\mathbf{x}, S) \triangleq \min_{\mathbf{s} \in S}(\mathrm{dist}(\mathbf{x}, \mathbf{s})).$$

Definition: The Hamming sphere of radius r centered on the vector $\mathbf{x} \in \boldsymbol{F}_q^n$ is:

$$B_r(\mathbf{x}) \triangleq \{\mathbf{y} \in \boldsymbol{F}_q^n : \mathrm{dist}(\mathbf{x}, \mathbf{y}) \leq r\}$$

and its cardinality, or volume, is

$$V_q(n, r) \triangleq \sum_{i=0}^{r} \left(\begin{array}{c} n \\ i \end{array} \right) (q-1)^i.$$

The Hamming distance satisfies the triangle inequality and is a metric for the Hamming space.

Lemma 1.1 *Given any three vectors* $\mathbf{x}$*,* $\mathbf{y}$ *and* $\mathbf{z}$ *in* $\boldsymbol{F}_q^n$*,* $\mathit{dist}(\mathbf{x}, \mathbf{y}) + \mathit{dist}(\mathbf{y}, \mathbf{z}) \geq \mathit{dist}(\mathbf{x}, \mathbf{z})$*.*

Proof: Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \boldsymbol{F}_q^1$. If we assume that $\mathbf{x} = \mathbf{z}$ then $\mathrm{dist}(\mathbf{x}, \mathbf{z}) = 0$. Therefore, $\mathrm{dist}(\mathbf{x}, \mathbf{y}) + \mathrm{dist}(\mathbf{y}, \mathbf{z}) \geq \mathrm{dist}(\mathbf{x}, \mathbf{z})$ as distance cannot be negative. If $\mathbf{x} \neq \mathbf{z}$ then $\mathrm{dist}(\mathbf{x}, \mathbf{z}) = 1$. Again $\mathrm{dist}(\mathbf{x}, \mathbf{y}) + \mathrm{dist}(\mathbf{y}, \mathbf{z}) \geq \mathrm{dist}(\mathbf{x}, \mathbf{z})$ as $\mathbf{y}$ cannot be equal to both $\mathbf{x}$ and $\mathbf{z}$.

Now let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \boldsymbol{F}_q^k$.

$$\begin{aligned} \mathrm{dist}(\mathbf{x}, \mathbf{y}) + \mathrm{dist}(\mathbf{y}, \mathbf{z}) &= \sum_{1 \leq i \leq k} \mathrm{dist}(x_i, y_i) + \mathrm{dist}(y_i, z_i) \\ &\geq \sum_{1 \leq i \leq k} \mathrm{dist}(x_i, z_i) = \mathrm{dist}(\mathbf{x}, \mathbf{z}). \end{aligned}$$

□

The probability that a symbol is altered is p, so the probability that e_i, the i^{th} coordinate of the error vector $\mathbf{e}$, is nonzero is also p, with all nonzero values equally likely. Thus the probability that $\mathrm{wt}(\mathbf{e}) = j$ is $p^j(1-p)^{n-j}$ where $\mathbf{e}$ has length n, and j is a natural number. Also $p < 1/2$, so $p < 1-p$ and therefore

$$(1-p)^n > p(1-p)^{n-1} > \cdots > p^n.$$

This implies that given two error vectors the one with smaller weight is more likely to have occurred. Since $\mathrm{dist}(\mathbf{c}, \mathbf{r}) = \mathrm{wt}(\mathbf{r} - \mathbf{c}) = \mathrm{wt}(\mathbf{e})$ we can perform maximum likelihood decoding for the binary symmetric channel by decoding the received vector to the codeword that is nearest to it. This is known as nearest neighbor decoding.

Definition: An important parameter of a code is the minimum distance between codewords.

$$\text{minimum distance of } C \triangleq \min_{\mathbf{a},\mathbf{b} \in C, \mathbf{a} \neq \mathbf{b}} \mathrm{dist}(\mathbf{a}, \mathbf{b}).$$

One reason minimum distance is important is that it gives a condition when nearest neighbor decoding is guaranteed to be successful.

Theorem 1.2 *A code with minimum distance d can correct $\lfloor (d-1)/2 \rfloor$ errors.*

Proof: Assume that the codeword $\mathbf{c}$ was sent while $\mathbf{r} = \mathbf{c} + \mathbf{e}$ was received with the weight of $\mathbf{e}$ less than or equal to $(d-1)/2$. Given any other codeword $\mathbf{x}$ the spheres centered on $\mathbf{x}$ and $\mathbf{c}$ of radius $\lfloor (d-1)/2 \rfloor$ must be disjoint. As $\mathbf{r}$ is within the sphere centered on $\mathbf{c}$ its distance to $\mathbf{x}$ must be greater than its distance to $\mathbf{c}$. Therefore $\mathbf{r}$ will be decoded correctly to $\mathbf{c}$ via nearest neighbor decoding. A visual way to see Theorem 1.2 is given in Figure 1.3.

□

1.2 Covering Radius

Another well studied parameter associated with a code is its covering radius. **Definition:** The covering radius of a code is the smallest radius such that every vector in the Hamming space of the code is contained in a ball of that radius centered around some codeword. In other words, the covering radius of a code $t(C)$ is the smallest integer r such that $\forall \mathbf{v} \in \boldsymbol{F}^n,\ \exists \mathbf{c} \in C : \mathrm{dist}(\mathbf{c}, \mathbf{v}) \leq r$.

Like minimum distance, covering radius relates to nearest neighbor decoding. However, covering radius measures the largest number of errors in any correctable error vector.

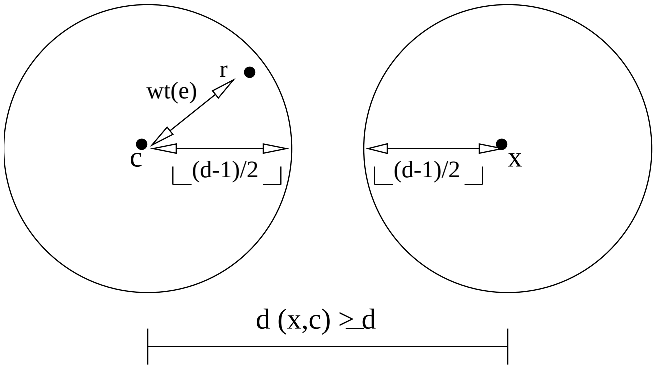

Figure 1.3: A code with minimum distance d can correct $\lfloor(d-1)/2\rfloor$ errors.

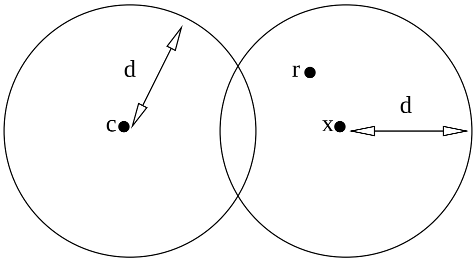

Figure 1.4: A code with covering radius d cannot correct more than d errors.

Theorem 1.3 *A code with covering radius d cannot correct a codeword that has be subject to more than d errors using nearest neighbor decoding.*

Proof: Given a code $C \subseteq \boldsymbol{F}^n$ with covering radius d assume that we have received a vector $\mathbf{r} = \mathbf{c}+\mathbf{e}$, where $\mathbf{c} \in C$, $\mathbf{e} \in \boldsymbol{F}^n$, and the weight of $\mathbf{e}$ is greater than d. This then implies that there must be another codeword $\mathbf{x}$, such that $\text{dist}(\mathbf{x}, \mathbf{r}) \leq d$, as d is the covering radius of the code. Therefore, $\mathbf{r}$ is guaranteed to be decoded incorrectly by nearest neighbor decoding. □

Definition: If the covering radius of a code is less than its minimum distance it is called maximal.

Theorem 1.4 *If a code is maximal then no vector can be added to the code without decreasing its minimum distance.*

1.3 Multicovering Radius

The multicovering radius is a generalization of the covering radius that was proposed by Klapper [11] in the course of studying stream ciphers. A stream cipher can be thought of as a pseudo-random (PR) sequence that is added to the message bit by bit. Given a piece of the PR sequence an attacker would like to be able to predict the rest or at least most of the bits. Thus the designer of the stream cipher wants families of sequences that asymptotically resist all attacks i.e. $\forall m$ $\exists$ sequence S so that for all $1 \leq i \leq m$, S is far from the sequence predicted by attack A_i. This implies that for all $1 \leq i \leq m$, $\overline{S}$ is close to every sequence predicted. The question becomes do such families exists and if they do how hard is it to find such families?

Klapper [12] has shown that there exist stream ciphers that are resistant to synthesis attacks. Some of these results depend on the multicovering radius of Reed-Muller codes. Therefore, the multicovering radius is interesting from a cryptographic standpoint as well as a natural generalization of the covering radius.

Definition: Given a code of length n, the m-covering radius is the smallest radius such that every m-tuple of vectors in the ambient space, the Hamming space of which the code is a subset, of the code is contained in a ball of that radius centered around some codeword. Specifically the m-covering radius of a code $t_m(C)$ is the smallest integer r such that $\forall \mathbf{v}.1, \mathbf{v}.2, \ldots, \mathbf{v}.m \in \boldsymbol{F}^n : \exists \mathbf{c} \in C : \forall i = 1, \ldots, m : \text{dist}(\mathbf{c}, \mathbf{v}.i) \leq r$. A visual example of a vector covering a m-tuple is given in Figure 1.5.

1.4 Translate Leader

Definition: For any $\mathbf{x} \in \boldsymbol{F}_q^n$ and code $C \subseteq \boldsymbol{F}_q^n$ the set

$$\mathbf{x} + C \triangleq \{\mathbf{x} + \mathbf{c} : \mathbf{c} \in C\}$$

is called a translate of C.

Distance is preserved under translation so a code and it translate have the same covering radius. More generally we can define a translate over a set of m vectors.

Definition: Let S be a set of m vectors.

$$S + C \triangleq \{\mathbf{c} + S : \mathbf{c} \in C\}.$$

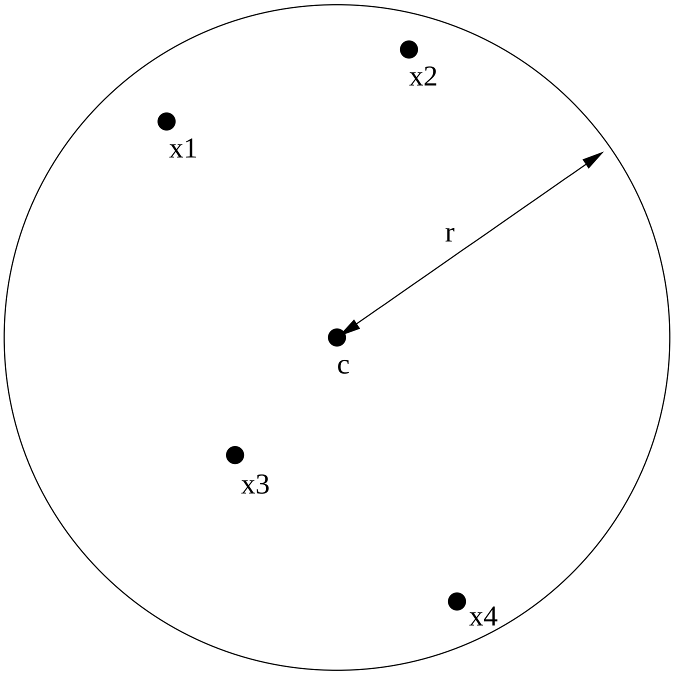

Figure 1.5: Covering of an m-tuple of vectors

A translate leader is an m-tuple $T \in S + C$ such that the weight of T is minimal. The m-covering radius of C is the weight of the maximal weight translate leader.

1.5 Notation

We will use the following notation:

$\mathbf{a}|\mathbf{b} = \mathbf{ab} =$ The concatenation of $\mathbf{a}$ and $\mathbf{b}$.

$\mathbf{0}^n$ and $\mathbf{1}^n$ = the all 0 and all 1 vectors of length n respectively.

Given a binary vector $\mathbf{v}$, $\overline{\mathbf{v}}$ = the complement of $\mathbf{v} = \mathbf{1}^n + \mathbf{v}$.

$v_i = i^{\text{th}}$ component or block of v.

$v.i = i^{\text{th}}$ element of a countable set V.

A code with length n, cardinality K, and minimum distance d is called a (n, K, d) code or if the minimum distance is not needed just a (n, K) code.

A linear code with length n, dimension k, and minimum distance d is called a $[n, k, d]$ or $[n, k]$ code.

Given a set S and vector $\mathbf{x}$, $\text{cov}(\mathbf{x}, S) = \max\{\text{dist}(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in S\}$ = the radius of the smallest ball centered at x that contains all of S.

$\text{cov}(C, S)$ =min$\{\text{cov}(\mathbf{c}, S) : \mathbf{c} \in C\}$.

Using the above notation we can write the m-covering radius of the code $C \subseteq \boldsymbol{F}^n$ as, $t_m(C) = \max\{\text{cov}(C, S) : |S| = m \text{ and } S \subseteq \boldsymbol{F}^n\}$.

Chapter 2

Properties of the Multicovering Radii of Codes

There are some basic relations involving the multicovering radius that hold as the parameters of a code vary, and in this chapter we will look at some of the previous work that has been done to find such properties. The results in this chapter are due to Klapper and Honkala. These properties will also be used in later chapters. Note that in some cases there are differences in the behavior of these relations when m is not equal to one. For example

Theorem 2.1 *If $m \geq 2$ then $t_m(C) \geq \lceil n/2 \rceil$.*

This is true as the closest that one can mutually be to two complementary vectors is $\lceil n/2 \rceil$. Some other straightforward properties of the m-covering radius follow:

Theorem 2.2

1. *If C is a subcode of S then $t_m(C) \geq t_m(S)$.*

2. *For any code C and any $m \geq 1$, $t_m(C) \leq t_{m+1}(C)$.*

Theorem 2.3 (Klapper [11]) *Let C be any code of length n. Then for any positive m, $t_m(C) \leq t_1(C) + t_m(\boldsymbol{F}^n)$.*

Proof: Given any m-tuple of vectors in $\boldsymbol{F}^n$ there exists a vector $\mathbf{x}$ that covers them within radius $t_m(\boldsymbol{F}^n)$. Furthermore, there exists a codeword $\mathbf{c} \in C$ with distance at most $t_1(C)$ from $\mathbf{x}$. The distance from $\mathbf{c}$ to each vector in the m-tuple is less than or equal to $t_1(C) + t_m(\boldsymbol{F}^n)$ by the triangle inequality. This is depicted in Figure 2.1. □

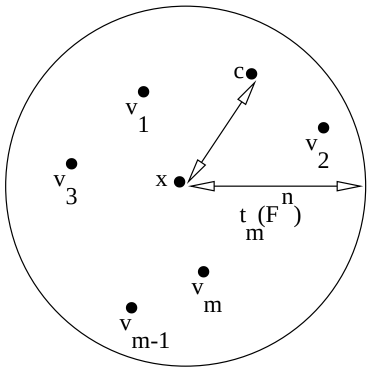

Figure 2.1: $t_m(C) \leq t_1(C) + t_m(\boldsymbol{F}^n)$

2.1 Constructions

2.1.1 Cartesian Product

Definition: Given two codes, A and B, let $C = A \times B = \{\mathbf{a}|\mathbf{b} : \mathbf{a} \in A, \mathbf{b} \in B\}$. The code C is called the Cartesian product or direct sum of A and B. If both A and B are linear then so is their Cartesian product.

Theorem 2.4 (Klapper [11]) *Given two codes, A and B,*

$$t_m(A \times B) \leq t_m(A) + t_m(B).$$

When $m = 1$ the above inequality becomes an equality.

Proof: When $m = 1$ the equality is easy to see since $\mathrm{dist}(\mathbf{x}|\mathbf{y}, A \times B) = \mathrm{dist}(\mathbf{x}, A) + \mathrm{dist}(\mathbf{y}, B)$. Similarly when $m \geq 2$, if S is a set of m vectors in $\boldsymbol{F}^{n_A+n_B}$, where n_A and n_B are the lengths of the codes A and B respectively, then the projections onto the first n_A and following n_B coordinates are within $t_m(A)$ and $t_m(B)$ of some codewords in A and B. □

Note that for $m \geq 2$ the inequality of Theorem 2.4 may be strict. For example if $C = \{00, 01\}$, then $t_2(C) = 2$ but $t_2(C \times C) = 3$.

Theorem 2.5 (Klapper [11]) *Given two codes, $C_1 \subseteq \boldsymbol{F}^{n_1}$ and $C_2 \subseteq \boldsymbol{F}^{n_2}$, and two natural numbers m_1 and m_2*

$$t_{m_1 m_2}(C_1 \times C_2) \geq t_{m_1}(C_1) + t_{m_2}(C_2).$$

Proof: Suppose S_i is a set of m_i vectors of length n_i whose distance to C_i equals d_i for $i = 1$ and 2. Then the distance from $S_1 \times S_2$ to $C_1 \times C_2$ is $d_1 + d_2$. □

2.1.2 Repetition

Definition: For any positive integer r the r-fold repetition of the code C is the code C' consisting of every codeword of the original code concatenated with itself r times, $C' = \{\mathbf{c}|\mathbf{c}| \cdots |\mathbf{c} : \mathbf{c} \in C\}$. This construction also preserves linearity if C is linear.

Theorem 2.6 (Klapper [11]) *If C is a (n, K, d) code and C' is the r-fold repetition of C then C' is (rn, K, rd) code with $rt_m(C) \leq t_m(C') \leq rt_{rm}(C)$.*

Proof: Let $S = \{\mathbf{v}.1, \ldots, \mathbf{v}.m\}$ be a set of vectors of length n such that $t_m(C) = \text{cov}(C, S)$. Have $\mathbf{v}'.i$ be $\mathbf{v}.i$ concatenated with itself r times and $S' = \{\mathbf{v}'.1, \ldots, \mathbf{v}'.m\}$. Then $rt_m(C) = \text{cov}(C', S') \leq t_m(C')$.

For the other inequality, let $S = \{\mathbf{v}.1, \ldots, \mathbf{v}.m\}$ be a set of vectors of length rn with $\mathbf{v}.i = \mathbf{v}.i_1|\mathbf{v}.i_2| \cdots |\mathbf{v}.i_r$ where each $\mathbf{v}.i_j$ has length n. Then from the definition of m-covering radius there is a $\mathbf{c} \in C$ such that $\text{dist}(\mathbf{c}, \mathbf{v}.i_j) \leq t_{rm}(C)$ for every i and j. Thus $\text{dist}(\mathbf{c}|\mathbf{c}| \cdots |\mathbf{c}, \mathbf{v}.i) \leq rt_{rm}(C)$ for every i. Therefore, $t_m(C') \leq rt_{rm}(C)$. □

Note that it is possible for both of the inequalities to be strict. For example, if $C = \{00, 01, 10\}$ and $r = 2$ then $t_2(C) = 1$, $t_2(C') = 3$ and $t_4(C) = 4$.

2.1.3 New Parity Checks

A linear $[n, k]$ code C can be lengthened by adding another parity check. This amounts to adding another column to the code's generator matrix. The new code is a $[n + 1, k]$ code whose codewords consist of the vectors of the form $\mathbf{c}|\mathbf{c} \cdot \mathbf{p}$, where $\mathbf{c}$ is a codeword of the original code C and $\mathbf{p}$ is some fixed vector corresponding to the new parity check. This process is called extending a code and the resulting code is called the extended code.

When $\mathbf{p}$ is the all one vector we say that we have added an overall parity check. In this case, the extended code is formed by adding a 0 at the end of each even weight codeword and a 1 at the end of each odd weight codeword of the original code. Thus all of the codewords in the extended code are even.

For any positive integer m the m-covering radius of the new code is either $t_m(C)$ or $t_m(C)+1$. The following theorem characterizes when the multicovering radius increases.

Theorem 2.7 (Klapper [11]) *Suppose the code C is lengthened to C' by the addition of the parity check $\mathbf{p}$. Then $t_m(C') = 1 + t_m(C)$ if and only if there is a translate leader $S = \{\mathbf{v}.1, \ldots, \mathbf{v}.m\}$ of C with weight $t_m(C)$ and a vector $\mathbf{e} \in \boldsymbol{F}^m$ such that whenever $\mathbf{c} \in C$ and $S + \mathbf{c}$ is a translate leader, we have $e_i \neq \mathbf{c} \cdot \mathbf{p}$ for some i such that the weight of $\mathbf{v}.i + \mathbf{c}$ is maximal.*

Proof: Let $S = \{\mathbf{v}.1, \ldots, \mathbf{v}.m\}$ be a set of binary vectors of length n. We can extend S to be a set of vectors of length $n+1$ in 2^m ways, each of the form $S_{\mathbf{e}} = \{\mathbf{v}.1|e_1, \ldots, \mathbf{v}.m|e_m\}$ for some vector $\mathbf{e} \in \boldsymbol{F}^m$. The weight of each element of $S_{\mathbf{e}}$ can be at most one more than its corresponding element in S. Thus the weight of the translate leader of $S_{\mathbf{e}}$ can be at most one more than the translate leader of S. For a given S and $\mathbf{e}$ the translate of $S_{\mathbf{e}}$ consists of m-tuples of the form

$$\{\mathbf{v}.1 + \mathbf{c}|e_1 + \mathbf{c}\cdot\mathbf{p}, \ldots, \mathbf{v}.m + \mathbf{c}|e_m + \mathbf{c}\cdot\mathbf{p}\},$$

where $\mathbf{c} \in C$ and $\mathbf{p}$ is a parity check. The weight of an element of such an m-tuple is greater than that of $S + \mathbf{c}$ exactly when $e_i \neq \mathbf{c}\cdot\mathbf{p}$ for some i such that $\mathrm{wt}(\mathbf{v}.i + \mathbf{c})$ is maximal and the m-covering radius increases if and only if the weight of every translate leader of some maximal weight translate increases. □

Corollary 2.8 (Klapper [11]) *Appending a zero or overall parity check to a code increases the m-covering radius by one.*

Proof: In the case of a zero parity let $\mathbf{e} = \mathbf{1}^m$. Then e_i is never equal to $\mathbf{c}\cdot\mathbf{0}^n$.

In the case of an overall parity check for any set $S = \{\mathbf{v}.1, \ldots, \mathbf{v}.m\}$ let S_{odd} be the set of odd vectors in S, and let S_{even} be the set of even vectors. Since the maximum weight of a vector in S must be either even or odd, the maximum weight vectors of S must all be in S_{odd} or all be in S_{even}. Suppose S is a maximum weight translate leader and all of the maximum weight vectors of S have even weight. Let $e_i = 0$ if $\mathbf{v}.i \in S_{\text{odd}}$ and $e_i = 1$ if $\mathbf{v}.i \in S_{\text{even}}$. Let $T = \mathbf{c} + S$, where T has the same weight as S. The maximum weight elements of T are in T''. If c has even weight then $T'' = \mathbf{c} + S''$ and $e_i = 1 \neq \mathbf{c}\cdot\mathbf{p}$. If $\mathbf{c}$ has odd weight then $T'' = \mathbf{c} + S'$ and $e_i = 0 \neq \mathbf{c} \cdot \mathbf{p}$. The argument is similar if the maximum weight vectors of S are odd, simply reverse the definition of $\mathbf{e}$. □

2.2 Relative Covering Radius

Definition: Let C and S be codes of length n, and let m be a positive integer. The k-covering radius of C relative to S, $R_k(S, C)$, is the smallest integer r such that for every k-tuple, $\{\mathbf{c}.1, \ldots, \mathbf{c}.k\}$, of elements of C there is an element $\mathbf{s}$ of S such that $\mathrm{dist}(\mathbf{c}.i, \mathbf{s}) \leq r$ for all $i = 1, \ldots, k$. Also, $t_k(m, C) = \min\{R_k(S, C) : |S| = m\}$. That is, we are only covering m-tuples of vectors in C, not all of the Hamming space with vectors in S.

Theorem 2.9 (Honkala and Klapper [10]) *Let C be a code of length n. Then $t_m(C) = n - t_1(m, C)$.*

Proof: Let S be any (n, m) code. Then from the definition of relative covering radius

$$R_1(\overline{S}, C) \geq t_1(m, C), \tag{2.1}$$

where $\overline{S}$ is the set of complements the of elements of S. Furthermore, equation (2.1) holds with equality for at least one such S. Therefore, there is some $\mathbf{c} \in C$ such that for every $\bar{\mathbf{s}} \in \overline{S}$, $\mathrm{dist}(\mathbf{c}, \bar{\mathbf{s}}) \geq t_1(m, C)$. Thus, there is some $\mathbf{c} \in C$ such that for every $\mathbf{s} \in S$, $\mathrm{dist}(\mathbf{c}, \mathbf{s}) \leq n - t_1(m, C)$. As this holds for every (n, m) code S

$$R_m(S, C) \leq n - t_1(m, C).$$

If equation (2.1) holds with equality then for every $\mathbf{c} \in C$ there exists an $\bar{\mathbf{s}} \in \overline{S}$ such that $\mathrm{dist}(\mathbf{c}, \bar{\mathbf{s}}) \leq t_1(m, C)$. In other words, for every $\mathbf{c} \in C$, there exists an $\mathbf{s} \in S$ such that $\mathrm{dist}(\mathbf{c}, \mathbf{s}) \geq n - t_1(m, C)$. Therefore $R_m(S, C) \geq n - t_1(m, C)$ for at least one (n, m) code S. Thus $R_m(C) = n - t_1(m, C)$. □

Theorem 2.9 states that we can find bounds on the m-covering radius of a code by finding bounds on the relative covering radius.

2.3 Lower Bounds

Theorem 2.10 (Klapper [10], [11]) *For every m and n satisfying $m \leq 2^n$, we have $t_m(\boldsymbol{F}^n) \geq \lceil (n + \lfloor \log_2(m) \rfloor - 1)/2 \rceil$, with equality for $m = 2, 3, 4, 5, 6$.*

2.3.1 Sphere Bound

Assume that C is a q-ary (n, K) code with covering radius R. Since each codeword covers $V_q(n, R)$ vectors within a distance of R, it must be that $K \geq q^n / V_q(n, R)$. This inequality is known as the sphere-covering bound, and it generalizes to the multicovering radius case.

Theorem 2.11 (Klapper [11]) *For any (n, K) code C*

$$K \geq \frac{\begin{pmatrix} q^n \\ m \end{pmatrix}}{\begin{pmatrix} V_q(n, t_m(C)) \\ m \end{pmatrix}}.$$

Proof: Since each codeword $\mathbf{c}$ can r-cover only the m-tuples chosen from the ball $B_r(\mathbf{c})$, there is a total of $V_q(n, r)$ choose m such m-tuples. The total number of possible m-tuples is q^m choose m. This gives rise to the above inequality. □

2.3.2 Other Methods

Other methods have been used to improve upon the sphere bound, such as the method of counting excesses and the method of linear inequalities. In the method of counting excesses instead of looking at how the whole space is covered one considers whether a ball of some small radius can be covered perfectly. In other words, can every point in the ball be covered by exactly one codeword? If not then there must already be some "excess".

Given C, a $(n, K)R$ code and vector $\mathbf{x} \in \boldsymbol{F}^n$, let $A_i(\mathbf{x}) \triangleq |\{\mathbf{c} \in C : \text{dist}(\mathbf{c}, \mathbf{x}) = i\}|$. The method of linear inequalities gives bounds on K by deriving linear inequalities on $A_i(\mathbf{0}^n)$ and $A_i(\mathbf{x})$. Both of these methods have been extended to the multicovering radius case by Klapper [11], [13].

Chapter 3

The Multicovering radius of Specific Codes

In this chapter we describe new results on the multicovering radius of specific codes. Currently precise results for the multicovering radius of code are somewhat rare. Even in the case of the Hamming space the only tight results currently known are given by Theorem 2.10 while some bounds are known for $m > 6$. Therefore, we will look at some of the techniques that have been successful in determining the multicovering radius.

Let C be a code with covering radius 1, then Theorems 2.3 and 2.10 give the following bounds:

$$\lceil n/2 \rceil \leq t_m(C) \leq \lceil n/2 \rceil + 1$$

if $m = 2$ or 3 and

$$\lceil n + 1/2 \rceil \leq t_m(C) \leq \lceil n + 1/2 \rceil + 1$$

if $m = 4$, 5, or 6. We would like to know under what conditions equality holds in these bounds. First, various techniques are used to examine the case of the even weight code which has covering radius 1. Then we will take a look at linear codes with 1-covering radius one.

Definition: The even weight code is the code consisting of all the even weight vectors of a fixed length. This code will be denoted by $\boldsymbol{E}^n \triangleq \{\mathbf{x} : \mathbf{x} \in \boldsymbol{F}^n \text{ and } \mathrm{wt}(\mathbf{x}) \text{ is even}\}$.

3.1 The 2-covering radius of the even weight code

In this section we give an upper bound on the 2-covering radius of the even weight code. We then prove that this bound is tight by constructing appropriate deep holes (m-tuples that meet the upper bound).

Lemma 3.1 *For any n, $t_2(\boldsymbol{E}^n) \leq \lceil (n+1)/2 \rceil$.*

Proof: Let $\mathbf{a}, \mathbf{b} \in \boldsymbol{F}^n$. Assume that $\mathbf{a}$ or $\mathbf{b}$ has even weight. Without loss of generality, let $\mathbf{a}$ have even weight. We have $\mathrm{wt}(\mathbf{a}+\mathbf{x}) = \mathrm{wt}(\mathbf{a}) + \mathrm{wt}(\mathbf{x}) - 2\ |\mathrm{supp}(\mathbf{a}) \cap \mathrm{supp}(\mathbf{x})|$, so $\mathbf{a}+\mathbf{x} \in \boldsymbol{E}^n$ whenever $\mathbf{a}, \mathbf{x} \in \boldsymbol{E}^n$. Also, distance is preserved under translation. Therefore, there is an $\mathbf{x}$ in $\boldsymbol{E}^n$ such that $\mathrm{cov}(\mathbf{x}, \{\mathbf{a}, \mathbf{b}\}) = d$ if and only if there is a $\mathbf{y}$ in $\boldsymbol{E}^n$ such that $\mathrm{cov}(\mathbf{y}, \{\mathbf{0}^n, \mathbf{c}\}) = d$; i.e. where $\mathbf{c} = \mathbf{a}+\mathbf{b}$ (set $\mathbf{y} = \mathbf{a}+\mathbf{x}$). Therefore, we only need to consider coverings of $\{\mathbf{0}^n, \mathbf{c}\}$.

Let $r = \lfloor \mathrm{wt}(\mathbf{c})/2 \rfloor$ or $\lfloor (\mathrm{wt}(\mathbf{c})+2)/2 \rfloor$, whichever is even. Let $\mathbf{y}$ be a vector with ones in r coordinates in which $\mathbf{c}$ has ones, and zeros elsewhere.

Suppose $\mathbf{y}$ has $\lfloor \mathrm{wt}(\mathbf{c})/2 \rfloor$ ones. Then, since $\mathrm{wt}(\mathbf{c}) \leq n$,

$$\mathrm{dist}(\mathbf{0}^n, \mathbf{y}) = \left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor \leq \left\lfloor \frac{n}{2} \right\rfloor.$$

Also, as $\mathrm{wt}(\mathbf{c}) \leq n$,

$$\mathrm{dist}(\mathbf{c}, \mathbf{y}) = \mathrm{wt}(\mathbf{c}) - \left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor = \left\lceil \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rceil \leq \left\lceil \frac{n}{2} \right\rceil.$$

Suppose $\mathbf{y}$ has $\lfloor (\mathrm{wt}(\mathbf{c})+2)/2 \rfloor$ ones. Then

$$\mathrm{dist}(\mathbf{0}^n, \mathbf{y}) = \left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor + 1 \leq \left\lfloor \frac{n}{2} \right\rfloor + 1 = \left\lceil \frac{n+1}{2} \right\rceil.$$

Also,

$$\mathrm{dist}(\mathbf{c}, \mathbf{y}) = \mathrm{wt}(\mathbf{c}) - \left(\left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor + 1 \right) = \left\lceil \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rceil - 1 \leq \left\lceil \frac{n}{2} \right\rceil - 1.$$

Now assume $\mathbf{a}$ and $\mathbf{b}$ both have odd weight. Let $\mathbf{a} = \mathbf{a}'|\mathbf{a}''$, where $\mathbf{a}' \in \boldsymbol{F}^1$ and $\mathbf{a}'' \in \boldsymbol{F}^{n-1}$. Then by translating each vector by $0|\mathbf{a}''$ or $1|\mathbf{a}''$, whichever has even weight, we see that there is an $\mathbf{x}$ in $\boldsymbol{E}^n$ such that $\mathrm{cov}(\mathbf{x}, \{\mathbf{a}, \mathbf{b}\}) = d$ if and only if there is a $\mathbf{y}$ in $\boldsymbol{E}^n$ such that $\mathrm{cov}(\mathbf{y}, \{1|\mathbf{0}^{n-1}, \mathbf{c}\}) = d$.

Let $r = \lfloor \mathrm{wt}(\mathbf{c})/2 \rfloor$ or $\lceil \mathrm{wt}(\mathbf{c})/2 \rceil$, whichever is even. Let $\mathbf{y}$ be a vector with ones in r coordinates in which $\mathbf{c}$ has ones, and zeros elsewhere.

Suppose $\mathbf{y}$ has $\lfloor \mathrm{wt}(\mathbf{c})/2 \rfloor$ ones and $c_1 = 1$. Then

$$\mathrm{dist}(1|\mathbf{0}^{n-1}, y) = \left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor - 1 \leq \left\lfloor \frac{n}{2} \right\rfloor - 1.$$

If $c_1 = 0$, then

$$\mathrm{dist}(1|\mathbf{0}^{n-1}, \mathbf{y}) = \left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor + 1 \leq \left\lfloor \frac{n-1}{2} \right\rfloor + 1 = \left\lceil \frac{n}{2} \right\rceil.$$

Also,

$$\mathrm{dist}(\mathbf{c}, \mathbf{y}) = \mathrm{wt}(\mathbf{c}) - \left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor = \left\lceil \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rceil \leq \left\lceil \frac{n}{2} \right\rceil.$$

Suppose $\mathbf{y}$ has $\lceil \mathrm{wt}(\mathbf{c})/2 \rceil$ ones and $c_1 = 1$. Then

$$\mathrm{dist}(1|\mathbf{0}^{n-1}, \mathbf{y}) = \left\lceil \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rceil - 1 \leq \left\lceil \frac{n}{2} \right\rceil - 1.$$

If $c_1 = 0$, then

$$\mathrm{dist}(1|\mathbf{0}^{n-1}, \mathbf{y}) = \left\lceil \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rceil + 1 \leq \left\lceil \frac{n-1}{2} \right\rceil + 1 = \left\lceil \frac{n+1}{2} \right\rceil.$$

Also,

$$\mathrm{dist}(\mathbf{c}, \mathbf{y}) = \mathrm{wt}(\mathbf{c}) - \left\lceil \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rceil = \left\lfloor \frac{\mathrm{wt}(\mathbf{c})}{2} \right\rfloor \leq \left\lfloor \frac{n}{2} \right\rfloor.$$

So in all possible cases $\mathrm{cov}(\boldsymbol{E}^n, \{\mathbf{a}, \mathbf{b}\}) \leq \lceil (n+1)/2 \rceil$. □

Theorem 3.2 *For any $n \geq 1$, $t_2(\boldsymbol{E}^n) = \lceil (n+1)/2 \rceil$.*

Proof: From Lemma 3.1 and Theorem 2.10 we know that $\lceil n/2 \rceil \leq t_2(\boldsymbol{E}^n) \leq \lceil (n+1)/2 \rceil$. However, in the case that n is odd, $\lceil n/2 \rceil = \lceil (n+1)/2 \rceil$. So all that remains is the case when the length is even. We now construct deep holes that meet the bound given in Lemma 3.1. There are two cases.

Suppose $n \equiv 2 \mod 4$. Our deep hole is $\{\mathbf{0}^n, \mathbf{1}^n\}$. Let $\mathbf{z} \in \boldsymbol{E}^n$. Then $\mathrm{wt}(\mathbf{z}) = 2i$ and $n = 2 + 4j$ for some natural numbers i and j. Let

$$f(i) = \mathrm{dist}(\mathbf{0}^n, \mathbf{z}) = 2i$$

and

$$g(i) = \mathrm{dist}(\mathbf{1}^n, \mathbf{z}) = n - 2i = 2 + 4j - 2i.$$

Both functions are linear in i. Given the above we can see that

$$\operatorname{cov}(\boldsymbol{E}^n, \{\mathbf{0}^n, \mathbf{1}^n\}) = \min_{0\leq i\leq 1+2j} \max(f(i), g(i))$$

from the definition of $\operatorname{cov}(\boldsymbol{E}^n, \{\mathbf{0}^n, \mathbf{1}^n\})$. If i was not restricted to the natural numbers, then this minimum would occur when $f(i) = g(i)$ (when $i = \frac{1}{2} + j$). Over the natural numbers, the minimum occurs for either $i = j$ or $i = 1 + j$. Thus

$$\begin{aligned}\min_{0\leq i\leq 1+2j} \max(f(i), g(i)) &= \max(2 + 2j, 2j) \\ &= \frac{n}{2} + 1 = \left\lceil \frac{n+1}{2} \right\rceil.\end{aligned}$$

Suppose $n \equiv 0 \mod 4$. We claim that $\{1|\mathbf{0}^{n-1}, 0|\mathbf{1}^{n-1}\}$ is a deep hole that meets the given bound. Let $\mathbf{z} \in \boldsymbol{E}^n$. Then $\operatorname{wt}(\mathbf{z}) = 2i$ and $n = 4j$ for some natural numbers i and j. Let

$$f(i) = \operatorname{dist}(1|\mathbf{0}^{n-1}, \mathbf{z}) = \begin{cases} 2i - 1 & \text{if } z_1 = 1 \\ 2i + 1 & \text{if } z_1 = 0 \end{cases}$$

and

$$g(i) = \operatorname{dist}(0|\mathbf{1}^{n-1}, \mathbf{z}) = n - 2i = \begin{cases} 4j - 2i + 1 & \text{if } z_1 = 1 \\ 4j - 2i - 1 & \text{if } z_1 = 0 \end{cases}.$$

Again both functions are linear in i and:

$$\operatorname{cov}(\boldsymbol{E}^n, \{\mathbf{0}^n, \mathbf{1}^n\}) = \min_{0\leq i\leq 2j} \max(f(i), g(i)).$$

If i was not restricted to the natural numbers, then this minimum would occur when $f(i) = g(i)$. This is when $i = \frac{1}{2} + j$ if $z_1 = 1$ or when $i = j - \frac{1}{2}$ if $z_1 = 0$. Over the natural numbers, the minimum occurs for either $i = j$ or $i = 1 + j$ when z_1, or either $i = j$ or $i = j - 1$ when $z_1 = 0$. In either case

$$\begin{aligned}\min_{0\leq i\leq 2j} \max(f(i), g(i)) &= \max(2j + 1, 2j - 1) \\ &= \frac{n}{2} + 1 = \left\lceil \frac{n+1}{2} \right\rceil.\end{aligned}$$

□

3.2 The 3-covering radius of the even weight code

We now form bounds on the 3-covering radius of the even weight code. The case when the length of the code is even is already known from the results of the last two sections.

Therefore, we need only tighten our bounds for the case when the length is odd. To do this we develop bounds on the relative covering radius as in Honkala and Klapper's paper [10].

Lemma 3.3 (Honkala and Klapper [10]) *A binary code of odd length n, cardinality three and covering radius $(n-1)/2$ contains a word-complement pair.*

Lemma 3.4 *For n odd and $n \geq 3$, $t_1(3, \boldsymbol{E}^n) = (n-1)/2$.*

Proof: Assume that there is a code $C = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ of length n such that C covers $\boldsymbol{E}^n$ with balls of radius $(n-3)/2$. As the 1-covering radius of $\boldsymbol{E}^n$ is one, C covers the whole space with balls of radius $(n-1)/2$. Therefore, by Lemma 3.3 C contains a word-complement pair. Without loss of generality, let $\mathbf{b} = \overline{\mathbf{c}}$. Furthermore, the 2-covering radius of $\boldsymbol{E}^n$ is $(n+1)/2$. So there exists $\mathbf{e} \in \boldsymbol{E}^n$ such that $\operatorname{dist}(\mathbf{e}, \mathbf{b})$ and $\operatorname{dist}(\mathbf{e}, \mathbf{c})$ are $(n-1)/2$ and $(n+1)/2$ in some order. Also, $\operatorname{dist}(\mathbf{e}, \mathbf{b}) \neq \operatorname{dist}(\mathbf{e}, \mathbf{c})$ by the triangle inequality. So we may let $\operatorname{dist}(\mathbf{e}, \mathbf{b}) = (n+1)/2$ and i be a coordinate where $\overline{\mathbf{e}}$ and $\mathbf{c}$ are different and where $\overline{\mathbf{e}}$ and $\mathbf{b}$ are the same. While $\overline{\mathbf{e}}$ does not have even weight, as the length of our code is odd, we can find an element of $\boldsymbol{E}^n$ that is close. Let $\mathbf{e}'$ be the vector $\overline{\mathbf{e}}$ with the i^{th} bit complemented. Then $\mathbf{e}'$ is an element of $\boldsymbol{E}^n$,

$$\begin{aligned} \operatorname{dist}(\mathbf{e}', \mathbf{b}) &= \operatorname{dist}(\overline{\mathbf{e}}, \mathbf{b}) + 1 \\ &= \frac{n-1}{2} + 1 \end{aligned}$$

and

$$\begin{aligned} \operatorname{dist}(\mathbf{e}', \mathbf{c}) &= \operatorname{dist}(\overline{\mathbf{e}}, \mathbf{c}) - 1 \\ &= \frac{n+1}{2} - 1. \end{aligned}$$

So neither $\mathbf{e}$ nor $\mathbf{e}'$ is an element of the ball $B_{(n-3)/2}(\mathbf{b})$ or the ball $B_{(n-3)/2}(\mathbf{c})$. Since $\operatorname{dist}(\mathbf{e}, \mathbf{e}') = n-1$, $\mathbf{e}$ and $\mathbf{e}'$ cannot both belong to $B_{(n-3)/2}(\mathbf{a})$. This contradicts our assumption so $t_1(3, \boldsymbol{E}^n) \geq (n-1)/2$. The reverse inequality comes from the bounds on $t_3(\boldsymbol{E}^n)$ which we have already established and from Theorem 2.9. □

From Lemma 3.4, Theorem 2.9, and our previous bounds, we obtain the following result.

Theorem 3.5 *For all $n \geq 2$, $t_3(\boldsymbol{E}^n) = \lceil (n+1)/2 \rceil$.*

3.3 Higher order covering radii of the even weight code

Lemma 3.6 *If n is odd and $n \geq 3$ then $t_4(\boldsymbol{E}^n) = \lceil (n+1)/2 \rceil + 1$.*

Proof: There are two cases:

Suppose $n \equiv 1 \mod 4$. There exists an integer k such that $n = 1 + 4k$. Let

$$S = \{\mathbf{0}^{n-2}|\mathbf{0}^2, \mathbf{0}^{n-2}|\mathbf{1}^2, \mathbf{1}^{n-2}|\mathbf{0}|\mathbf{1}, \mathbf{1}^{n-2}|\mathbf{1}|\mathbf{0}\}.$$

Also assume that there exists an even weight vector $\mathbf{v}$ of length n such that $\text{cov}(\mathbf{v}, S) = \lceil (n+1)/2 \rceil = 2k+1$. Let $\mathbf{v} = \mathbf{v}'|\mathbf{v}''$, where $\mathbf{v}' \in \boldsymbol{F}^{n-2}$ and $\mathbf{v}'' \in \boldsymbol{F}^2$. So $\text{dist}(\mathbf{v}', \mathbf{0}^{n-2}) \leq 2k+1$ and $\text{dist}(\mathbf{v}', \mathbf{1}^{n-2}) \leq 2k+1$. This implies that $2k - 2 \leq \text{wt}(\mathbf{v}') \leq 2k+1$. The following table examines each of these possible cases.

$\text{wt}(\mathbf{v}') = 2k-2$	$\text{wt}(\mathbf{v}') = 2k-1$	$\text{wt}(\mathbf{v}') = 2k$	$\text{wt}(\mathbf{v}') = 2k+1$
$\text{wt}(\mathbf{v}'') = 0$ or 2	$\text{wt}(\mathbf{v}'') = 1$	$\text{wt}(\mathbf{v}'') = 0$ or 2	$\text{wt}(\mathbf{v}'') =1$
$\mathbf{1}^{n-2}|\mathbf{0}|\mathbf{1}$	$\mathbf{1}^{n-2}|\mathbf{0}|\mathbf{1}$	$\mathbf{0}^{n-2}|\mathbf{0}^2$	$\mathbf{0}^{n-2}|\mathbf{0}^2$
$\mathbf{1}^{n-2}|\mathbf{1}|\mathbf{0}$	$\mathbf{1}^{n-2}|\mathbf{1}|\mathbf{0}$	$\mathbf{0}^{n-2}|\mathbf{1}^2$	$\mathbf{0}^{n-2}|\mathbf{1}^2$

In each of the above cases, one of the listed vectors has distance $2k+2$ to $\mathbf{v}$. This contradicts the assumption that $\text{cov}(\mathbf{v}, S) = \lceil (n+1)/2 \rceil = 2k+1$.

Suppose $n \equiv 3 \mod 4$.

This case is similar, but let

$$S = \{\mathbf{0}^{n-2}|\mathbf{0}|\mathbf{1}, \mathbf{0}^{n-2}|\mathbf{1}|\mathbf{0}, \mathbf{1}^{n-2}|\mathbf{0}^2, \mathbf{1}^{n-2}|\mathbf{1}^2\}.$$

Also there exists an integer k such that $n = 3+4k$ and we can assume that there exists an even weight vector $\mathbf{v}$ of length n such that $\text{cov}(\mathbf{v}, S) = \lceil (n+1)/2 \rceil = 2k+2$. Let $\mathbf{v} = \mathbf{v}'|\mathbf{v}''$, where $\mathbf{v}' \in \boldsymbol{F}^{n-2}$ and $\mathbf{v}'' \in \boldsymbol{F}^2$. So $\text{dist}(\mathbf{v}', \mathbf{0}^{n-2}) \leq 2k+2$ and $\text{dist}(\mathbf{v}', \mathbf{1}^{n-2}) \leq 2k+2$. This implies that $2k-1 \leq \text{wt}(\mathbf{v}') \leq 2k+2$. The following table examines each of these possible cases.

$\mathrm{wt}(\mathbf{v}') = 2k-1$	$\mathrm{wt}(\mathbf{v}') = 2k$	$\mathrm{wt}(\mathbf{v}') = 2k+1$	$\mathrm{wt}(\mathbf{v}') = 2k+2$
$\mathrm{wt}(\mathbf{v}'') = 1$	$\mathrm{wt}(\mathbf{v}'') = 0$ or 2	$\mathrm{wt}(\mathbf{v}'') = 1$	$\mathrm{wt}(\mathbf{v}'') = 0$ or 2
$\mathbf{1}^{n-2}\vert\mathbf{0}^2$	$\mathbf{1}^{n-2}\vert\mathbf{0}^2$	$\mathbf{0}^{n-2}\vert\mathbf{0}\vert\mathbf{1}$	$\mathbf{0}^{n-2}\vert\mathbf{0}\vert\mathbf{1}$
$\mathbf{1}^{n-2}\vert\mathbf{1}^2$	$\mathbf{1}^{n-2}\vert\mathbf{1}^2$	$\mathbf{0}^{n-2}\vert\mathbf{1}\vert\mathbf{0}$	$\mathbf{0}^{n-2}\vert\mathbf{1}\vert\mathbf{0}$

In each of the above cases, one of the listed vectors has distance $2k+3$ to $\mathbf{v}$. This contradicts the assumption that $\mathrm{cov}(\mathbf{v}, S) = \lceil (n+1)/2 \rceil = 2k+2$.

Therefore, $t_4(\boldsymbol{E}^n) \geq \lceil n + 1/2 \rceil + 1$. The reverse inequality comes from our previous bounds. □

Lemma 3.6 and our previous bounds yield the following.

Theorem 3.7 *If n is odd, $n \geq 3$ and $m = 4, 5$, or 6, then $t_m(\boldsymbol{E}^n) = \lceil (n+1)/2 \rceil + 1$.*

In the case of even length we again use the relative covering radius, and the following definitions.

Definition: If each of the four possible pairs 00, 01, 10 and 11 occur at least once in every two coordinates of the codewords, we call the code 2-independent.

Definition: Let $K_{\text{even}}(n, r)$ denote the minimum number of codewords in any code C of length n such that $t_1(C, \boldsymbol{E}^n) = r$. Similarly let $K_{\text{odd}}(n, r)$ denote the minimum number of codewords in any code C of length n such that $t_1(C, \{\mathbf{x} \in \boldsymbol{F}^n : \mathrm{wt}(\mathbf{x}) \text{ is odd}\}) = r$.

Lemma 3.8 *$K_{even}(n, r) = K_{odd}(n, r)$.*

Proof: Let C be a (n, k) code that r covers $\boldsymbol{E}^n$, and $\mathbf{x}$ be any odd weight vector. Then

$$\mathrm{dist}(\mathbf{x}, \mathbf{1}|\mathbf{0}^{n-1} + C) = \mathrm{dist}(\mathbf{1}|\mathbf{0}^{n-1} + \mathbf{x}, C) \leq r.$$

So $\mathbf{1}|\mathbf{0}^{n-1} + C$ covers all of the odd weight vectors with balls of radius r. Thus $K_{\text{even}}(n, r) \geq K_{\text{odd}}(n, r)$.

The reverse inequality can be shown in a similar fashion. Let $\mathbf{x}$ be any even weight vector and C be a (n, k) code that r covers the set of all odd vectors of length n. Then $\mathbf{1}|\mathbf{0}^{n-1} + C$ covers all of the even weight vectors with balls of radius r. Thus $K_{\text{even}}(n, r) \leq K_{\text{odd}}(n, r)$.

Therefore, $K_{\text{even}}(n,r) = K_{\text{odd}}(n,r)$. □

Definition: For an arbitrary set S of cardinality n, we call two subsets A and B of S 2-independent if membership (non-membership) in one neither implies nor excludes membership (non-membership) in the other. Thus

$$A \cap B, A \cap \overline{B}, \overline{A} \cap B, \text{ and } \overline{A} \cap \overline{B}$$

are all not empty. A collection of subsets of S is said to be 2-independent if every pair in the collection is 2-independent.

Theorem 3.9 (Kleitman and Spencer [14]) *The maximal size of a 2-independent collection of subsets of an n element set S is* $\begin{pmatrix} n-1 \\ \lfloor n/2 \rfloor - 1 \end{pmatrix}$.

Lemma 3.10 $K_{even}(2R+4, R) \geq 6$.

Proof: We prove this by induction on R. First suppose $R = 0$. Then each vector can only cover itself. Thus $K_{\text{even}}(4,0) \geq 8$.

Now assume by induction that $K_{\text{even}}(2i+2, i-1) \geq 6$ for some $i \geq 1$. Further assume that there exists a $(2i+4, 5)$ code C that covers $\boldsymbol{E}^{2i+4}$ with balls of radius i. By Theorem 3.9, C is not 2-independent.

Suppose the pair 00 or the pair 11 does not appear in two of the coordinates of C. Without loss of generality we may assume 00 does not appear in the first two coordinates. Let C' be the code C punctured on these coordinates. If there exists any even weight vector $\mathbf{x}$, such that $\text{dist}(\mathbf{x}, \mathbf{C'}) > i-1$ then $\text{dist}(\mathbf{00}|\mathbf{x}, C) > i$. However, this contradicts the fact that C covers the even weight vectors with radius i. Thus C' covers the even weight vectors with radius at most $i-1$. But this now contradicts the fact that $K_{\text{even}}(2i+2, i-1) \geq 6$. Therefore, no such code C can exist and $K_{\text{even}}(2i+4, i) \geq 6$.

The case when 10 or 01 does not appear in two of the coordinates of C is similar. The code $\mathbf{1}|\mathbf{0}^{2i+3} + C$ covers the odd vectors with radius i, the pair 00 or the pair 11 does not appear in $\mathbf{1}|\mathbf{0}^{2i+3} + C$. $K_{\text{even}}(2i+2, i-1) \geq 6$ implies that $K_{\text{odd}}(2i+2, i-1) \geq 6$ by Lemma 3.8 and the remaining argument is the same as above. □

Theorem 3.11 *If n is even, $n \geq 2$, and $m = 4$ or 5 then $t_m(\boldsymbol{E}^n) = \lceil (n+1)/2 \rceil$.*

m	n	$t_m(\boldsymbol{E}^n)$
$2, 3$	—	$\lceil (n+1)/2 \rceil$
$4, 5, 6$	odd	$\lceil (n+1)/2 \rceil + 1$
$4, 5$	even	$\lceil (n+1)/2 \rceil$
6	even	$\lceil (n+1)/2 \rceil \leq t_m(\boldsymbol{E}^n) \leq \lceil (n+1)/2 \rceil + 1$

Table 3.1: Multicovering radius of the even weight code.

Proof: Given that n is even, $n \geq 2$ and $m = 4$ or 5, our previous bounds show that $t_1(m, \boldsymbol{E}^n)$ is $(n-2)/2$ or $(n-4)/2$. However, Lemma 3.10 shows that $t_1(m, \boldsymbol{E}^n) \neq (n-4)/2$. □

Table 3.1 shows our results on the multicovering radius of the even weight code. Our methods have illustrated some of the techniques that are available to determine multicovering radii. After this work was done Honkala showed that $t_m(\boldsymbol{E}^n) = t_m(\boldsymbol{F}^{n-1}) + 1$ using some facts previously proven by Klapper.

Theorem 3.12 (Honkala [9]) $t_m(\boldsymbol{E}^n) = t_m(\boldsymbol{F}^{n-1}) + 1$.

Proof: This can be seen by using Corollary 2.8 and noting that the even weight code of length n is equal to the Hamming space of length $n-1$ appended with a parity check. □

However, this method is unlikely to work for more general codes of 1-covering radius one as it relies on the fact that the code can be constructed by appending zero or overall parity checks to the Hamming space of some length. However, much of the proof of Lemma 3.1 still holds for any linear code with 1-covering radius one. In the case when the length of the code is odd it only remains to show that any such linear code must contain a vector of weight $\lfloor n/2 \rfloor$ or $\lceil n/2 \rceil$. We will see how to complete this proof in this case in Section 3.4.

3.4 The Multicovering Radii of Linear Codes with Covering Radius One

Here we prove two lemmas on the structure of linear codes of odd length with 1-covering radius one and show that such linear codes have 2-covering radius $(n+1)/2$. Note that non-linear codes with covering radius one may have 2-covering radius $\lceil n/2 \rceil + 1$. The code consisting of all vectors except those of weight $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$ is an example of such a code. This code cannot cover $\{\mathbf{0}^n, \mathbf{1}^n\}$ within radius $\lceil n/2 \rceil$.

Lemma 3.13 *Let C be a linear code of length n and 1-covering radius one, where n is odd and greater than 3. Then C contains a code word of weight $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$.*

Proof: Assume that such a code C does not contain a codeword of weight $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$. Then every vector of weight $\lfloor n/2 \rfloor$ is covered by codewords of weight $\lfloor n/2 \rfloor - 1 = (n-3)/2$.

Suppose that $n \equiv 1 \mod 4$. To make it easier to compare the supports of vectors we will write them in the form $\mathbf{v} = \mathbf{v}_1|\mathbf{v}_2|\mathbf{v}_3|\mathbf{v}_4|\mathbf{v}'$ where each $\mathbf{v}_i$ has length $(n-1)/4$ and $\mathbf{v}' \in \{0,1\}$. The vector $\mathbf{1}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{0}$ is covered by a codeword α. Without loss of generality let $\alpha = \mathbf{1}^{(n-1)/4}|\mathbf{1}^{(n-5)/4}\mathbf{0}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{0}$.

The vector $\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{0}$ is covered by a codeword of the form $\beta = \mathbf{a}|\mathbf{0}^{(n-1)/4}|\mathbf{b}|\mathbf{0}^{(n-1)/4}|\mathbf{0}$. If the weight of $\mathbf{a}$ is $(n-5)/4$, then the cardinality of the intersection of the supports of α and β is $(n-5)/4$. Therefore,

$$\text{wt}(\alpha + \beta) = n - 3 - \frac{n-5}{2} = \frac{n-1}{2},$$

which contradicts our assumption that there are no weight $\lfloor n/2 \rfloor$ codewords. Thus, the weight of $\mathbf{b}$ is $(n-5)/4$, and the weight of $\mathbf{a}$ is $(n-1)/4$.

Similarly the vector $\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{0}$ is covered by a codeword of the form $\gamma = \mathbf{c}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{d}|\mathbf{0}$, where the weight of d is $(n-5)/4$. The vector $\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{0}$ is covered by a codeword of the form $\delta = \mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{e}|\mathbf{f}|\mathbf{0}$. If the weight of $\mathbf{e}$ is $(n-1)/4$ then $\beta + \delta = \mathbf{a}|\mathbf{0}^{(n-1)/4}|\overline{\mathbf{b}}|\mathbf{f}|\mathbf{0}$, which has weight $(n-1)/4 + 1 + (n-5)/4 = (n-1)/2$. If the weight of $\mathbf{f}$ is $(n-1)/4$ then $\gamma + \delta = \mathbf{c}|\mathbf{0}^{(n-1)/4}|\mathbf{e}|\overline{\mathbf{d}}|\mathbf{0}$, which also has weight $(n-1)/2$. Thus, in all cases we contradict our assumption that C does not contain a codeword of weight $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$.

Suppose that $n \equiv 3 \mod 4$. This time we will break our vectors into blocks of size $(n-3)/4$ and will write them in the form $\mathbf{v} = \mathbf{v}_1|\mathbf{v}_2|\mathbf{v}_3|\mathbf{v}_4|\mathbf{000}$ where $\mathbf{v}_i$ has length $(n-3)/4$. There must be at least one codeword α with weight $(n-3)/2$. Without loss of generality let $\alpha = \mathbf{1}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}\mathbf{0}|\mathbf{0}^{(n-3)/4}|\mathbf{000}$.

The vector $\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{100}$ is covered by a codeword of the form $\beta = \mathbf{a}|\mathbf{0}^{(n-3)/4}|\mathbf{b}|\mathbf{0}^{(n-3)/4}|\mathbf{100}$. We may assume that β ends in $\mathbf{100}$ since if it did not we could permute the coordinates, exchange the $(n-2)^{\text{th}}$ coordinate with a coordinate in the third block $\mathbf{b}$, to place it in this form without changing the structure of α. If the weight of $\mathbf{a}$ is $(n-7)/4$, then the cardinality of the intersection of the supports of α and β is $(n-7)/4$. Therefore,

$$\mathrm{wt}(\alpha+\beta) = n-3-\frac{n-7}{2} = \frac{n+1}{2},$$

which contradicts our assumption that there are no weight $\lceil n/2 \rceil$ vectors. Thus, the weight of $\mathbf{b}$ is $(n-7)/4$, and the weight of $\mathbf{a}$ is $(n-3)/4$.

Similarly the vector $\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{010}$ is covered by a codeword of the form $\gamma = \mathbf{c}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{d}|\mathbf{010}$. We may assume that γ ends in $\mathbf{010}$ since if it did not we could permute the coordinates, exchange the $(n-1)^{\text{th}}$ coordinate with a coordinate in the fourth block $\mathbf{d}$, to place it in this form without changing the structure of α or β. Also the weight of $\mathbf{d}$ is $(n-7)/4$. The vector $\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{001}$ is covered by a codeword of the form $\delta = \mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{e}|\mathbf{f}|\mathbf{000}$. We may assume that δ ends in $\mathbf{000}$ since if it did not we could permute the coordinates, exchange the n^{th} coordinate with the coordinate in the third block where $\mathbf{b}$ is zero, to place it in this form without changing the structure of α, β, or γ. The weight of $\mathbf{e}$ and $\mathbf{f}$ is then $(n-3)/4$. $\beta+\delta = \mathbf{a}|\mathbf{0}^{(n-3)/4}|\overline{\mathbf{b}}|\mathbf{f}|\mathbf{100}$, which has weight $(n-3)/4+1+(n-3)/4+1 = (n+1)/2$. Thus in all cases we contradict our assumption that C does not contain a codeword of weight $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$. □

The proof of the following lemma is analogous to the proof of Lemma 3.13.

Lemma 3.14 *Let C be a linear code of length n and 1-covering radius one, where n is odd and greater than 3. Then C contains a codeword at distance $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$ from the vector* $\mathbf{1}|\mathbf{0}^{n-1}$.

Proof: Assume C does not contain a codeword at distance $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$ from the vector $\mathbf{a} = \mathbf{1}|\mathbf{0}^{n-1}$. Then C contains no codewords of the form $\mathbf{0}|\mathbf{v}_1$, $\mathbf{0}|\mathbf{v}_2$, $\mathbf{1}|\mathbf{v}_2$, or $\mathbf{1}|\mathbf{v}_3$ where $\mathbf{v}_i$ has length $n-1$, $\mathbf{v}_1$ has weight $(n-3)/2$, $\mathbf{v}_2$ has weight $(n-1)/2$, and $\mathbf{v}_3$ has weight $(n+1)/2$.

Suppose that $n \equiv 1 \mod 4$. The vector $\mathbf{1}|\mathbf{1}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}$ is covered by a codeword α. Let $\alpha = \mathbf{1}|\mathbf{a}|\mathbf{b}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}$, where $\mathrm{wt}(\mathbf{ab}) = (n-3)/2$. Note that α must have this structure as otherwise it will be in one of the prohibited forms. Without loss of generality let the weight of $\mathbf{a}$ be $(n-5)/4$.

The vector $\mathbf{1}|\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}$ is covered by a codeword of the form $\beta = \mathbf{1}|\mathbf{c}|\mathbf{0}^{(n-1)/4}|\mathbf{d}|\mathbf{0}^{(n-1)/4}|\mathbf{0}$, where $\mathrm{wt}(\mathbf{cd}) = (n-3)/2$. As before β must have this structure to avoid the prohibited forms. If the weight of $\mathbf{d}$ is $(n-5)/4$, then $\alpha+\beta = \mathbf{0}|\overline{\mathbf{a}}|\mathbf{b}|\mathbf{d}|\mathbf{0}^{(n-1)/4}$, which has weight $1+(n-1)/4+(n-5)/4 = (n-1)/2$. This contradicts our assumption that C does not contain a codeword of the form $\mathbf{0}|\mathbf{v}_2$, where $\mathbf{v}_2$ has weight $(n-1)/2$. Thus, the weight of $\mathbf{c}$ is $(n-5)/4$, and the weight of $\mathbf{d}$ is $(n-1)/4$.

Similarly the vector $\mathbf{1}|\mathbf{1}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}$ is covered by a codeword of the form $\gamma = \mathbf{0}|\mathbf{e}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{f}$, where the weight of $\mathbf{e}$ is $(n-5)/4$. The vector $\mathbf{1}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}|\mathbf{1}^{(n-1)/4}$ must be covered by a codeword of the form $\delta = \mathbf{1}|\mathbf{0}^{(n-1)/4}|\mathbf{0}^{(n-1)/4}|\mathbf{g}|\mathbf{h}$. If the weight of $\mathbf{g}$ is $(n-5)/4$ then $\beta+\delta = \mathbf{0}|\mathbf{c}|\mathbf{0}^{(n-1)/4}|\overline{\mathbf{g}}|\mathbf{h}$, which has weight $(n-5)/4+1+(n-1)/4 = (n-1)/2$. If the weight of $\mathbf{h}$ is $(n-5)/4$ then $\gamma+\delta = \mathbf{0}|\mathbf{e}|\mathbf{0}^{(n-1)/4}|\mathbf{g}|\overline{\mathbf{h}}$, which also has weight $(n-1)/2$. Thus in all cases we contradict our assumption that C does not contain a codeword at distance $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$ from the vector $\mathbf{1}|\mathbf{0}^{n-1}$.

Suppose that $n \equiv 3 \mod 4$. The vector $\mathbf{u} = 1|\mathbf{1}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{10}$ is covered by a codeword. Because we have assumed that there are no codewords of the form $\mathbf{0}|\mathbf{v}_1$ or $\mathbf{1}|\mathbf{v}_2$, where $\mathbf{v}_1$ has weight $(n-3)/2$ and $\mathbf{v}_2$ has weight $(n-1)/2$, we cannot cover $\mathbf{u}$ by complementing its first coordinate or one of the coordinates that contains a zero. Thus we may assume that $\mathbf{u}$ is covered by the codeword $\alpha = \mathbf{1}|\mathbf{1}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{00}$.

The vector $\mathbf{1}|\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{10}$ is covered by a codeword of the form $\beta = \mathbf{1}|\mathbf{a}|\mathbf{0}^{(n-3)/4}|\mathbf{b}|\mathbf{0}^{(n-3)/4}|\mathbf{c0}$. If $\mathbf{c}$ is zero then the weight of $\mathbf{a}$ and $\mathbf{b}$ must both equal $(n-3)/4$. So $\alpha+\beta = \mathbf{0}|\mathbf{1}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{00}$, which is prohibited. If the weight of $\mathbf{b}$ is $(n-5)/4$ then $\alpha+\beta = \mathbf{0}|\mathbf{0}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{b}|\mathbf{0}^{(n-3)/4}|\mathbf{10}$, which has weight $(n-3)/4+(n-5)/4+1 = (n-3)/2$, and is thus prohibited. So the weight of $\mathbf{a}$ is $(n-5)/4$ and the weight of $\mathbf{b}$ is $(n-3)/4$.

Similarly the vector $\mathbf{1}|\mathbf{1}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{10}$ is covered by a codeword of the form $\gamma = \mathbf{1}|\mathbf{d}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{e}|\mathbf{10}$, where the weight of $\mathbf{d}$ is $(n-5)/4$. The vector $\mathbf{1}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{1}^{(n-3)/4}|\mathbf{10}$ is covered by a codeword of the form $\delta = \mathbf{1}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{f}|\mathbf{g}|\mathbf{h}0$. If $\mathbf{h}$ is zero then the weight of $\mathbf{f}$ and $\mathbf{g}$ must both equal $(n-3)/4$. So $\beta+\delta = \mathbf{0}|\mathbf{a}|\mathbf{0}^{(n-3)/4}|\mathbf{0}^{(n-3)/4}|\mathbf{g}|\mathbf{10}$, which has weight $(n-5)/4+(n-3)/4+1 = (n-3)/2$ and is thus prohibited. If the weight of $\mathbf{f}$ is $(n-5)/4$ then $\beta+\delta = \mathbf{0}|\mathbf{a}|\mathbf{0}^{(n-3)/4}|\overline{\mathbf{f}}|\mathbf{g}|\mathbf{00}$, which has weight $(n-3)/4$. If the weight of $\mathbf{g}$ is $(n-5)/4$ then $\gamma+\delta = \mathbf{0}|\mathbf{d}|\mathbf{0}^{(n-3)/4}|\mathbf{f}|\overline{\mathbf{g}}|\mathbf{00}$, which has weight $(n-3)/4$. Thus in all cases we contradict our assumption that C does not contain a codeword at distance $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$ from the vector $\mathbf{1}|\mathbf{0}^{n-1}$.

□

Theorem 3.15 *Let C be a linear code of length n and 1-covering radius one, where n is odd and greater than 3. Then C has 2-covering radius $(n+1)/2$.*

Proof: Let $\mathbf{a}, \mathbf{b} \in \boldsymbol{F}^n$. Assume that $\mathbf{a}$ or $\mathbf{b}$ is an element of C. Without loss of generality, let $\mathbf{a}$ be an element of C. If $\mathbf{x}$ is also a codeword then $\mathbf{a}+\mathbf{x}$ is a codeword as C is linear. Also, distance is preserved under translation. Therefore, there is an $\mathbf{x}$ in C such that $\operatorname{cov}(\mathbf{x}, \{\mathbf{a}, \mathbf{b}\}) = d$ if and only if there is a $\mathbf{y}$ in C such that $\operatorname{cov}(\mathbf{y}, \{\mathbf{0}^n, \mathbf{c}\}) = d$; i.e. where $\mathbf{c} = \mathbf{a} + \mathbf{b}$ (set $\mathbf{y} = \mathbf{a} + \mathbf{x}$). Therefore, we only need to consider coverings of $\{\mathbf{0}^n, \mathbf{c}\}$.

Suppose $\mathbf{c} \neq \mathbf{1}^n$. Let $r = \lfloor \operatorname{wt}(\mathbf{c})/2 \rfloor$ and let $\mathbf{y}$ be a vector with ones in r coordinates in which $\mathbf{c}$ has ones, and zeros elsewhere.

Then, since $\operatorname{wt}(\mathbf{c}) \leq n-1$,

$$\operatorname{dist}(\mathbf{0}^n, \mathbf{y}) = \left\lfloor \frac{\operatorname{wt}(\mathbf{c})}{2} \right\rfloor \leq \frac{n-1}{2}.$$

Also,

$$\operatorname{dist}(\mathbf{c}, \mathbf{y}) = \operatorname{wt}(\mathbf{c}) - \left\lfloor \frac{\operatorname{wt}(\mathbf{c})}{2} \right\rfloor = \left\lceil \frac{\operatorname{wt}(\mathbf{c})}{2} \right\rceil \leq \frac{n-1}{2}.$$

Since C has 1-covering radius one there exists a codeword $\mathbf{y}'$ with distance at most one from $\mathbf{y}$. Therefore,

$$\operatorname{dist}(\mathbf{0}^n, \mathbf{y}'), \operatorname{dist}(\mathbf{c}, \mathbf{y}') \leq \frac{n-1}{2} + 1 = \frac{n+1}{2}.$$

Suppose $\mathbf{c} = \mathbf{1}^n$. Lemma 3.13 says that C must have a codeword of weight $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$. This codeword covers $\{\mathbf{0}^n, \mathbf{1}^n\}$ with a ball of radius $(n+1)/2$.

Now assume neither $\mathbf{a}$ nor $\mathbf{b}$ is an element of C. As the 1-covering radius of C is one, there is a codeword α that differs from $\mathbf{a}$ in only one coordinate. Without loss of generality assume that this difference is in the first coordinate. By translating by α, we see that there is an $\mathbf{x}$ in C such that $\operatorname{cov}(\mathbf{x}, \{\mathbf{a}, \mathbf{b}\}) = d$ if and only if there is a $\mathbf{y}$ in C such that $\operatorname{cov}(\mathbf{y}, \{1|\mathbf{0}^{n-1}, \mathbf{c}\}) = d$.

As in the previous case, if dist($\mathbf{1}|\mathbf{0}^{n-1}, \mathbf{c}$) is less than or equal to $n-1$, then there is a vector $\mathbf{y}$ at distance at most (n-1)/2 from both. Furthermore, we can take $\mathbf{y}'$ to be a codeword at distance 1 from $\mathbf{y}$. So $\mathbf{y}'$ has distance at most $(n+1)/2$ to both $\mathbf{1}|\mathbf{0}^{n-1}$ and $\mathbf{c}$.

If $\mathbf{c} = \mathbf{0}|\mathbf{1}^n$ then Lemma 3.14 says that there is a codeword that has distance at most $(n+1)/2$ to both $\mathbf{1}|\mathbf{0}^{n-1}$ and $\mathbf{c}$. In all possible cases $\text{cov}(C, \{\mathbf{a}, \mathbf{b}\}) \leq (n+1)/2$ the reverse inequality comes from the bounds given in the introduction. □

3.5 The multicovering radius of BCH codes

The binary primitive BCH code of length $2^m - 1$ and designed distance $2e + 1$ is a cyclic

$$[n = 2^m - 1, k \geq 2^m - me - 1, d \geq 2e + 1]$$

code and is denoted BCH(e, m). BCH(e, m) is at least an e-error correcting code. Since its minimum distance is at least $2e + 1$ Theorem 1.2 shows that it can correct at least e errors. BCH($1, m$) is the Hamming code and $k = 2^m - me - 1$ if $2e - 1 \leq 2^{\lceil m/2 \rceil}$. The formal definition of BCH(e, m) can be found in MacWilliams and Sloane's book [18]. BCH codes are important because their correction capabilities are known and they can be easily encoded. The covering radius of the 2 and 3 error correcting BCH codes are known. We will focus on the 2-error correcting BCH code.

Theorem 3.16 (Gorenstein, Peterson and Zierler [8]) *The covering radius of the 2-error correcting BCH code, BCH$(2, m)$, for $m \geq 3$, is equal to 3.*

So Theorems 2.3 and 2.10 give the bound:$\lceil n/2 \rceil \leq t_2(\text{BCH}(2, m) \leq \lceil n/2 \rceil + 3$ when $m \geq 3$. To obtain the 2-covering radius of BCH($2, m$) we use certain well known relations between the weight distribution of a code and the weight distribution of its dual. These relations depend on certain polynomials known as the Krawtchouk polynomials. Here we describe only the properties of these polynomials that we will need. More thorough treatments of these relations and properties can be found in MacWilliams and Sloane's book [18] and Cohen et al.'s book [3].

Definition: The binary Krawtchouk polynomial of degree i in x $P_i^n(x)$ is defined by the following generating function:

$$\sum_{i=0}^{\infty} P_i^n(x) z^i = (1-z)^x (1+z)^{n-x}.$$

An explicit expression for a Krawtchouk polynomial is given by:

$$P_i^n(x) = \sum_{j=0}^{i} (-1)^j \binom{x}{j} \binom{n-x}{i-j}.$$

Usually n is fixed and is omitted. There are many relations involving Krawtchouk polynomials. Some can be found by rearranging binomial coefficients. Some relationships that we will use follow:

$$P_i(x) = \frac{(n-2i)P_i(x-1) - (x-1)P_i(x-2)}{n-x+1}, \tag{3.1}$$

$$P_i(x) = (-1)^i P_i(n-x), \tag{3.2}$$

$$\binom{n}{x} P_i(x) = \binom{n}{i} P_x(i), \tag{3.3}$$

and if n is even,

$$P_i^n(n/2) = \begin{cases} 0 & \text{if } i \text{ is odd,} \\ (-1)^{i/2}\binom{n/2}{i/2} & \text{if } i \text{ is even.} \end{cases} \tag{3.4}$$

Lemma 3.17 (Krasikov and Litsyn [15]) *For any integers x, n and i, with i even*

$$|P_i(x)| \leq \frac{\binom{n}{n/2}\binom{n/2}{i/2}}{\binom{n}{x}}.$$

Definition: Let $C \subseteq \boldsymbol{F}^n$ be a binary code. Its weight distribution $\mathbf{A}(C) = \mathbf{A} = (A_0, A_1, \ldots, A_n)$ is defined by

$$A_i = |\{c \in C : \mathrm{wt}(\mathbf{c}) = i\}|.$$

In other words the i^{th} component of $\mathbf{A}(C)$ is the number of codewords in C with weight i.

Definition: The distance distribution, $\mathbf{B}(C) = \mathbf{B} = (B_0, B_1, \ldots, B_n)$, is defined by

$$B_i = \frac{1}{|C|}|\{c_1, c_2 \in C : \mathrm{dist}(c_1, c_2) = i\}|.$$

Thus the i^{th} component of $\mathbf{B}(C)$ is the average number of codewords that are distance i from a codeword of C.

For linear codes the vectors $\mathbf{A}$ and $\mathbf{B}$ are equal. For a linear code C the distance distribution of the dual code $C^\perp$ is denoted $\mathbf{B}^\perp$ and is called the dual spectrum of C.

Theorem 3.18 (MacWilliams identities [18]) *For a linear code C of length n*

$$B_i^\perp = \frac{1}{|C|}\sum_{x=0}^{n} B_x P_i^n(x),$$

where $P_i^n(x)$ is the Krawtchouk polynomial of degree i.

Theorem 3.19 (Krasikov and Litsyn [15]) *Let C be the 2-error correcting BCH code of length $n' = n - 1 = 2^m - 1$. Then*

$$B_i = \frac{\binom{n'}{i}}{n^2}(1 + E_{i^*}),$$

where $i^* = i + 1$ if i is odd, $i^* = i$ if i is even,

$$|E_i| \leq \frac{n^2\binom{n}{n/2}\binom{n/2}{i/2}}{\binom{n}{i}\binom{n}{d}},$$

$d = 2^{m-1} - 2^{(m-1)/2}$ if m is odd, and $d = 2^{m-1} - 2^{m/2}$ is m is even.

We will also use Stirling's bound on factorials and the following bounds on binomial coefficients:

Lemma 3.20 *The following properties are satisfied for any non-negative integers n and k:*

1. $\binom{n}{k} \geq \left(\frac{n}{k}\right)^k$
2. $\binom{n}{k} \leq n^k$
3. $n! \in \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1 + \Theta\left(\frac{1}{n}\right)\right)$

Lemma 3.21 ([18] §9.9) *Let $\delta = 2^{m-1} - 2^{(m-1)/2}$ if m is odd and $\delta = 2^{m-1} - 2^{m/2}$ is m is even. The weight of any nonzero codeword of the dual of $BCH(2, m)$ lies in the range $[\delta, n - \delta]$.*

Adding an overall parity check appends a zero to all current vectors in the generator matrix of the dual and adds the all one vector to the generator matrix. Thus we have the following corollary.

Corollary 3.22 *Let $\delta = 2^{m-1} - 2^{(m-1)/2}$ if m is odd and $\delta = 2^{m-1} - 2^{m/2}$ is m is even. The weight of any codeword of the dual of the extended BCH$(2, m)$ code lies in the range $[\delta, n+1-\delta]$ or is equal to 0 or $n+1$.*

Lemma 3.21 tells much about the weight distribution of the dual 2-error correcting BCH code, and the MacWilliams identities, Theorem 3.18, can be used to examine the weight distribution of the BCH code from this information. However, many of our theorems and lemmas only apply when certain parameters are even. Because of this we will use the dual of the extended 2-error correcting BCH code, which only contains even weight vectors, to help us prove the following theorem. Extended codes are discussed in Section 2.1.3

Theorem 3.23 *Let $0 \leq a \leq 4$ and $0 \leq b \leq 3$. Let S and T be disjoint sets of coordinates with $|S| = a$ and $|T| = b$. Then for sufficiently large m there exists a codeword $\mathbf{v}$ of the code BCH$(2, m)$ with $(n-1)/2 - a + b \leq wt(\mathbf{v}) \leq (n+1)/2 + b$ and with zeros at all of the coordinates in S and ones at all of the coordinates in T, where $n = 2^m - 1$.*

Proof: Suppose i is a positive integer and U and V are disjoint sets of coordinates. Let $B_{i,U,V}$ be the number of codewords in the extended 2-error correcting BCH code with weight i, zeros in all of the coordinates in U and ones in all of the coordinates in V. If V is the empty set then we may omit it from our notation, i.e. $B_{i,U,\emptyset} = B_{i,U}$. Also, $B^{\perp}_{i,U,V}$ will denote the same quantity in the dual of B. We next establish a useful equation for $B_{i,U}$.

Suppose U is a set of coordinates from the primitive BCH code, and let the size of U be a. Let C_U be the subcode of the extended 2-error correcting BCH code with zeros in the coordinates of U and the last coordinate of the extended code. Since codewords of this subcode must have a zero in the last coordinate we can remove this coordinate and obtain a BCH codeword that has the same weight. The code C_U can be constructed by adding $a+1$ parity checks, namely the $a+1$ vectors that are all zero except in one of the coordinates in U or the last coordinate. From Corollary 3.22 we know that the minimal distance of the dual of the extended 2-error correcting BCH code is at least $(n+1)/2 - \sqrt{n+1}$. Therefore as long as $a + 1 < (n+1)/2 - \sqrt{n+1}$ the added parity checks are independent. Taking these additions into account and Corollary 3.22 we see that:

$$B^{\perp}_{i,U} = B^{\perp}_{n+1-i,U} = \begin{cases} \binom{a+1}{i} & \text{for } 0 \le i \le a+1 \\ 0 & \text{for } a+1 < i < \delta - a - 1 \end{cases},$$

where $\delta = (n+1)/2 - \sqrt{n+1}$. Also $\sum_{j=0}^{n+1} B^{\perp}_{j,U} = |C^{\perp}_U| = 2^{2m+1+a+1} = 2^{a+2}(n+1)^2$.

Using Theorem 3.18 and the above values for $B^{\perp}$ we have:

$$\begin{aligned} B_{i,U} &= \frac{1}{2^{a+2}(n+1)^2} \sum_{x=0}^{n+1} B^{\perp}_{x,U} P_i^{n+1}(x) \\ &= \begin{cases} 0 & \text{if } i \text{ is odd} \\ \frac{1}{2^{a+2}(n+1)^2} \left(\sum_{x=0}^{a+1} \binom{a+1}{x} \left(P_i^{n+1}(x) + P_i^{n+1}(n+1-x)\right) \right. \\ \left. + \sum_{x=\delta-a-1}^{n+2-\delta+a} B^{\perp}_{x,U} P_i^{n+1}(x) \right) & \text{if } i \text{ is even} \end{cases} \end{aligned}$$

When i is even we can use equation 3.2 to write $B_{i,U}$ as:

$$\begin{aligned} B_{i,U} &= \frac{1}{2^{a+2}(n+1)^2} \left(2 \sum_{x=0}^{a+1} \binom{a+1}{x} P_i^{n+1}(x) + \sum_{x=\delta-a-1}^{n+2-\delta+a} B^{\perp}_{x,U} P_i^{n+1}(x) \right), \\ &= \alpha_{i,a}(1 + \beta_{i,U}) \end{aligned}$$

where

$$\begin{aligned} \alpha_{i,a} &= \frac{A_{i,a}}{2^{a+2}(n+1)^2}, \\ \beta_{i,U} &= \frac{1}{A_{i,a}} \sum_{x=\delta-a-1}^{n+2-\delta+a} B^{\perp}_{x,U} P_i^{n+1}(x) \end{aligned}$$

with

$$A_{i,a} = 2 \sum_{x=0}^{a+1} \binom{a+1}{x} P_i^{n+1}(x).$$

Since both n and i are even in this case, we may use Lemma 3.17 to bound the absolute value of $\beta_{i,U}$.

$$\begin{aligned} |\beta_{i,U}| &= \frac{1}{A_{i,a}} \quad |\sum_{x=\delta-a-1}^{n+2-\delta+a} B^{\perp}_{x,U} P_i^{n+1}(x)| \\ &\le \frac{|C^{\perp}_S|}{A_{i,a}} \max\{|P_i^{n+1}(x)| \ : \ \delta - a - 1 \le x \le n + 2 - \delta + a\} \\ &\le \frac{\binom{n+1}{(n+1)/2}\binom{(n+1)/2}{i/2}}{\alpha_{i,a}\binom{n+1}{\delta-a-1}}. \end{aligned}$$

We denote the last quantity by $\gamma_{i,a}$. We now proceed by cases for different a and b.

Case($a = 4, b = 3$): Suppose S and T are arbitrary disjoint sets of coordinates with $|S| = 4$ and $|T| = 3$. It is sufficient to show $B_{(n+1)/2,S,T} \geq 1$ for sufficiently large n. This implies the existence of a weight $(n+1)/2$ codeword with the appropriate structure. Such a codeword satisfies the requirements of other cases as well, namely when $(n+1)/2$ and $(n-1)/2$ are in the range of acceptable weights. So any case where $(n-1)/2 - a + b \leq (n-1)/2 \leq (n+1)/2 + b$, which is equivalent to $b \leq a$, will also be proved. Also, any case where $(n-1)/2 - a + b \leq (n+1)/2$ and $a \leq 3$, in other words $b - 1 \leq a \leq 3$, will be satisfied. This leaves only the cases (0,2), (0,3) and (1,3) unsolved.

Let $T = \{t_1, t_2, t_3\}$. Using the inclusion exclusion principal we can write $B_{i,S,T}$ as follows:

$$\begin{aligned} B_{i,S,T} \quad = \quad & B_{i,S} \\ & - B_{i,S\cup\{t_1\}} - B_{i,S\cup\{t_2\}} - B_{i,S\cup\{t_3\}} \\ & + B_{i,S\cup\{t_1,t_2\}} + B_{i,S\cup\{t_1,t_3\}} + B_{i,S\cup\{t_2,t_3\}} \\ & - B_{i,S\cup T}. \end{aligned}$$

Rewriting this equation in terms of α and β yields:

$$\begin{aligned} B_{(n+1)/2,S,T} \quad = \quad & \alpha_{(n+1)/2,4}(1 + \beta_{(n+1)/2,S}) \\ & - \alpha_{(n+1)/2,5}(3 + \beta_{(n+1)/2,S\cup\{t_1\}} + \beta_{(n+1)/2,S\cup\{t_2\}} + \beta_{(n+1)/2,S\cup\{t_3\}}) \\ & + \alpha_{(n+1)/2,6}(3 + \beta_{(n+1)/2,S\cup\{t_1,t_2\}} + \beta_{(n+1)/2,S\cup\{t_1,t_3\}} + \beta_{(n+1)/2,S\cup\{t_2,t_3\}}) \\ & - \alpha_{(n+1)/2,7}(1 + \beta_{(n+1)/2,S\cup T}) \end{aligned} \tag{3.5}$$

To examine the asymptotic behavior of $B_{(n+1)/2,S,T}$ we consider the behavior of the α and β terms. Using equations 3.3 and 3.4 we can write $P^{n+1}_{(n+1)/2}(x)$ as,

$$\begin{aligned} P^{n+1}_{(n+1)/2}(x) \quad & = \quad \frac{\binom{n+1}{(n+1)/2} P^{n+1}_{x}((n+1)/2)}{\binom{n+1}{x}} \\ & = \begin{cases} \frac{(-1)^{x/2}\binom{n+1}{(n+1)/2}\binom{(n+1)/2}{x/2}}{\binom{n+1}{x}} & \text{if } x \text{ is even} \\ 0 & \text{if } x \text{ is odd} \end{cases} \end{aligned}$$

Thus,

$$\begin{aligned}
\alpha_{(n+1)/2,a} &= \frac{1}{2^{a+1}(n+1)^2} \sum_{x=0}^{a+1} \binom{a+1}{x} P^{n+1}_{(n+1)/2}(x) \\
&= \frac{\binom{n+1}{(n+1)/2}}{2^{a+1}(n+1)^2} \sum_{x=0}^{\lfloor \frac{a+1}{2} \rfloor} \frac{(-1)^x \binom{a+1}{2x} \binom{(n+1)/2}{x}}{\binom{n+1}{2x}} \\
&= \frac{\binom{n+1}{(n+1)/2}}{2^{a+1}(n+1)^2} \left(1 + \sum_{x=1}^{\lfloor \frac{a+1}{2} \rfloor} \frac{(-1)^x \binom{a+1}{2x} \binom{(n+1)/2}{x}}{\binom{n+1}{2x}} \right)
\end{aligned}$$

Since x and a are constant, $\binom{(n+1)/2}{x}$ is a polynomial of degree x, and $\binom{n+1}{2x}$ is a polynomial of degree $2x$, we have

$$\begin{aligned}
\frac{(-1)^x \binom{a+1}{2x} \binom{(n+1)/2}{x}}{\binom{n+1}{2x}} &\in \Theta\left(\frac{\binom{(n+1)/2}{x}}{\binom{n+1}{2x}} \right) \\
&\subseteq o(1).
\end{aligned}$$

for $x \geq 1$. Therefore,

$$\alpha_{(n+1)/2,a} \in \frac{\binom{n+1}{(n+1)/2}}{2^{a+1}(n+1)^2} \left(1 + o(1)\right).$$

Using our asymptotic bounds on α we can bound γ, which in turn bounds β.

$$\begin{aligned}
\gamma_{(n+1)/2,a} &= \frac{\binom{n+1}{(n+1)/2} \binom{(n+1)/2}{(n+1)/4}}{\alpha_{(n+1)/2,a} \binom{n+1}{\delta-a-1}} \\
&\in \frac{2^{a+1}(n+1)^2 \binom{(n+1)/2}{(n+1)/4}}{\binom{n+1}{\delta-a-1} (1+o(1))}
\end{aligned}$$

Using Stirling's formula we can estimate $\binom{(n+1)/2}{(n+1)/4}$. We have

$$\begin{aligned}
\binom{(n+1)/2}{(n+1)/4} &= \frac{((n+1)/2)!}{((n+1)/4)!((n+1)/4)!} \\
&\in \frac{\sqrt{\pi(n+1)}\ \frac{n+1}{2}^{(n+1)/2}\ e^{-(n+1)/2}\ (1+\Theta(\frac{1}{n}))}{(\sqrt{\pi(n+1)/2}\ \frac{n+1}{4}^{(n+1)/4}\ e^{-(n+1)/4}\ (1+\Theta(\frac{1}{n})))^2} \\
&\subseteq \Theta\left(\frac{2\sqrt{\pi(n+1)}\ (n+1)^{(n+1)/2}\ 4^{(n+1)/2}}{\pi(n+1)(n+1)^{(n+1)/2}\ 2^{(n+1)/2}} \right) \\
&\subseteq \Theta\left(\frac{2^{(n+1)/2}}{\sqrt{n+1}} \right).
\end{aligned}$$

We can also estimate $\binom{n+1}{\delta-a-1}$. We have

$$\begin{aligned}
\binom{n+1}{\delta-a-1} &= \binom{n+1}{\lceil (n+1)/2-\sqrt{n+1}\rceil-a-1} \\
&= \frac{\displaystyle\prod_{x=0}^{\lceil (n+1)/2-\sqrt{n+1}\rceil-a-2} (n+1-x)}{\displaystyle\prod_{x=0}^{\lceil (n+1)/2-\sqrt{n+1}\rceil-a-2} \left(\lceil (n+1)/2-\sqrt{n+1}\ \rceil-a-1-x\right)}
\end{aligned}$$

Since $\frac{n+1}{2}-\sqrt{n+1}+1 \geq \lceil \frac{n+1}{2}-\sqrt{n+1}\rceil$, we have

$$\begin{aligned}
n+1 &\geq n+1-2\sqrt{n+1}-2a-2x \\
&\geq 2\left(\left\lceil \frac{n+1}{2}-\sqrt{n+1}\right\rceil-a-1-x\right).
\end{aligned}$$

Also,

$$\begin{aligned}
\text{if} \qquad x &\geq \tfrac{1}{3}\left(n+1-4\sqrt{n+1}-4a\right) \\
\text{then} \qquad 3x &\geq \left(n+1-4\sqrt{n+1}-4a\right) \\
\text{then} \qquad -x &\geq \left(n+1-4\sqrt{n+1}-4a-4x\right) \\
\text{then} \quad n+1-x &\geq 2\left(n+1-2\sqrt{n+1}-2a-2x\right) \\
\text{then} \quad n+1-x &\geq 4\left(\lceil \tfrac{n+1}{2}-\sqrt{n+1}\rceil-a-1-x\right).
\end{aligned}$$

Thus

$$\begin{aligned}
\binom{n+1}{\delta-a-1} &\geq \left(\prod_{x=0}^{\lceil (n+1-4\sqrt{n+1}-4a)/3\rceil-1} 2\right)\left(\prod_{x=\lceil (n+1-4\sqrt{n+1}-4a)/3\rceil}^{\lceil (n+1)/2-\sqrt{n+1}\rceil-a-2} 2^2\right) \\
&= 2^{2\left(\lceil (n+1)/2-\sqrt{n+1}\rceil-a-2-\lceil (n+1-4\sqrt{n+1}-4a)/3\rceil+1\right)+\lceil (n+1-4\sqrt{n+1}-4a)/3\rceil} \\
&= 2^{2\lceil (n+1)/2-\sqrt{n+1}\rceil-2a-2-\lceil (n+1-4\sqrt{n+1}-4a)/3\rceil} \\
&\geq 2^{2\left((n+1)/2-\sqrt{n+1}\right)-2a-2-\left((n+1-4\sqrt{n+1}-4a)/3+1\right)} \\
&= 2^{\frac{2}{3}\left(n-\sqrt{n+1}-a-\frac{7}{2}\right)}.
\end{aligned}$$

For any $\epsilon > 0$, $2^{c\left(n-\sqrt{n+1}\right)} \in \Omega\left(2^{(c-\epsilon)n}\right)$. Therefore $\binom{n+1}{\delta-a-1} \in \Omega\left(2^{\left(\frac{2}{3}-\epsilon\right)n}\right)$ for any $\epsilon > 0$. So,

$$\gamma_{(n+1)/2,a} \in \frac{2^{a+1}(n+1)^2\Theta\left(\frac{2^{(n+1)/2}}{\sqrt{n+1}}\right)}{\Omega\left(2^{\left(\frac{2}{3}-\epsilon\right)n}\right)(1+o(1))}$$

This implies that $\gamma_{(n+1)/2,a}$ tends to 0 as n gets large and so must $\beta_{(n+1)/2,U}$, where U is of size a. Since $\beta_{(n+1)/2,U}$ tends to zero as n gets large equation 3.5 becomes

$$\begin{aligned}
B_{(n+1)/2,S,T} &= \alpha_{(n+1)/2,4} - 3\alpha_{(n+1)/2,5} + 3\alpha_{(n+1)/2,6} - \alpha_{(n+1)/2,7} \\
&\in \frac{\binom{n+1}{(n+1)/2}}{(n+1)^2}\left(\frac{1}{2^5} - \frac{3}{2^6} + \frac{3}{2^7} - \frac{1}{2^8} + o(1)\right) \\
&= \frac{\binom{n+1}{(n+1)/2}}{(n+1)^2}\left(\frac{1}{256} + o(1)\right).
\end{aligned}$$

So $B_{(n+1)/2,S,T}$ tends to infinity as n gets large and therefore there must exist BCH codewords of weight $(n+1)/2$ with the sought after structure for large enough m.

Case$(a=1, b=3)$: Suppose S and T are arbitrary disjoint sets of coordinates with $|S| = 1$ and $|T| = 3$. It is sufficient to show $B_{(n+5)/2,S,T} \geq 1$ for sufficiently large n. This is equivalent to the existence of a weight $(n+5)/2$ codeword with the appropriate structure. Such a codeword satisfies the requirements of cases (0,2) and (0,3), since $(n+5)/2$ is in the range of acceptable weights for those cases as well.

As in the previous case we let $T = \{t_1, t_2, t_3\}$. Using the inclusion exclusion principal we can write $B_{(n+5)/2,S,T}$ in terms of α and β

$$\begin{aligned}
B_{(n+5)/2,S,T} &= \alpha_{(n+5)/2,1}(1 + \beta_{(n+5)/2,S}) \\
&\quad -\alpha_{(n+5)/2,2}(3 + \beta_{(n+5)/2,S\cup\{t_1\}} + \beta_{(n+5)/2,S\cup\{t_2\}} + \beta_{(n+5)/2,S\cup\{t_3\}}) \\
&\quad +\alpha_{(n+5)/2,3}(3 + \beta_{(n+5)/2,S\cup\{t_1,t_2\}} + \beta_{(n+5)/2,S\cup\{t_1,t_3\}} + \beta_{(n+5)/2,S\cup\{t_2,t_3\}}) \\
&\quad -\alpha_{(n+5)/2,4}(1 + \beta_{(n+5)/2,S\cup T})
\end{aligned} \tag{3.6}$$

To examine the asymptotic behavior of $B_{(n+5)/2,S,T}$ we consider the behavior of the α and β terms. From the explicit expression for a Krawtchouk polynomial we have

$$P_i^n(n/2+2) = \sum_{j=0}^{i}(-1)^j\binom{\frac{n}{2}+2}{j}\binom{\frac{n}{2}-2}{i-j}.$$

This is a summation of polynomials in n of degree i and is therefore also a polynomial of degree i. So,

$$P_i^n(n/2+2) \in \Theta(n^i).$$

Also equation 3.3 implies

$$P_{n/2+2}^n(x) \in \frac{\binom{n}{\frac{n}{2}+2}\Theta(n^x)}{\binom{n}{x}}.$$

Thus,

$$
\begin{aligned}
\alpha_{(n+5)/2,a} &= \frac{1}{2^{a+1}(n+1)^2}\sum_{x=0}^{a+1}\binom{a+1}{x}P^{n+1}_{(n+5)/2}(x) \\
&\in \frac{\binom{n+1}{(n+5)/2}}{2^{a+1}(n+1)^2}\sum_{x=0}^{a+1}\frac{\binom{a+1}{x}\Theta(n^x)}{\binom{n+1}{x}} \\
&\subseteq \frac{\binom{n+1}{(n+5)/2}}{2^{a+1}(n+1)^2}\Theta(1).
\end{aligned}
$$

Using our asymptotic bounds on α we can bound γ, which in turn bounds β.

$$
\begin{aligned}
\gamma_{(n+5)/2,a} &= \frac{\binom{n+1}{(n+1)/2}\binom{(n+1)/2}{(n+5)/4}}{\alpha_{(n+5)/2,a}\binom{n+1}{\delta-a-1}} \\
&\in \frac{2^{a+1}(n+1)(n+5)(n+3)\binom{(n+1)/2}{(n+5)/4}}{(n-1)\binom{n+1}{\delta-a-1}\Theta(1)}
\end{aligned}
$$

Using Stirling's formula we can estimate $\binom{(n+1)/2}{(n+1)/4}$. We have

$$
\begin{aligned}
\binom{(n+1)/2}{(n+5)/4} &= \frac{((n+1)/2)!}{((n+5)/4)!((n-3)/4)!} \\
&= \frac{4(n-1)((n-3)/2)!}{(n+5)((n-3)/4)!((n-3)/4)!} \\
&\in \frac{4(n-1)\sqrt{\pi(n-3)}\ \frac{n-3}{2}^{(n-3)/2}\ e^{-(n-3)/2}\ (1+\Theta(\frac{1}{n}))}{(n+5)(\sqrt{\pi(n-3)/2}\ \frac{n-3}{4}^{(n-3)/4}\ e^{-(n-3)/4}\ (1+\Theta(\frac{1}{n})))^2} \\
&\subseteq \Theta\left(\frac{2\sqrt{\pi(n-3)}\ (n-3)^{(n-3)/2}\ 4^{(n-3)/2}}{\pi(n-3)(n-3)^{(n-3)/2}\ 2^{(n-3)/2}}\right) \\
&\subseteq \Theta\left(\frac{2^{(n-3)/2}}{\sqrt{n-3}}\right).
\end{aligned}
$$

As before $\binom{n+1}{\delta-a-1} \in \Omega\left(2^{\left(\frac{2}{3}-\epsilon\right)n}\right)$ for any $\epsilon > 0$. So,

$$
\gamma_{(n+5)/2,a} \in \frac{2^{a+1}(n+5)(n+3)(n+1)\Theta\left(\frac{2^{(n-3)/2}}{\sqrt{n-3}}\right)}{(n-1)\Omega\left(2^{\left(\frac{2}{3}-\epsilon\right)n}\right)\Theta(1)}.
$$

This implies that $\gamma_{(n+5)/2,a}$ tends to 0 as n gets large and so must $\beta_{(n+5)/2,U}$, where U is of size a. Since $\beta_{(n+5)/2,U}$ tends to zero as n gets large equation 3.6 becomes

$$
\begin{aligned}
B_{(n+5)/2,S,T} &= \alpha_{(n+5)/2,1} - 3\alpha_{(n+5)/2,2} + 3\alpha_{(n+5)/2,3} - \alpha_{(n+5)/2,4} \\
&\in \frac{\binom{n+1}{(n+5)/2}\Theta(1)}{(n+1)^2}\left(\frac{1}{2^2} - \frac{3}{2^3} + \frac{3}{2^4} - \frac{1}{2^5}\right) \\
&= \frac{\binom{n+1}{(n+5)/2}\Theta(1)}{32(n+1)^2}
\end{aligned}
$$

for large n. So $B_{(n+5)/2,S,T}$ tends to infinity as n gets large and therefore there must exist BCH codewords of weight $(n+5)/2$ with the sought after structure for large enough m. □

Corollary 3.24 *Given two vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *with* $a = n - dist(\mathbf{x}, \mathbf{y}) \leq 4$*, let* $\mathbf{z} = \overline{\mathbf{x}} + \mathbf{y}$*. There exist codewords* $\mathbf{u}$ *and* $\mathbf{v}$ *of the code* $BCH(2, m)$ *that satisfy the following properties for sufficiently large* m*:*

1. $b \triangleq dist(\mathbf{u}, \mathbf{x}) \leq 3$

2. $(n-1)/2 - a + b \leq wt(\mathbf{v}) \leq (n+1)/2 + b$

3. $supp(\mathbf{u} + \mathbf{x}) \subseteq supp(\mathbf{v})$

4. $supp(\mathbf{z}) \cap supp(\mathbf{u} + \mathbf{x} + \mathbf{v}) = \emptyset$

Proof: Property 1 can be satisfied since the covering radius of $\mathrm{BCH}(2, m) = 3$ for $m \geq 3$, Theorem 3.16. Note that $\mathrm{wt}(\mathbf{z}) = a$, $\mathrm{wt}(\mathbf{x} + \mathbf{u}) = b$, and $\mathrm{supp}(\mathbf{u} + \mathbf{x} + \mathbf{v}) = \mathrm{supp}(\mathbf{v}) - \mathrm{supp}(\mathbf{u} + \mathbf{x})$ so the fourth condition says that $\mathbf{v}$ has zeros wherever $\mathbf{z}$ is one and $\mathbf{u} + \mathbf{x}$ is zero. There are at most three such coordinates. The third condition says that $\mathbf{v}$ has ones wherever $\mathbf{u} + \mathbf{x}$ has ones. Thus by Theorem 3.23 with $S = \mathrm{supp}(\mathbf{z})$ and $T = \mathrm{supp}(\mathbf{x} + \mathbf{u})$ there exists a codeword $\mathbf{v}$ of $\mathrm{BCH}(2, m)$ that satisfies properties 2, 3, and 4. □

Theorem 3.25 $t_2(BCH(2, m)) = \frac{n+1}{2}$ *for sufficiently large* m*.*

Proof: Consider two vectors $\mathbf{x}$ and $\mathbf{y}$ with $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = n - a$, where $a \geq 5$. There exists a vector $\mathbf{v}'$ with distance at most $(n-5)/2$ to both $\mathbf{x}$ and $\mathbf{y}$ and there exists a codeword $\mathbf{v}$ with $\mathrm{dist}(\mathbf{v}, \mathbf{v}') \leq 3$. Thus the distance from $\mathbf{v}$ to both $\mathbf{x}$ and $\mathbf{y}$ is at most $(n-5)/2+3 = (n+1)/2$.

Consider two vectors $\mathbf{x}$ and $\mathbf{y}$ with $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = n - a$, where $a \leq 4$. Let $\mathbf{u}$, $\mathbf{v}$, $\mathbf{z}$, and b be as in Corollary 3.24. Then

$$\begin{aligned}
\mathrm{dist}(\mathbf{x}, \mathbf{u} + \mathbf{v}) &= \mathrm{wt}(\mathbf{u} + \mathbf{v} + \mathbf{x}) \\
&= \mathrm{wt}(\mathbf{v}) - \mathrm{wt}(\mathbf{u} + \mathbf{x}) \\
&= \mathrm{wt}(\mathbf{v}) - b \\
&\leq \frac{n+1}{2},
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{dist}(\mathbf{y}, \mathbf{u}+\mathbf{v}) &= \mathrm{wt}(\mathbf{u}+\mathbf{v}+\mathbf{y}) \\
&= \mathrm{wt}(\mathbf{u}+\mathbf{v}+\overline{\mathbf{x}}+\mathbf{z}) \\
&= n - \mathrm{wt}(\mathbf{u}+\mathbf{v}+\mathbf{x}+\mathbf{z}) \\
&= n - \mathrm{wt}(\mathbf{u}+\mathbf{v}+\mathbf{x}) - \mathrm{wt}(\mathbf{z}) \\
&= n - \mathrm{wt}(\mathbf{v}) + b - a \\
&\leq \frac{n+1}{2}.
\end{aligned}$$

□

Chapter 4

Complexity

In this chapter we introduce aspects of complexity theory and study the computational complexity of various problems related to the multicovering radii of codes. There are several problems that frequently occur when dealing with codes, such as: performing minimum distance decoding, determining minimum distance, finding the covering radius, etc. Therefore, knowing the complexity of such problems becomes important so we can apply appropriate techniques to them. We examine the complexity of various decision problems relating to error correcting codes, focusing on problems involving covering radii. For a detailed look at complexity theory see Garey and Johnson's book [7].

For now we will only consider problems that consist of a question with a "yes" or "no" answer. Such problems are called decision problems. We restrict our attention to decision problems since they have a natural formal analog known as a language. Given any finite set of symbols Σ we denote the set of all finite strings of symbols over Σ as Σ^*. If L is a subset of Σ^* then it is called a language over the alphabet Σ.

The correspondence between decision problems and language is given by the encoding scheme that is used to encode the instances of a problem. A problem and encoding scheme partition Σ^* into three parts: strings that are not encodings of instances of the problem, strings that are encodings of "no" instances and strings that are encoding of "yes" instances. The later class of strings is taken to be the language corresponding to the problem. Note that this language is dependent on the encoding scheme. We follow the more informal practice of considering only "reasonable" encoding schemes and properties that are independent under this restriction. For an encoding scheme to be accepted as reasonable it should be concise, not padded with unnecessary symbols, numbers should be represented in binary or some other larger base, and decodable in that any component of an instance could be extracted from the encoded instance in polynomial-time.

An algorithm A solves a decision problem if for every instance of that problem A returns the correct answer. The size of an instance of a problem is the length of the string encoding the instance by some reasonable encoding scheme. The time complexity function for an algorithm A that halts on all input strings over Σ is given by:

$$T_A(n) = \max\{t : \exists x \in \Sigma^* \text{ s.t. } |x| = n \text{ and } A \text{ takes time } t \text{ to halt on input } x\ \}$$

Algorithms are often classified by the asymptotic performance of this worst case running time. For a given function $g(n)$ we define the following sets of functions:

$$\begin{aligned}
\Theta(g(n)) &= \{f(n) : \exists c_1, c_2, n_0 > 0 \text{ s.t. } 0 \le c_1 g(n) \le f(n) \le c_2 g(n), \forall n \ge n_0\} \\
\mathcal{O}(g(n)) &= \{f(n) : \exists c, n_0 > 0 \text{ s.t. } 0 \le f(n) \le c g(n), \forall n \ge n_0\} \\
\Omega(g(n)) &= \{f(n) : \exists c, n_0 > 0 \text{ s.t. } 0 \le c g(n) \le f(n), \forall n \ge n_0\} \\
o(g(n)) &= \{f(n) : \forall c > 0, \exists n_0 > 0 \text{ s.t. } 0 \le f(n) < c g(n), \forall n \ge n_0\} \\
\omega(g(n)) &= \{f(n) : \forall c > 0, \exists n_0 > 0 \text{ s.t. } 0 \le c g(n) < f(n), \forall n \ge n_0\}
\end{aligned}$$

A polynomial-time algorithm is one whose time complexity function is bounded by some polynomial $p(n)$, where n is the size of an instance. That is $T_A(n) \in \mathcal{O}(p(n))$ for some polynomial $p(n)$. The class of polynomial-time solvable problems is denoted by P. Another fundamental class of decision problems is NP. A decision problem belongs to NP if it can be solved by a polynomial time nondeterministic algorithm. This is an algorithm that can be thought of as having two stages, a guessing stage and a verification stage. The guessing stage produces some structure s. The verification stage is deterministic and uses s to solve the problem. Such an algorithm solves a problem if whenever the correct answer is “no” there does not exists a guess s that can produce a “yes” in the verification stage, and if whenever the correct answer is “yes” there exists some guess s that produces a “yes” in the verification stage.

For example consider the following decision problem

NAME: Satisfiability (SAT)

INSTANCE: A set V of variables and a Boolean formula F over V.

QUESTION: Can F be satisfied?

Theorem 4.1 *SAT is in NP.*

4.1 NP Completeness

The concept of NP-completeness gives us a way of showing that a problem in NP is as difficult as any other problem in NP. An informal definition for NP-completeness is that a problem A in NP is NP-complete if any problem in NP could be solved in polynomial time if A can be solved in polynomial time. To formalize this idea we will introduce the idea of a reduction. A set A has a polynomial time mapping reduction to a set B, denoted $A \preceq_m^p B$, if there exists a function f that is computable in polynomial time so that

$$x \in A \Leftrightarrow f(x) \in B.$$

The function f can be thought of transforming an instance of problem A into an instance of problem B. Note that $\preceq_m^p$ is a binary relation over Σ^*. Some properties of $\preceq_m^p$ follow.

Theorem 4.2

1. *$\preceq_m^p$ is reflexive*
2. *$\preceq_m^p$ is transitive*
3. *$A \preceq_m^p B$ if and only if $\overline{A} \preceq_m^p \overline{B}$*
4. *$A \preceq_m^p B$ and $B \in P$ implies $A \in P$*
5. *$A \preceq_m^p B$ and $B \in NP$ implies $A \in NP$*
6. *If $A \in P$, then for all $B \neq \Sigma^*, \emptyset$ then $A \preceq_m^p B$*

Proof:

1. $\preceq_m^p$ is reflexive

   Let A be any set, and let f be the identity function ($f : x \rightarrow x$). Thus f is computable in $\mathcal{O}(|x|)$ time and $x \in A \Leftrightarrow f(x) = x \in A$. Therefore $A \preceq_m^p A$.

2. $\preceq_m^p$ is transitive

   Let A, B, and C be sets such that $A \preceq_m^p B \preceq_m^p C$. So there exists polynomial-time reductions f and g from A to B and B to C respectively. So $g(f(x))$ is a polynomial-time computable function. Furthermore, $x \in A \Leftrightarrow f(x) \in B \Leftrightarrow g(f(x)) \in C$ as f and g are reductions. Therefore $A \preceq_m^p C$.

3. $A \preceq_m^p B \Leftrightarrow \overline{A} \preceq_m^p \overline{B}$

   (⇒) Assume that $A \preceq_m^p B$ via a function f. So $x \in A \Leftrightarrow f(x) \in B$. Taking the contrapositive of this statement we see that $f(x) \notin B \Leftrightarrow x \notin A$. Thus, $\overline{A} \preceq_m^p \overline{B}$.

   (⇐) Similar to the above case.

4. $A \preceq_m^p B$ and $B \in$ P implies $A \in$ P

   Since $A \preceq_m^p B$ there must exist a function $f(x)$ that transforms instances of A into instances of B in polynomial time. Also there exists a polynomial time algorithm M that solves B because $B \in$ P. To solve an instance x of A in polynomial time we need only compute $f(x)$ and use $f(x)$ as the input to M.

5. $A \preceq_m^p B$ and $B \in$ NP implies $A \in$ NP

   Since B is in NP there exists a polynomial time nondeterministic algorithm M_B that solves it. A polynomial time nondeterministic algorithm to solve A can then be formed from M_B and the reduction between A and B. Given the input x use M_B on $f(x)$.

6. If $A \in$ P, then for all $B \neq \Sigma^*, \emptyset$ then $A \preceq_m^p B$

   Since B is not equal to Σ^* or $\emptyset$ there exists $y \in B$ and $z \notin B$. The set A is in P so there is a polynomial time algorithm that determines membership in A. The function $f(x)$ for the reduction can use this algorithm to determine whether a given input x is in A or not. If so $f(x) = y$ otherwise $f(x) = z$.

□

Definition: A set A is NP-complete if

1. $A \in$ NP
2. For every set $L \in$ NP, $L \preceq_m^p A$.

If A satisfies the second property A is said to be NP-hard.

Perhaps the biggest open problem in complexity theory is the question of whether P = NP. If A is NP-complete then every problem in NP can be transformed into A with only a polynomial increase in the amount of time needed to solve it. This leads us to the following characterization of when P is equal to NP.

Theorem 4.3 *If A is NP-complete, then $A \in P$ if and only if $P = NP$.*

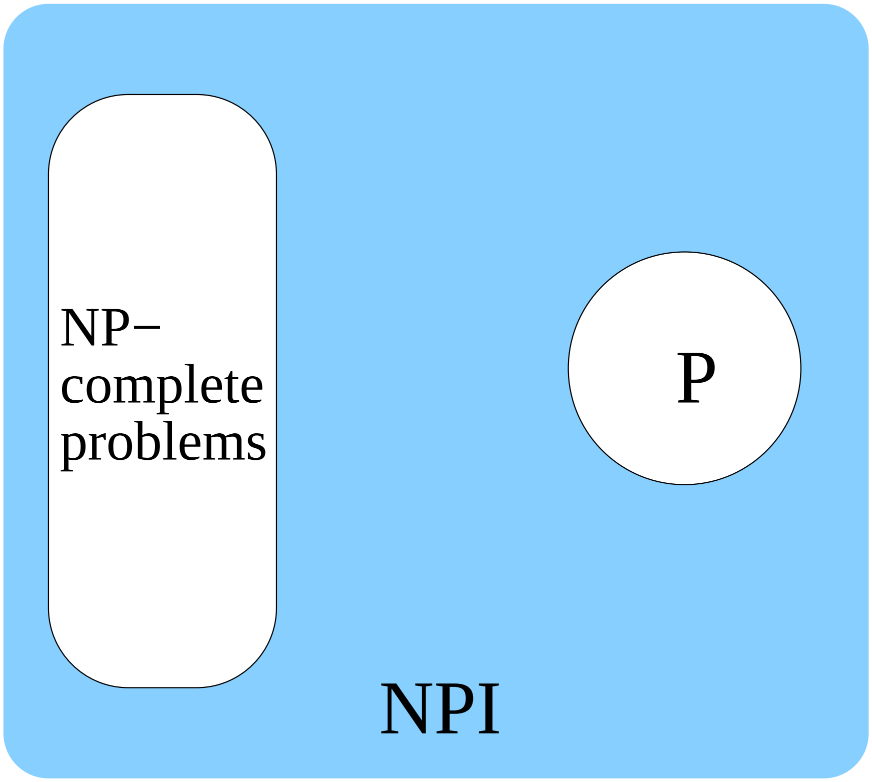

Figure 4.1: If P $\neq$ NP

Proof: If a problem can be solved in polynomial time deterministically then it can also be solved in polynomial time nondeterministically by discarding the structure generated from the guessing stage and proceeding with the original algorithm. Thus P $\subseteq$ NP.

Now suppose that A is NP-complete and let L be in NP. From the definition of NP-completeness $L \preceq_m^p A$. Since A belongs to P so does L by Theorem 4.2.4. Therefore NP $\subseteq$ P. □

This gives us three possibilities: P = NP, NP = P $\cup$ NP-complete, and NP = P $\cup$ NP-complete $\cup$ NPI. Where NPI = NP - (P $\cup$ NP-complete) is some set of languages having "intermediate" difficulty. Ladner [16] was able to show that if P $\neq$ NP then NPI is not empty.

Theorem 4.4 (Ladner) *If P $\neq$ NP, then there is a language in NP which is neither in P nor is it NP-complete.*

This gives us the picture in Figure 4.1 if P is not equal to NP. One question is whether NP-complete languages exist. It turns out that there exist hundreds of natural NP-complete problems. Many of these are cataloged in Garey and Johnson's book [7]. Working independently Cook [4] and Levin [17] were the first to discover natural NP-complete problems. Cook proved that the problem of determining whether a Boolean formula is satisfiable is NP-complete.

Theorem 4.5 (Cook's Theorem) *SAT is NP-complete*

The proof of Cook's Theorem is quite lengthy since it must be shown that any language in NP can be reduced to SAT. Fortunately once an NP-complete problem is known it is easier to prove that other languages are NP-complete using the following theorem.

Theorem 4.6 *If A is NP-complete, $A \preceq_m^p B$, and $B \in$ NP, then B is NP-complete.*

Proof: For any $L \in$ NP, $L \preceq_m^p A$ since A is NP-complete. If $A \preceq_m^p B$ then by the transitivity of $\preceq_m^p$, $L \preceq_m^p B$. Thus B is NP-hard. If B is also in NP then it is NP-complete. □

Therefore to show that a language B is NP-complete we need to show that B is in NP and that $A \preceq_m^p B$, where A is known to be NP-complete.

It is known that the following coding theoretic problems are NP-complete:

Covering Radius:

Lower bounding the covering radius of an arbitrary code. [6]

Weight of error:

Given a linear code C with parity check matrix $\mathbf{H}$, a vector $\mathbf{v}$, and a non-negative integer w, is it true that $\text{dist}(\mathbf{v}, C) \leq w$? [19]

Minimal weight:

Given a linear code with parity check matrix $\mathbf{H}$ and a non-negative integer w, is there a non-zero codeword $\mathbf{c}$ with weight less than or equal to w? [26]

Codeword of given weight:

Given a linear code with parity check matrix $\mathbf{H}$ and a non-negative integer w, is there a codeword $\mathbf{c}$ with weight equal to w? [19]

Also, when restricted to linear codes, the problem of lower bounding the covering radius is known to be Σ_2^p-complete [20]. Σ_2^p is a class of problems that will be introduced in the next section. The problem of minimal weight relates to finding the minimum distance of a linear code; the minimum distance of a linear code is the minimum weight of any non-zero codeword. Furthermore, Dumer, Micciancio and Sudan have shown that even approximating the minimum distance of a linear code within a constant factor, or within an additive error that is linear in the length of the code, is not possible in random polynomial time (RP), unless RP equals NP [22]. For an overview of computational complexity as it relates to coding theory see Barg's survey [1].

For each class of problems that we define we can also define the class of problems that are complementary to it. Given a set of problems S, let co-S be the set of instances that have their answers reversed. If a problem π can be solved by a deterministic polynomial time algorithm then the complement of π can as well. Simply reverse the output of the algorithm. However the same cannot be easily seen for nondeterministic algorithms. Note that P $=$ co-P $\subseteq$ NP $\cap$ co-NP. Since many problems in co-NP do not seem to be in NP, one might conjecture that NP is not equal to co-NP. This is in fact a stronger conjecture than P $\neq$ NP in that NP $\neq$ co-NP would imply P $\neq$ NP while the converse is not true. However there is a connection between NP-completeness and the conjecture that NP is not equal to co-NP. Figure 4.2 shows the state of affairs if NP is not equal to co-NP.

Theorem 4.7 *A is $\preceq_m^p$-complete for co-NP if and only if $\overline{A}$ is $\preceq_m^p$-complete for NP.*

Proof: ($\Rightarrow$) Assume that A is $\preceq_m^p$-complete for co-NP and let S be any element of NP. Thus $\overline{S}$ is an element of co-NP, A is an element of NP, and $\overline{S} \preceq_m^p A$. Then by Theorem 4.2.3 $S \preceq_m^p \overline{A}$. Therefore $\overline{A}$ is $\preceq_m^p$-complete for NP as the choice of S was arbitrary.

($\Leftarrow$) Similar to the above case. □

Theorem 4.8 *co-NP $=$ NP if and only if there exists a NP-complete problem A such that $\overline{A}$ is in NP.*

Proof: Suppose that co-NP $=$ NP. Then for any NP-complete problem, for example SAT, its complement will be in co-NP=NP.

Suppose there exists a NP-complete language A such that $\overline{A} \in$ NP. Theorem 4.7 says that $\overline{A}$ is then co-NP-complete. So if $\overline{L} \in$ co-NP, $\overline{L} \preceq_m^p \overline{A}$. Since $\overline{A}$ is in NP so is $\overline{L}$ by Theorem 4.2.5. Thus co-NP $\subseteq$ NP. The opposite containment can be shown similarly. □

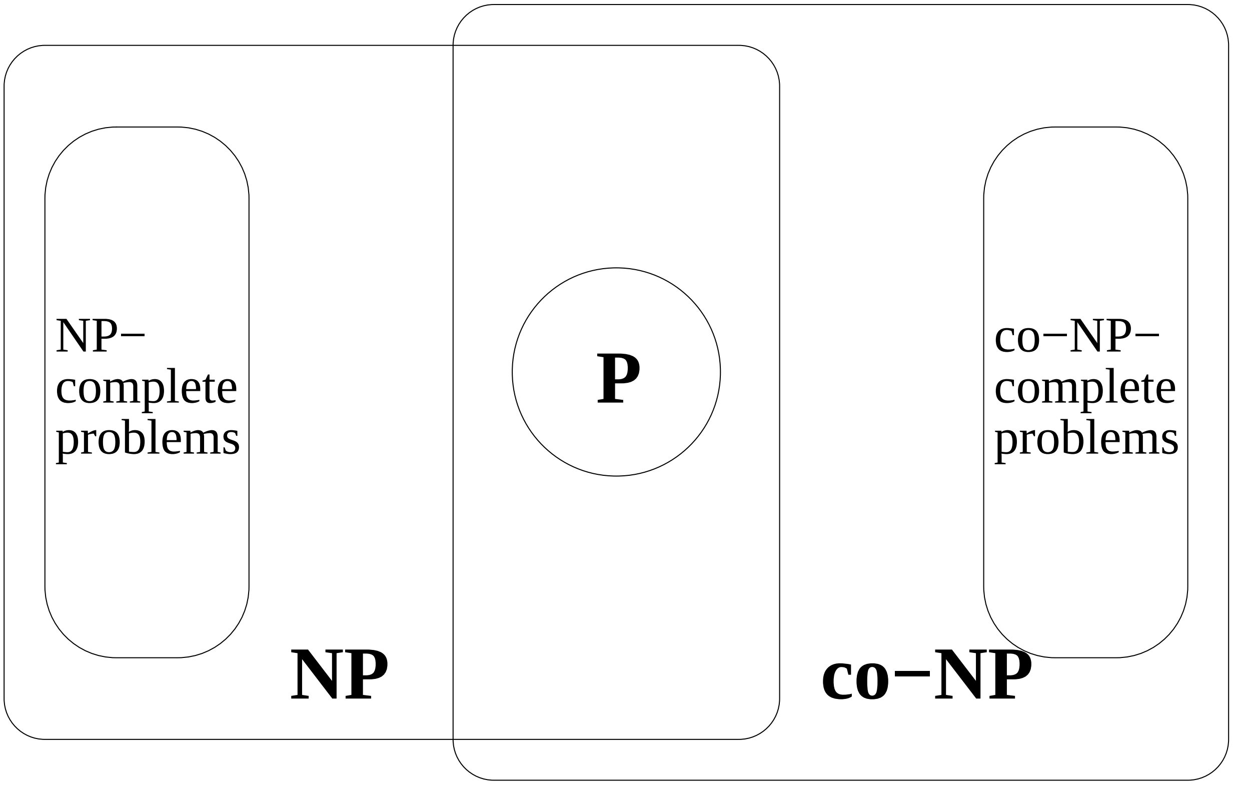

Figure 4.2: If NP $\neq$ co-NP

Theorem 4.9 *The problem of determining whether a formula of propositional logic is a tautology (TAUT) is $\preceq_m^p$-complete for co-NP.*

Proof: We know from Cook's Theorem that SAT is NP-complete, and so Theorem 4.7 implies that $\overline{\text{SAT}}$ is co-NP-complete. Note that $\overline{\text{SAT}}$, when given a propositional formula F, asks whether F is unsatisfiable. However, $\overline{\text{SAT}}$ polynomial-time reduces to TAUT. Map F to its negation (F is always false if and only if $\neg F$ is always true). Thus, TAUT is $\preceq_m^p$-complete for co-NP. □

4.2 Introduction to the polynomial hierarchy

The polynomial hierarchy formalizes a way to discuss problems that are NP-hard but might be harder than NP-complete problems. The following definitions are from Garey and Johnson's book [7]. The language classes P^Y and NP^Y are defined as follows, where Y is a set of languages:

$$\begin{aligned} \mathrm{P}^Y &= \{L : \exists L' \in Y \text{ such that there is a Turing reduction from } L \text{ to } L' \} \\ \mathrm{NP}^Y &= \left\{ L : \begin{array}{l} \exists L' \in Y \text{ such that there is a polynomial time} \\ \text{nondeterministic Turing reduction from } L \text{ to } L' \end{array} \right\}. \end{aligned}$$

Meyer and Stockmeyer [21] observed that you could continue to build these classes inductively giving an infinite hierarchy of classes of apparently growing difficulty. Let $\Sigma_0^p = \Pi_0^p = \Delta_0^p = \mathrm{P}$ and for all $k \geq 0$, $\Delta_{k+1}^p = \mathrm{P}^{\Sigma_k^p}$, $\Sigma_{k=1}^p = \mathrm{NP}^{\Sigma_k^p}$, and $\Pi_{k+1}^p = \mathrm{co} - \Sigma_{k+1}^p$. So $\Sigma_1^p = \mathrm{NP}$, $\Pi_1^p =$ co-NP, and $\Delta_1^p = \mathrm{P}$. The idea is that a problem is in Σ_k^p if it can be solved by a polynomial time nondeterministic algorithm that has access to an oracle that can provide solutions for some problem in Σ_{k-1}^p. We have the following containment relationships between classes in the polynomial hierarchy. The set Δ_k^p is contained by both Σ_k^p and Π_k^p. They are in turn both contained by Δ_{k+1}^p.

When we wish to determine whether a language is complete for a given class we must first show that the language is an element of the class. In the case of the polynomial hierarchy this would be difficult if we had to use the inductive definition itself. Fortunately, there is a more direct approach using relations. The following theorem is due to Wrathall [28].

Theorem 4.10 (Wrathall) *Let $L \subseteq \Gamma^*$ be a language, with $|\Gamma| \geq 2$. For any $k \geq 1$, $L \in \Sigma_k^p$ if and only if there exist polynomials $p_1, p_2, \ldots, p_k$ and a polynomial time recognizable relation R of dimension $k+1$ over Γ^* such that for all $x \in \Gamma^*$*

$$\begin{aligned} x \in L \quad \leftrightarrow \quad & (\exists y_1 \in \Gamma^* \textit{ such that } |y_1| \leq p_1(|x|)) \\ & (\forall y_2 \in \Gamma^* \textit{ such that } |y_2| \leq p_2(|x|)) \\ & \qquad \vdots \\ & (Q y_k \in \Gamma^* \textit{ such that } |y_k| \leq p_k(|x|)) \\ & [\langle x, y_1, y_2, \ldots, y_k \rangle \in R] \end{aligned}$$

where Q is an existential quantifier if k is odd and a universal quantifier if k is even (in general the quantifiers alternate).

A similar characterization for Π_k^p holds when the quantifiers alternate the other way $(\forall\exists\forall \ldots)$. Also each class in the polynomial hierarchy is closed under $\preceq_m^p$. So if A is complete for the class C in the polynomial hierarchy, $A \preceq_m^p B$, and B is in C, then B is complete for C.

Therefore, the situation is similar to the case of NP-completeness, it becomes much simpler to prove completeness for a particular class of the polynomial hierarchy once there is a known problem that is complete for that class. The following family of quantified versions of satisfiability performs that role for the polynomial hierarchy.

NAME: $\exists_1\forall_2\exists_3 \ldots Q_k - 3-$Satisfiability ($\exists_1\forall_2\exists_3 \ldots Q_k - 3-$SAT), where the quantifiers alternate and Q_k denotes $\exists$ if k is odd and $\forall$ if k is even.

INSTANCE: A k-tuple of integers $m_1, \ldots, m_k$ and a quantified Boolean expression

$$\exists u_{1,1} \ldots u_{1,m_1} \forall u_{2,1} \ldots u_{2,m_2} \exists u_{3,1} \ldots u_{3,m_3} \ldots Q u_{k,1} \ldots u_{k,m_k} E,$$

where E is in conjunctive normal form, there are three distinct literals in each clause and the quantified variables are all the variables of E.

QUESTION: Can E be satisfied with respect to the quantifiers?

Meyer and Stockmeyer [21] proved the following completeness result.

Theorem 4.11 (Meyer and Stockmeyer) *For all $k \geq 1$, $\exists_1\forall_2\exists_3 \ldots Q_k-3-SAT$ is complete for Σ_k^p, and the complementary problem $\forall_1\exists_2\forall_3 \ldots Q_k - 3-SAT$ is complete for Π_k^p.*

The hierarchy is somewhat fragile in the sense that equality at one level implies equality at all levels above it.

Theorem 4.12 *If there exists an $i \geq 1$ such that $\Sigma_i^p = \Pi_i^p$, then for all $j > i$ $\Sigma_j^p = \Pi_j^p = \Delta_j^p = \Sigma_i^p$.*

If the conclusion of Theorem 4.12 holds, then the polynomial hierarchy is said to collapse to the i^{th} level.

Corollary 4.13 *If P = NP, or if NP = co-NP, then the polynomial hierarchy collapses to the first level.*

So without knowing that P $\neq$ NP, the polynomial hierarchy cannot be shown to be a true hierarchy of classes each properly containing the last. However, the polynomial hierarchy remains interesting. The levels of the hierarchy do contain natural problems, many of which are complete for some class in the hierarchy.

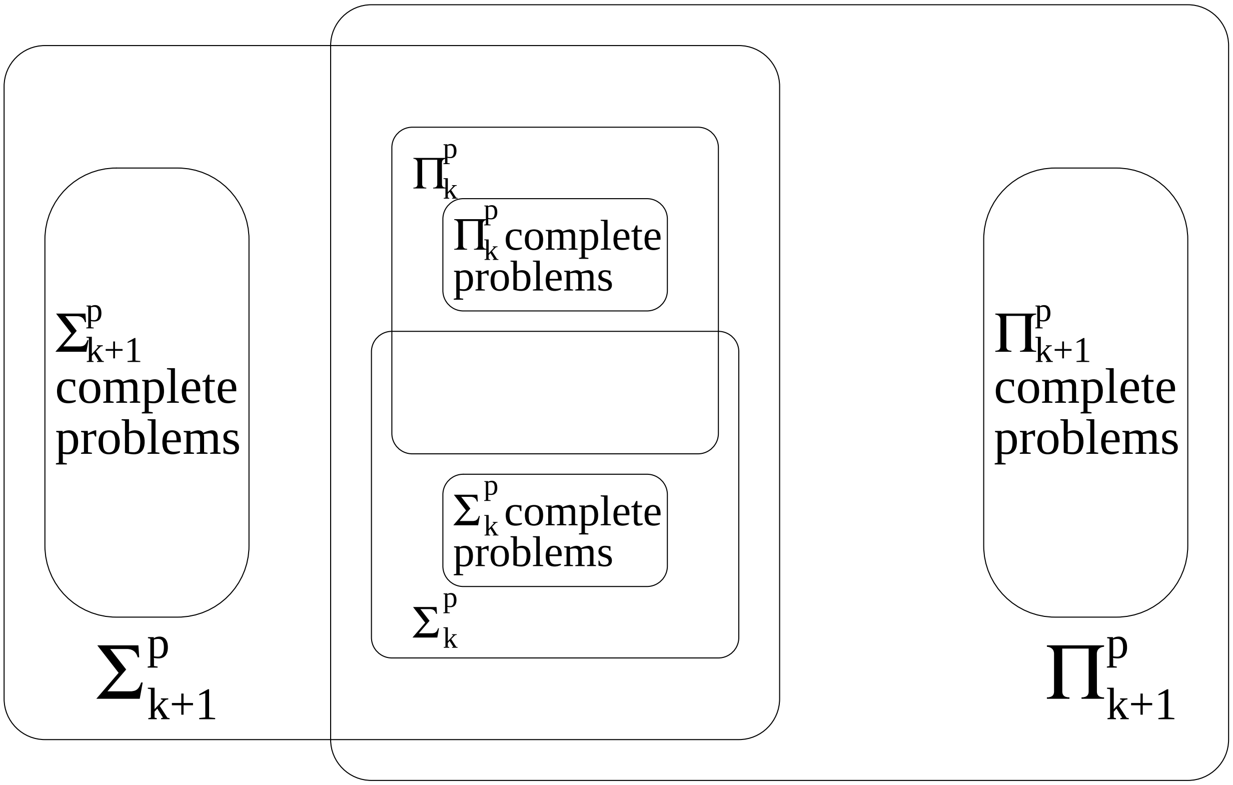
Π_k^p
Π_k^p complete problems
Σ_{k+1}^p complete problems
Π_{k+1}^p complete problems
Σ_k^p complete problems
Σ_k^p
Σ_{k+1}^p
Π_{k+1}^p

Figure 4.3: If $\Sigma_{k+1}^p \neq \Pi_{k+1}^p$ for $k \geq 1$

4.3 Complexity of bounding the 1-covering radius

The complexity of bounding the 1-covering radius of binary codes was first studied in the linear case where McLoughlin [20] showed that lower bounding the 1-covering radius of a linear binary code was Σ_2^p-complete. Later, Frances and Litman [6] proved that lower bounding the 1-covering radius in the unrestricted case was NP-complete. This difference in complexity can be explained by the fact that a linear code can be represented in a very compact form via a generator or parity check matrix. The problem of computing a lower bound of the multicovering radius of a binary code can be stated in the form of the following family of decision problems, where $m(n)$ is a function from the positive integers to the positive integers:

NAME: $\mathrm{LBC}_{m(n)}$ Lower bounding the multicovering radius of an arbitrary code:

INSTANCE: A binary code $C \subseteq \boldsymbol{F}^n$, given as a list of all codewords, and a positive integer w.

QUESTION: Does there exist an $m(n)$-tuple $(\mathbf{y}_1, \ldots, \mathbf{y}_{m(n)})$ of binary vectors of length n such that for every codeword c in C there is some $\mathbf{y}_i$ that is at least distance w from $\mathbf{c}$?

If we focus our attention only on linear codes the problem can be stated as:

NAME: $\mathrm{LBLC}_{m(n)}$ Lower bounding the multicovering radius of a linear code:

INSTANCE: A linear binary code $C \subseteq \boldsymbol{F}^n$, given by a parity check matrix $\mathbf{H}$ with dimensions $(n-k) \times n$, and a positive integer w.

QUESTION: Does there exist an $m(n)$-tuple $(\mathbf{y}_1, \ldots, \mathbf{y}_{m(n)})$ of binary vectors of length n such that for every vector $\mathbf{c} \in \boldsymbol{F}^n$, where $\mathbf{H}\mathbf{c}^{\mathrm{T}} = \mathbf{0}^{n-k}$, there is some $\mathbf{y}_i$ that is at least distance w from $\mathbf{c}$?

Note that the parameter $m(n)$ could be thought of as a constant integer, but we will take the more general view that m is a function of the length n of the code C. Often we will use the shorthand LBC_i or LBLC_i, where i is an integer, to mean that $m(n)$ is the constant function $m : n \mapsto i$. Also, a (n, K) code when represented by a list, or a $[n, k]$ code when represented by a generator or parity check matrix has size $\mathcal{O}(nK)$ or $\mathcal{O}(n^2)$ respectively.

To show that LBC_1 is NP-complete Frances and Litman constructed a polynomial-time reduction from 3-SAT, which was shown to be NP-complete by a transformation from SAT by Cook [4].

NAME: 3-satisfiability (3-SAT)

INSTANCE: A Boolean formula $E = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, in conjunctive normal form, with exactly three distinct literals in each clause.

QUESTION: Can E be satisfied?

To describe this reduction we will use facts about doubled vectors. We say that a vector $\mathbf{v} = v_1 v_2 \cdots v_{2n} \in \boldsymbol{F}^{2n}$ is doubled if $v_{2i-1} = v_{2i}$ for all i between 1 and n. Doubled vectors of length $2n$ can be characterized by their distances to a set Y_{2n}. This set can be constructed from the vector $\mathbf{u}(i) = 0101 \cdots 01 \in \boldsymbol{F}^{2i}$. Define $Y_{2n}^1 = \{01|\mathbf{u}(n-1), 10|\mathbf{u}(n-1)$, $01|\overline{\mathbf{u}}(n-1)$,$10|\overline{\mathbf{u}}(n-1)\}$.

Lemma 4.14 *If $\mathbf{v} \in \boldsymbol{F}^{2n}$ is such that for all $y \in Y_{2n}^1$, dist$(\mathbf{v}, \mathbf{y}) \leq n$, then $v_1 = v_2$.*

Proof: Note that

$$\text{dist}(\mathbf{v}, 01|\mathbf{u}(n-1)) = \text{dist}(v_1 v_2, 01) + \text{dist}(v_3 \ldots v_{2n}, \mathbf{u}(n-1)) \leq n \tag{4.1}$$

by our hypothesis. Similarly, $\text{dist}(v_1 v_2, 01) + \text{dist}(v_3 \ldots v_{2n}, \overline{\mathbf{u}}(n-1))$ is less than or equal to n. So

$$\text{dist}(v_1 v_2, 01) + 2(n-1) - \text{dist}(v_3 \ldots v_{2n}, u(n-1)) \leq n \tag{4.2}$$

as $\text{dist}(\mathbf{x}, \mathbf{y}) = k - \text{dist}(\mathbf{x}, \overline{\mathbf{y}})$ if $\mathbf{x}$ and $\mathbf{y}$ have length k. Adding the two inequalities 4.1 and 4.2 together we see that $\text{dist}(v_1 v_2, 01) \leq 1$. The same argument using the other two vectors in Y_{2n}^1 shows that $\text{dist}(v_1 v_2, 10) \leq 1$. Therefore $v_1 = v_2$. □

Let s_j denote the circular right shift by $2j-2$ bits and $Y_{2n}^j = \{s_j(\mathbf{y}) : \mathbf{y} \in Y_{2n}^1\}$.

Lemma 4.15 *If $\mathbf{v} \in \boldsymbol{F}^{2n}$ is such that for all $\mathbf{y} \in Y_{2n}^j$, dist$(\mathbf{v}, \mathbf{y}) \leq n$, then $v_{2j-1} = v_{2j}$.*

Let $Y_{2n} = \bigcup_{j=1}^{n} Y_{2n}^j$.

Lemma 4.16 *If $\mathbf{v} \in \boldsymbol{F}^{2n}$ is doubled if and only if dist$(\mathbf{v}, \mathbf{y}) \leq n$ for all $\mathbf{y} \in Y_{2n}$.*

Theorem 4.17 (Frances and Litman) *The decision problem LBC_1 is NP-complete.*

Proof: Let $E = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be an instance of 3-SAT using the variables $x_1, x_2, \ldots, x_n$. For each clause C_j define the vector $\mathbf{z}(C_j) \in \boldsymbol{F}^{2n}$ where:

$$\begin{aligned} z_{2i-1} &= & z_{2i} &= 0 & &\text{if } C_j \text{ contains } \overline{x}_i \\ z_{2i-1} &= & z_{2i} &= 1 & &\text{if } C_j \text{ contains } x_i \\ z_{2i-1} &= & 0, z_{2i} &= 1 & &\text{otherwise.} \end{aligned}$$

We then define the code $C \subseteq \boldsymbol{F}^{2n+2}$ to be the set $\{\mathbf{z}(C_j)|00 : i \leq j \leq m\} \cup Y_{2(n+1)}$, and $w = n+1$. The code C and integer w form our instance of LBC$_1$. The cardinality of C is $2n+4+m$ so this construction can be done in polynomial-time with respect to the size of E.

To see that this is indeed a reduction we must show that positive and negative instances coincide. Assume that E can be satisfied. Any satisfying assignment can be represented by a vector $\mathbf{v} \in \boldsymbol{F}^n$, where the i^{th} coordinate of $\mathbf{v}$ is one if x_i is assigned to be true and zero otherwise. Now let $\mathbf{v}^* = v_1v_1v_2v_2\cdots v_nv_n|00 \in \boldsymbol{F}^{2n+2}$. This vector v^* is doubled so $\text{dist}(\mathbf{c}, \mathbf{v}^*) \leq n+1 = w$ for every $\mathbf{c} \in Y_{2n}$ by Lemma 4.16. In each clause there is at least one literal which is set true by the assignment corresponding to $\mathbf{v}$ as it satisfies E. So $\text{dist}(\mathbf{z}(C_j)|00, \mathbf{v}^*) \leq 2+2+0+(n-3) = w$. So for every $\mathbf{c} \in C$, $\text{dist}(\mathbf{c}, \overline{\mathbf{v}}^*) \geq 2n+2-(n+1) = n+1$. Thus C has covering radius at least w.

Assume that C has covering radius at least w. Then there exists a vector $\mathbf{v}^* \in \boldsymbol{F}^{2n+2}$ such that $\text{dist}(\mathbf{v}^*, \mathbf{c}) \geq n+1$. As $\text{dist}(\overline{\mathbf{v}}^*, \mathbf{c}) \leq n+1$ for every $\mathbf{c}$ in C, $\overline{\mathbf{v}}^*$ is doubled. Let $\overline{\mathbf{v}}^* = v_1v_1v_2v_2\ldots v_nv_nv_{n+1}v_{n+1}$. Then $\text{dist}(v_1v_1v_2v_2\ldots v_nv_n, \mathbf{z}(C_j)) \leq n+1$. Therefore there exists an i such that $z_{2i-1} = z_{2i} = v_i$. So the assignment corresponding to this vector satisfies E. □

Theorem 4.18 (McLoughlin) *The decision problem LBLC$_1$ is Σ_2^p-complete.*

4.4 Complexity of bounding the m-covering radius

We now show that LBC$_{m(n)}$ is NP-complete and LBLC$_{m(n)}$ is Σ_2^p-complete for any function $m(n)$ that is polynomially bounded in n. This is done by showing that there is a polynomial time mapping reduction from LBC$_1$ to LBC$_{m(n)}$, and from LBLC$_1$ to LBLC$_{m(n)}$. First we need a lemma using the minimum distance of a code.

Lemma 4.19 *If C is a (n, K, d) code, then $\forall \mathbf{y} \in \boldsymbol{F}^n$, $\mathbf{c}.1, \mathbf{c}.2 \in C$ such that $\mathbf{c}.1 \neq \mathbf{c}.2$, we have dist$(\mathbf{y}, \mathbf{c}.1) +$ dist$(\mathbf{y}, \mathbf{c}.2) \leq 2n - d$.*

Proof: For any distinct $\mathbf{c}.1$, $\mathbf{c}.2 \in C$ and $\mathbf{y} \in \boldsymbol{F}^n$ we have:

$$\begin{aligned} \text{dist}(\mathbf{y}, \mathbf{c}.1) + \text{dist}(\mathbf{y}, \mathbf{c}.2) &= 2n - (\text{dist}(\mathbf{y}, \overline{\mathbf{c}.1}) + \text{dist}(\mathbf{y}, \overline{\mathbf{c}.2})) \\ &\leq 2n - \text{dist}(\overline{\mathbf{c}.1}, \overline{\mathbf{c}.2}) \\ &\leq 2n - d. \end{aligned}$$

□

We are now ready to prove our first completeness result. The heart of our result is a construction that takes a given code and produces another code whose m-covering radius is only dependent on the 1-covering radius and length of the original code. Given a binary (n, K) code C and positive integer m, let $l = \lfloor \log(m) \rfloor$. Note that $\log(m) \geq l \geq \log(m+1) - 1$. Let S be the code that has the following $l \times l(2n+1)$ matrix as a generator matrix:

$$\begin{pmatrix} \mathbf{1}^{2n+1} & \mathbf{0}^{2n+1} & \mathbf{0}^{2n+1} & \cdots & \mathbf{0}^{2n+1} \\ \mathbf{0}^{2n+1} & \mathbf{1}^{2n+1} & \mathbf{0}^{2n+1} & \cdots & \mathbf{0}^{2n+1} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0}^{2n+1} & \mathbf{0}^{2n+1} & \mathbf{0}^{2n+1} & \cdots & \mathbf{1}^{2n+1} \end{pmatrix}$$

Thus S is a $[r = l(2n+1), l, d = 2n+1]$ linear code. We also use the direct product of codes. Given the codes A and B their direct product is $A \times B = \{\mathbf{a}|\mathbf{c} : \mathbf{a} \in A, \mathbf{b} \in B\}$.

Lemma 4.20 *A (n, K) code C has 1-covering radius w if and only if $C' = C \times S$ has m-covering radius $w + r$, where r is the length of S.*

Proof: Suppose the 1-covering radius of C is at least w. That is, there exists a $\mathbf{y} \in \boldsymbol{F}^n$ such that for every $\mathbf{c} \in C$ the distance from $\mathbf{y}$ to $\mathbf{c}$ is at least w. Let $\mathbf{s}.1, \mathbf{s}.2, \ldots, \mathbf{s}.2^l$ be the elements of S and $\mathbf{y}.i = \mathbf{y}|\overline{\mathbf{s}.i}$, where $1 \leq i \leq 2^l$. If $\mathbf{c}' \in C'$, then $\mathbf{c}' = \mathbf{c}|\mathbf{s}.i$ for some $\mathbf{c} \in C$ and $\mathbf{s}.i \in S$. Therefore, the distance from $\mathbf{y}.i$ to $\mathbf{c}|\mathbf{s}.i$ is at least $w + r$. Since l is at most $\log(m)$, the cardinality of S and $\{\mathbf{y}.i : i = 1, \ldots, 2^l\}$ is at most m. Thus $t_m(C') \geq t_{2^l}(C') \geq w + r$.

Now suppose that the m-covering radius of C' is at least $w + r$. So there exist vectors $\mathbf{y}.1, \ldots, \mathbf{y}.m$ such that for every $\mathbf{c}' \in C'$ the distance from $\mathbf{c}'$ to one or more $\mathbf{y}.i$ is at least $w + r$. Let T_i be a minimal subset of $\{\mathbf{y}.1, \ldots, \mathbf{y}.m\}$ such that for every $\mathbf{c} \in C$ there is a $\mathbf{y}$ in T_i where the distance from $\mathbf{y}$ to $\mathbf{c}|\mathbf{s}.i$ is at least $w + r$; hence $1 \leq |T_i| \leq m$.

We claim that for every $\mathbf{y} \in \boldsymbol{F}^{n+r}$, if there exist vectors $\mathbf{c}.1 \in C$ and $\mathbf{s}.i \in S$ such that the distance from $\mathbf{y}$ to $\mathbf{c}.1|\mathbf{s}.i$ is at least $w + r$, then there do not exist vectors $\mathbf{c}.2 \in C$ and $\mathbf{s}.j \in S$, such that $\mathbf{s}.i \neq \mathbf{s}.j$ and the distance from $\mathbf{y}$ to $\mathbf{c}.2|\mathbf{s}.j$ is greater than or equal to $w+r$. In other words, if there is a vector that is far (distance at least $w+r$) from a codeword that ends in $\mathbf{s}.i$, then it cannot be far from a codeword of any other form. Assume that the claim is false; there exist vectors $\mathbf{y} \in \boldsymbol{F}^{n+r}$, $\mathbf{c}.1, \mathbf{c}.2 \in C$, and distinct vectors $\mathbf{s}.i, \mathbf{s}.j \in S$ such that:

$$\text{dist}(\mathbf{y}, \mathbf{c}.1|\mathbf{s}.i) = \text{dist}(\mathbf{y}', \mathbf{c}.1) + \text{dist}(\mathbf{y}'', \mathbf{s}.i) \geq w + r \tag{4.3}$$

$$\text{dist}(\mathbf{y}, \mathbf{c}.2|\mathbf{s}.j) = \text{dist}(\mathbf{y}', \mathbf{c}.2) + \text{dist}(\mathbf{y}'', \mathbf{s}.j) \geq w + r \tag{4.4}$$

where $\mathbf{y}'$ consists of the first n coordinates of $\mathbf{y}$ and $\mathbf{y}''$ the remaining r. Adding inequalities (4.3) and (4.4) yields:

$$\text{dist}(\mathbf{y}', \mathbf{c}.1) + \text{dist}(\mathbf{y}'', \mathbf{s}.i) + \text{dist}(\mathbf{y}', \mathbf{c}.2) + \text{dist}(\mathbf{y}'', \mathbf{s}.j) \geq 2(w + r). \tag{4.5}$$

Also from Lemma 4.19 we have the inequality:

$$\text{dist}(\mathbf{y}'', \mathbf{s}.i) + \text{dist}(\mathbf{y}'', \mathbf{s}.j) \leq 2r - d. \tag{4.6}$$

Thus,

$$\text{dist}(\mathbf{y}', \mathbf{c}.1) + \text{dist}(\mathbf{y}', \mathbf{c}.2) \geq 2(w + r) - 2r + d = 2w + d > 2n,$$

which is a contradiction as the length of each of the above vectors is n.

So if $\mathbf{y}$ is an element of T_i, then it cannot also be an element of T_j, where $i \neq j$. Therefore, there exists a T_i with cardinality one. Otherwise, each T_i contains at least 2 distinct vectors for a total of at least 2^{l+1} distinct vectors in the union of every T_i. So, $m \geq 2^{l+1}$, but this contradicts the fact that $l \geq \log(m + 1) - 1$. Furthermore, if $\text{dist}(\mathbf{y}, \mathbf{c}|\mathbf{s}.i) \geq w + r$ then $\text{dist}(\mathbf{y}', \mathbf{c}) \geq w + r - \text{dist}(y'', \mathbf{s}.i) \geq w$. So we have a vector whose distance to each codeword is at least w. □

We use Lemma 4.20 to form a polynomial time mapping reduction from LBC$_1$ to LBC$_{m(n)}$.

Theorem 4.21 *The decision problem LBC$_{m(n)}$ is NP-complete for any $m(n)$ that is polynomially bounded by the length of the code.*

Proof: Let C be a binary (n, K) code and $m(n)$ be polynomially bounded by n. So there exists an integer i such that $m(n) \in \mathcal{O}(n^i)$. The Hamming distance between any two vectors of length n can be found in time $\mathcal{O}(n)$ by simple comparisons. Similarly $\text{cov}(\mathbf{y}, C)$ and $\text{cov}(C, S)$ can be computed in time $\mathcal{O}(Kn)$ and $\mathcal{O}(m(n)Kn)$ respectively, where $|S| = m(n)$. So, $\mathcal{O}(m(n)Kn) \subseteq \mathcal{O}(Kn^{i+1})$. Thus the problem $\text{LBC}_{m(n)}$ belongs to NP, since it can be verified in polynomial time with respect to the size of the code C.

To show that $\text{LBC}_{m(n)}$ is also NP-hard we give a reduction from LBC_1. Given any instance of LBC_1, a binary (n, K) code C and a positive integer w, let $C' = C \times S$ be the code and w' the integer given in the construction of Lemma 4.20. As stated in the construction S is a $[r = l(2n+1), l = \lfloor \log(m(n)) \rfloor, d = 2n+1]$ code. So C' is a $(n+r, K2^l)$ code. However, the total size of C' is still polynomial in the size of C since l is bounded by the logarithm of $m(n)$. This shows that our construction can be done in polynomial time. Furthermore, Lemma 4.20 says that $C \in \text{LBC}_1$ if and only if $C' \in \text{LBC}_{m(n)}$. □

In a similar fashion we form a reduction from LBLC_1 to $\text{LBLC}_{m(n)}$.

Theorem 4.22 *The decision problem $LBLC_{m(n)}$ is Σ_2^p-complete for any $m(n)$ that is polynomially bounded by the length of the code.*

Proof: Let U be the set of all $\langle \mathbf{H}, w, Y, \mathbf{c} \rangle$ where $\mathbf{H}$ is a binary matrix, w is an integer, Y is a set of binary vectors, and $\mathbf{c}$ is a binary vector. We define the relation R to be the subset of U where if $\mathbf{H}$ is a $(n-k) \times n$ matrix then every vector in Y, as well as $\mathbf{c}$, have length n, $\mathbf{H}\mathbf{c}^{\mathrm{T}} = \mathbf{0}^{n-k}$, and $\text{cov}(\mathbf{c}, Y) \geq w$. It is trivial to determine if Y and $\mathbf{c}$ have the appropriate length in polynomial time with respect to the size of $\mathbf{H}$ and the cardinality of Y. The matrix multiplication can also be done in polynomial time. As we discussed in the proof of Theorem 4.21, $\text{cov}(\mathbf{c}, Y)$ can be computed in time $\mathcal{O}(m(n)n)$. Therefore when $m(n)$ is polynomially bounded by n, membership in R can be determined in polynomial time. Furthermore, $\text{LBLC}_{m(n)}$ can be written as: Given a $n \times k$ binary matrix $\mathbf{H}$ and integer w, is it true that

$$\exists \mathbf{y}.1, \ldots, \mathbf{y}.m(n) \in \boldsymbol{F}^n \forall \mathbf{c} \in \boldsymbol{F}^n \langle \mathbf{H}, w, \{\mathbf{y}.1, \ldots, \mathbf{y}.m(n)\}, \mathbf{c} \rangle \in R?$$

Thus $\text{LBLC}_{m(n)}$ is in Σ_2^p.

Now we show that $\text{LBLC}_{m(n)}$ is Σ_2^p-hard. Let the matrix $\mathbf{H}$ and positive integer w be any instance of LBLC_1. Since the generator matrix of the $[r, l]$ code S used in Lemma 4.20

can be formed in polynomial time, so too can the parity check matrix $\mathbf{H}'$ of $C' = C \times S$. Also, Lemma 4.20 says that $C \in \mathrm{LBLC}_1$ if and only if $C' \in \mathrm{LBLC}_{m(n)}$. □

Another way of forming the decision problem of computing a lower bound of the m-covering radius of a binary code would be to include m in the instance of the problem. This can be stated in the form of the following decision problem:

NAME: LBCM Lower bounding the multicovering radius of an arbitrary binary code:

INSTANCE: A binary code $C \subseteq \boldsymbol{F}^n$, given as a list of all codewords, and positive integers w and m.

QUESTION: Does there exist an m-tuple $(\mathbf{y}.1, \ldots, \mathbf{y}.m)$ of binary vectors of length n such that for every codeword $\mathbf{c}$ in C there is some $\mathbf{y}.i$ that is at least distance w from c?

Theorem 4.23 *The decision problem LBCM is NP-complete.*

Proof: Let C, a binary (n, K) code, and positive integers w and m be an instance of LBCM. If $w > n$, then the answer must be "no" since any collection of vectors can be covered in radius n by any one vector. If $w \leq \lceil n/2 \rceil$, then the answer must be "yes" since the closest that a vector can mutually be to two complementary vectors is $\lceil n/2 \rceil$. If $m \geq K$, then $t_m(C) = n$ as the m-tuple can contain the complement of the code C. The only time when there is ambiguity in the answer is when $1 \leq m < K$ and $\lceil n/2 \rceil < w \leq n$, and this case can be verified in polynomial time. Thus, LBCM is in NP. Furthermore, $\langle C, w \rangle \in \mathrm{LBC}_1$ if and only if $\langle C, w, 1 \rangle \in \mathrm{LBCM}$. So LBCM is NP-complete. □

The case when the code is linear is similar and the problem is Σ_2^p-complete.

4.5 Complexity of approximating the m-covering radius

We have seen that it is doubtful that one can exactly determine the m-covering radius of a code in polynomial time. So we ask what happens if we relax things and require only an approximation of the m-covering radius. We show that for at least one type of performance guarantee this is also as hard as the original problem.

First let us set the stage with some definitions. A combinatorial optimization problem is either a minimization or a maximization problem that consists of the following:

- I, a set of instances.
- For each i in I, a finite set $S(i)$. The elements of $S(i)$ are called the candidate solutions for i.
- A function m, called the objective function, that takes an instance i and a candidate solution s in $S(i)$ and returns a positive rational number $m(i,s)$, called the solution value of s.

If a problem is a minimization or respectively a maximization problem, then an optimal solution of an instance is a candidate solution with the minimum or respectively the maximum solution value. An approximation algorithm for an optimization problem is any algorithm A that when given an instance i of the problem returns a candidate solution $sA(i)$. The solution value of $sA(i)$ will be denoted $A(i)$. That is, $A(i) = m(i, sA(i))$. If $A(i)$ is always an optimal solution then A is called an optimization algorithm. The problem of approximating the multicovering radius can be given in the form of the following maximization problems:

NAME: $\mathrm{ACR}_{m(n)}$ Approximating the multicovering radius of an arbitrary binary code:

INSTANCES: I is the set of all binary codes.

CANDIDATES: for each (n, K) code C let $S(C)$ be the set of all $m(n)$-tuples of binary vectors with length n.

OBJECTIVE FUNCTION: Given any code C and $m(n)$-tuple Y, let $m(C,Y) = \mathrm{cov}(C,Y)$.

NAME: $\mathrm{ACRL}_{m(n)}$ Approximating the multicovering radius of a linear code:

INSTANCES: I is the set of all parity check matrices for linear binary codes.

CANDIDATES: for each $[n, k]$ code C let $S(C)$ be the set of all $m(n)$-tuples of binary vectors with length n.

OBJECTIVE FUNCTION: Given any code C and $m(n)$-tuple Y, let $m(C,Y) = \mathrm{cov}(C,Y)$.

We start with a lemma.

Lemma 4.24 *Given a function $f : \mathbb{Z}^+ \to \mathbb{N}$ and the following properties:*

1. *There exists an $\epsilon < 1$ such that f is in $o(n^\epsilon)$.*

2. *There exists a positive integer valued function $g(n)$ that is polynomially bounded by n such that $f(ng(n)) < g(n)$ for all sufficiently large n.*

3. *There exists an $\epsilon < 1$ such that f is not in $\omega(n^\epsilon)$.*

Then $(1) \Rightarrow (2) \Rightarrow (3)$.

Proof: $(1) \Rightarrow (2)$ Given that f is in $o(n^\epsilon)$ for some $\epsilon < 1$, then for some such ϵ, $f(n) < n^\epsilon$ for all n sufficiently large. Let $d = \lceil \epsilon/(1-\epsilon) \rceil$ and $g(n) = n^d$. So $f(ng(n)) < (ng(n))^\epsilon \leq n^{\epsilon(d+1)} \leq g(n)$ for all n sufficiently large.

$(2) \Rightarrow (3)$ Assume not, then for every $\epsilon < 1$, f is in $\omega(n^\epsilon)$. Since g is positive integer valued then $n \leq ng(n)$ for any integer n. Also since g is polynomially bounded there exists a d such that $g(n) \leq n^d$ for n sufficiently large. Let ϵ be a real number less than one such that $\epsilon/(1-\epsilon) > d$. This is possible since $\epsilon/(1-\epsilon)$ tends to infinity as ϵ tends to one from below. Because $f(n)$ is in $\omega(n^\epsilon)$ then $f(n) > n^\epsilon$ for all n sufficiently large. Thus, $(ng(n))^\epsilon < f(ng(n)) < g(n)$ for large n. However, this implies that $n^\epsilon < g(n)^{1-\epsilon}$ and $n^{\epsilon/(1-\epsilon)} < g(n) \leq n^d$ for n sufficiently large. This is a contradiction as $\epsilon/(1-\epsilon) > d$. □

We will call any function that satisfies property 2 *asymptotically sublinear.*

Theorem 4.25 *If $P \neq NP$ and f is an asymptotically sublinear function, then there does not exist a polynomial time approximation algorithm A_1 for ACR_1 that guarantees $t_1(C) - A_1(C) \leq f(n)$.*

Proof: Suppose that P is not equal to NP and that there is such an algorithm A_1. Given a (n, K) binary code C and function f that satisfies the hypothesis, let C' be the direct product of C with itself $g(n)$ times, $C' = C^{g(n)}$. Since $g(n)$ is polynomially bounded in n, the C' can be constructed in polynomial time. We will see that we can use the approximation algorithm A_1 on C' to find the 1-covering radius of C exactly and in polynomial time for all n sufficiently large. This contradicts Theorem 4.17.

Let $sA_1(C') = \mathbf{s}_1|\mathbf{s}_2|\dots|\mathbf{s}_{g(n)}$ be the candidate solution found by A_1 on input C', where $\mathbf{s}_1, \dots, \mathbf{s}_{g(n)}$ all have length n. We have $A_1(C') = \text{cov}(C', sA_1(C')) = \sum_{i=1}^{g(n)} \text{cov}(C, \mathbf{s}_i)$. Now let $\mathbf{s}_{\max}$ be the $\mathbf{s}_i$ that has the greatest distance from C. In other words, $\text{cov}(C, \mathbf{s}_{\max}) = \max_{1 \leq i \leq g(n)}(\text{cov}(C, \mathbf{s}_i))$. Because $\text{cov}(C, \mathbf{s}_i)$ can be computed in time $\mathcal{O}(Kn)$, $\text{cov}(C, \mathbf{s}_{\max})$ can be found in time $\mathcal{O}(g(n)Kn)$ i.e. in polynomial time. Since $\mathbf{s}_{\max}$ has the greatest distance from C it makes the largest contribution to $A_1(C')$ and $\mathrm{cov}(C, \mathbf{s}_{\max}) \geq \lceil A_1(C')/g(n) \rceil$. We have the guarantee

$$t_1(C') - A_1(C') \leq f(n') \tag{4.7}$$

where $n' = ng(n)$ is the length of C'. Also, since C' was formed by taking the direct product of C $g(n)$ times, $t_1(C') = g(n)t_1(C)$. Using the fact that $f(ng(n)) < g(n)$ the inequality (4.7) can be written as:

$$t_1(C) - \frac{A_1(C')}{g(n)} \leq \frac{f(ng(n))}{g(n)} < 1$$

for all n sufficiently large. Therefore,

$$\begin{array}{rcl} t_1(C) & < & 1 + \frac{A_1(C')}{g(n)} \\ & \leq & 1 + \left\lceil \frac{A_1(C')}{g(n)} \right\rceil \\ & \leq & 1 + \mathrm{cov}(C, \mathbf{s}_{\max}). \end{array}$$

Since $\mathrm{cov}(C, \mathbf{s}_{\max})$ must be an integer and is also at most $t_1(C)$, it must be equal to $t_1(C)$. Thus we have found the 1-covering radius of C in polynomial time. □

There are some interesting guarantees that Theorem 4.25 rules out. For example, $f(n) = c$ for some constant integer c is asymptotically sublinear, with $g(n) = c+1$. So no polynomial time approximation algorithm for ACR_1 can guarantee that its error is at most a fixed constant. Also, $f(n) = n^\epsilon$ for any $\epsilon < 1$ is asymptotically sublinear, with $g(n) = n^\delta$ where $\delta > \epsilon/(1-\epsilon)$. Furthermore, as we have seen in Lemma 4.24, just having f in $o(n^\epsilon)$ for some $\epsilon < 1$ is sufficient. On the other hand, any approximation algorithm can guarantee its error is no more than n. Also, Theorem 4.25 and Lemma 4.20 give a similar result for the m-covering radius.

Theorem 4.26 *If P $\neq$ NP, $f(n)$ is an asymptotically sublinear function, and $m(n)$ is polynomially bounded in n, then there does not exist a polynomial time approximation algorithm $A_{m(n)}$ for $ACR_{m(n)}$ that guarantees $t_{m(n)}(C) - A_{m(n)}(C) \leq f(n)$.*

Proof: Suppose that P is not equal to NP and there is such an algorithm $A_{m(n)}$. Given a (n, K) binary code C, let $C' = C \times S$ where S is the code defined in Section 3. We can approximate the 1-covering radius of C by applying the algorithm $A_{m(n)}$ to C'. Namely, let the algorithm A_1 have $A_1(C) = A_{m(n)}(C') - r$ where r is the length of S and $sA_1(C)$ be the first n coordinates of $sA_{m(n)}(C')$. Lemma 4.20 says that $t_{m(n)}(C') = t_1(C) + r$. Thus,

$t_{m(n)}(C') - A_{m(n)}(C') \leq f(n)$ implies that $t_1(C) - A_1(C) \leq f(n)$. This contradicts Theorem 4.25. □

There are analogous results for the linear case.

Theorem 4.27 *If NP $\neq \Sigma_2^p$ and $f(n)$ is an asymptotically sublinear function, then there does not exist a polynomial time approximation algorithm B_1 for ACRL$_1$ that guarantees $t_1(C) - B_1(C) \leq f(n)$.*

Proof: This proof is similar to the proof for the unrestricted case. Suppose that NP is not equal to Σ_2^p and that there is such an algorithm B_1. Given a parity check matrix $\mathbf{H}$ for a $[n, k]$ binary linear code C and a function f that satisfies the hypothesis, let C' be the direct product of C with itself $g(n)$ times, $C' = C^{g(n)}$, and let $\mathbf{H}'$ be its parity check matrix. We will see that we can use the approximation algorithm B_1 on $\mathbf{H}'$ to find the covering radius of C exactly for all sufficiently large lengths.

Let $sB_1(C') = \mathbf{s}_1|\mathbf{s}_2|\ldots|\mathbf{s}_{g(n)}$ be the candidate solution found by B_1 on input $\mathbf{H}'$, where $\mathbf{s}_1, \ldots, \mathbf{s}_{g(n)}$ all have length n. We have $B_1(C') = \text{cov}(C', sB_1(C')) = \sum_{i=1}^{g(n)} \text{cov}(C, \mathbf{s}_i)$. Now let $\mathbf{s}_{\max}$ be the $\mathbf{s}_i$ that has the greatest distance from C. Since such an $\mathbf{s}_{\max}$ takes the largest radius to contain all of C, it makes the largest contribution to $B_1(C')$ and $\text{cov}(C, \mathbf{s}_{\max}) \geq \lceil B_1(C')/g(n) \rceil$. Now the general question of whether $\text{cov}(C, \mathbf{y}) < w$ is in NP as it can be verified in polynomial time. Thus the question of whether $\text{cov}(C, \mathbf{y}) \geq w$ and finding $\mathbf{s}_{\max}$ is in co-NP. As in Theorem 4.25, the guarantee $t_1(C') - B_1(C') \leq f(n')$ implies that $\text{cov}(C, \mathbf{s}_{\max})$ is the covering radius of C. Therefore, finding the covering radius of a linear code can be reduced to a problem in co-NP. Since LBLC$_1$ is Σ_2^p-complete and Σ_2^p contains co-NP, co-NP is equal to Σ_2^p. Also, Σ_2^p contains NP. So co-NP contains NP and is therefore equal to NP. This contradicts our assumption that NP is not equal to Σ_2^p. □

Using the same method as in Theorem 4.26 we get a similar result for the m-covering radius.

Theorem 4.28 *If NP $\neq \Sigma_2^p$, $f(n)$ is an asymptotically sublinear function, and $m(n)$ is polynomially bounded in n, then there does not exist a polynomial time approximation algorithm $B_{m(n)}$ for ACRL$_{m(n)}$ that guarantees $t_{m(n)}(C) - B_{m(n)}(C) \leq f(n)$.*

Chapter 5

Parameterized Complexity

Another view on the complexity of computational problems was formulated by Downey and Fellows [5]. They note that many problems consist of a pair of inputs such as:

Graph Genus Takes as input a pair (G, k) where G is a graph and k is a positive integer. The question is whether G can be embedded onto the surface of genus k.

Vertex Cover Takes as input a pair (G, k) as in Graph Genus. However, the question is whether there is a set S of k vertices in G such that every edge in G has at least one element in S.

Dominating Set Take as input a pair (G, k) as in Graph Genus. However, the question is whether there is a set S of k vertices such that every vertex either is a member of S or has an adjacent vertex that is.

Weighted CNF Satisfiability Takes as input a pair (f, k) where f is a Boolean formula in conjunctive normal form and k a positive integer. The question is whether there is a truth assignment of weight k that satisfies f.

All of the above problems are known to be NP-complete. However it may be the case that in practice only a small range of the parameters are really important. Therefore the NP-completeness of the general problem may be misleading.

For the first two examples there is a constant c such that for every fixed parameter k the problem can be solved in time $\mathcal{O}(n^c)$. For Graph Genus it was shown that we may take c to be 3 by Robertson and Seymour [25]. For Vertex Cover, from the work of Buss and Goldsmith [2], we may take c to be 1.

However the situation is different with Dominating Set and Weighted CNF Satisfiability. For both of these problems the best known algorithms run in time $\mathcal{O}(n^{k+1})$ for fixed k. In other words they are no faster than exhaustive search. Parameterized complexity theory gives a framework to discuses such differences. The following are some of the main definitions of that theory.

A parameterized problem (language) is a set $L \subseteq \Sigma^* \times \Sigma^*$ where Σ is a fixed alphabet. Often for ease of reading L is considered to be a subset of $\Sigma^* \times \mathbb{N}$. The set $L_k = \{y : (y, k) \in L\}$ is called the k^{th} slice of L and denotes an associated fixed parameter problem.

The parameterized problem L is fixed parameter tractable (FPT) if and only if there exists a computable function f, a constant c, and a deterministic algorithm M such that for all x, k, (x, k) is in L if and only if $M(x, k)$ accepts and the running time of $M(x, k)$ is less than $f(k)|x|^c$.

The parameterized problem L is reducible to L', denoted $L \leq_{fpt} L'$, if and only if there exists a deterministic algorithm M, functions $f, g : \mathbb{N} \to \mathbb{N}$ and a constant c such that:

$$\begin{aligned}
&M : (G, k) \mapsto (G', k') \\
&M((G, k)) \text{ runs in time } \leq g(k)|G|^c \\
&k' \leq f(k) \\
&(G, k) \in L \Leftrightarrow (G', k') \in L'.
\end{aligned}$$

Lemma 5.1 (Downey and Fellows [5]) *If the parameterized problem L reduces to the problem L' and L' is FPT the L is FPT.*

Now consider again the problem of Weighted CNF Satisfiability.

NAME: Weighted CNF SAT Weighted CNF Satisfiability

INPUT: A Boolean formula E in CNF

PARAMETER: A non-negative integer k

QUESTION: Does E have a satisfying truth assignment of weight k?

Similarly we can define a weighted version of 3-CNF SAT. Furthermore it is known that CNF SAT and 3-CNF SAT are many-one equivalent. CNF SAT can be shown to be many-one reducible to 3-CNF SAT by a passing argument. That is, a given clause can be turned into an equivalent set of clauses by adding new literals. However this would not give a parametric reduction from WEIGHTED CNF SAT to WEIGHTED 3-CNF SAT since a k weight assignment for the CNF formula could be transformed into any other weight assignment for the corresponding 3-CNF formula. Downey and Fellows conjecture that in fact there is no parametric reduction from WEIGHTED CNF SAT to WEIGHTED 3-CNF SAT. Consider the following parameterized problems:

NAME: Weighted t-POS SAT Weighted t-product of sum satisfiability

INPUT: A Boolean formula E in product of sums of products of... with t alternations.

PARAMETER: A non-negative integer k

QUESTION: Does E have a satisfying truth assignment of weight k?

Downey and Fellows defined classes:

$$\begin{array}{lll}
W[1] & = \text{set of languages fpt-equivalent to} & \text{Weighted 3-CNF SAT} \\
W[2] & = \text{set of languages fpt-equivalent to} & \text{Weighted CNF SAT} \\
 & \vdots & \\
W[t] & = \text{set of languages fpt-equivalent to} & \text{Weighted t-POS SAT} \\
 & \vdots & \\
W[SAT] & = \text{set of languages fpt-equivalent to} & \text{Weighted SAT.}
\end{array}$$

The classes form a hierarchy

$$W[1] \subseteq W[2] \subseteq \cdots \subseteq W[SAT].$$

Downey and Fellows further conjectured that these classes are distinct.

The parameterized complexity of some coding theoretic problems has also been studied. Consider the problems

NAME: Maximum Likelihood Decoding

INPUT: A binary $m \times n$ matrix $\mathbf{H}$, a target vector $\mathbf{s} \in \boldsymbol{F}^m$, and an integer $k > 0$.

PARAMETER: k

QUESTION: Is there a set of at most k columns of $\mathbf{H}$ that sum to $\mathbf{s}$?

and

NAME: Weight Distribution

INPUT: A binary $m \times n$ matrix $\mathbf{H}$ and an integer $k > 0$.

PARAMETER: k

QUESTION: Is there a set of k columns of $\mathbf{H}$ that sum to the zero vector?

Downey et. al. [27] showed that both are W[1]-hard and that they belong to W[2]. A simple modification to the proof of Theorem 4.17 shows that the following parameterized version of the covering radius problem is W[1]-hard.

NAME: PLBC Lower bounding the covering radius

INPUT: A binary code $C \subseteq \boldsymbol{F}^n$, and integer w and an integer k.

PARAMETER: k

QUESTION: Is there a vector $\mathbf{y} \in \boldsymbol{F}^n$ such that $\mathrm{wt}(\mathbf{y}) = k$ and $\forall \mathbf{c} \in C$ $\mathrm{dist}(\mathbf{y}, \mathbf{c}) \geq w$?

Theorem 5.2 *Weighted 3-CNF SAT $\leq_{ftp}$ PLBC*

Proof: Let $E = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ and k be an instance of Weighted 3-CNF SAT using the variables $x_1, x_2, \ldots, x_n$. The primary modification is in how we define $\mathbf{z}(C_j)$. For each clause C_j define the vector $\mathbf{z}(C_j) \in \boldsymbol{F}^{2n}$ where:

$$\begin{aligned} z_{2i-1} &= & z_{2i} &= 1 \quad \text{if } C_j \text{ contains } \overline{x}_i \\ z_{2i-1} &= & z_{2i} &= 0 \quad \text{if } C_j \text{ contains } x_i \\ z_{2i-1} &= & 0, z_{2i} &= 1 \quad \text{otherwise} \end{aligned}$$

We then define the code $C \subseteq \boldsymbol{F}^{2n+2}$ to be the set $\{\mathbf{z}(C_j)|00 : i \leq j \leq m\} \cup Y_{2(n+1)}$, $w = n+1$ and $k' = 2(k+1)$. The code C and integer w form the desired instance of PLBC. The cardinality of C is $2n+4+m$ so this construction can be done in polynomial-time with respect to the size of E.

All that remains to see that this is indeed a FPT reduction is to show that positive and negative instances coincide with respect to the parameters. Assume that E can be satisfied by an assignment with weight k. Any satisfying assignment can be represented by a vector $\mathbf{v} \in \boldsymbol{F}^n$, where the i^{th} coordinate of $\mathbf{v}$ is one if x_i is assigned to be true and zero otherwise. Now let $\mathbf{v}^* = v_1 v_1 v_2 v_2 \cdots v_n v_n | 11 \in \boldsymbol{F}^{2n+2}$ and $\overline{\mathbf{v}}^*$ be its complement. Both vectors $\mathbf{v}^*$

and $\overline{\mathbf{v}}^*$ are doubled so $\text{dist}(\mathbf{c}, \overline{\mathbf{v}}^*) \leq n + 1 = w$ for every $\mathbf{c} \in Y_{2n}$ by Lemma 4.16. In each clause there is at least one literal which is set true by the assignment corresponding to $\mathbf{v}$ as it satisfies E. So $\text{dist}(\mathbf{z}(C_j)|00, \overline{\mathbf{v}}^*) \leq 2 + 2 + 0 + (n - 3) = w$. So for every $\mathbf{c} \in C$, $\text{dist}(\mathbf{c}, \mathbf{v}^*) \geq 2n + 2 - (n + 1) = n + 1$. Thus C has a deep hole with distance at least w and weight $k' = 2(k + 1)$.

Assume that C has a deep hole with distance at least w and weight k'. Then there exists a vector $\mathbf{v}^* \in \boldsymbol{F}^{2n+2}$ such that $\text{dist}(\mathbf{v}^*, \mathbf{c}) \geq n+1$. As $\text{dist}(\overline{\mathbf{v}}^*, \mathbf{c}) \leq n+1$ for every $\mathbf{c}$ in C, $\overline{\mathbf{v}}^*$ and $\mathbf{v}^*$ are doubled. Let $\overline{\mathbf{v}}^* = v_1 v_1 v_2 v_2 \ldots v_n v_n v_{n+1} v_{n+1}$. Then $\text{dist}(v_1 v_1 v_2 v_2 \ldots v_n v_n, \mathbf{z}(C_j)) \leq n + 1$. Therefore there exists an i such that $z_{2i-1} = z_{2i} = v_i$. So the assignment of weight k corresponding to the vector $\mathbf{v}^*$ satisfies E. □

Chapter 6

Open Questions

Our technique for the finding the 2 covering radius of the 2-error correcting BCH code could be applied to other codes. For instance you could prove that a code satisfied the hypothesis of a theorem like the following.

Theorem 6.1 *Suppose C is a linear code with $t_1(C) = r$. If for any two vectors $\mathbf{x}$ and $\mathbf{y}$, with $a = n - dist(\mathbf{x}, \mathbf{y}) \leq r + 1$, let $\mathbf{z} = \overline{\mathbf{x}} + \mathbf{y}$, there exists codewords $\mathbf{u}$ and $\mathbf{v}$ that satisfy the following properties:*

1. $b \triangleq dist(\mathbf{u}, \mathbf{x}) \leq r$
2. $\lfloor (n - r + 2)/2 \rfloor - a + b \leq wt(\mathbf{v}) \leq \lceil (n + r - 2)/2 \rceil + b$
3. $supp(\mathbf{u} + \mathbf{x}) \subseteq supp(\mathbf{v})$
4. $supp(\mathbf{z}) \cap supp(\mathbf{u} + \mathbf{x} + \mathbf{v}) = \emptyset$

then $t_2(C) \leq \left\lceil \frac{n+r-2}{2} \right\rceil$.

Proof: Consider two vectors $\mathbf{x}$ and $\mathbf{y}$ with $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = n - a$, where $a \geq r + 2$. There exists a vector $\mathbf{v}'$ with distance at most $\lceil (n - r - 2)/2 \rceil$ to both $\mathbf{x}$ and $\mathbf{y}$ and there exists a codeword $\mathbf{v}$ with $\mathrm{dist}(\mathbf{v}, \mathbf{v}') \leq r$. Thus the distance from $\mathbf{v}$ to both $\mathbf{x}$ and $\mathbf{y}$ is at most $\lceil (n - r - 2)/2 \rceil + r = \lceil (n + r - 2)/2 \rceil$.

Consider two vectors $\mathbf{x}$ and $\mathbf{y}$ with $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = n - a$, where $a \leq r + 1$. Let $\mathbf{u}$, $\mathbf{v}$, $\mathbf{z}$, and $\mathbf{b}$ be as in the hypothesis. Then

$$\begin{aligned}
\mathrm{dist}(\mathbf{x}, \mathbf{u} + \mathbf{v}) &= \mathrm{wt}(\mathbf{u} + \mathbf{v} + \mathbf{x}) \\
&= \mathrm{wt}(\mathbf{v}) - \mathrm{wt}(\mathbf{u} + \mathbf{x}) \\
&= \mathrm{wt}(\mathbf{v}) - b \\
&\leq \lceil (n + r - 2)/2 \rceil ,
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{dist}(\mathbf{y}, \mathbf{u} + \mathbf{v}) &= \mathrm{wt}(\mathbf{u} + \mathbf{v} + \mathbf{y}) \\
&= \mathrm{wt}(\mathbf{u} + \mathbf{v} + \overline{\mathbf{x}} + \mathbf{z}) \\
&= n - \mathrm{wt}(\mathbf{u} + \mathbf{v} + \mathbf{x} + \mathbf{z}) \\
&= n - \mathrm{wt}(\mathbf{u} + \mathbf{v} + \mathbf{x}) - \mathrm{wt}(\mathbf{z}) \\
&= n - \mathrm{wt}(\mathbf{v}) + b - a \\
&\leq \lceil (n + r - 2)/2 \rceil .
\end{aligned}$$

□

To prove the hypothesis of such a theorem in the case of the 2-error correcting BCH code we took advantage of the fact that both the covering radius and the dual distribution were known. In the case of the dual distribution we needed it to be concentrated around $n/2$. In other words no small or large weight codewords could be in the dual. Other codes have this properties as well, the 3-error correcting BCH code for example.

While we have focused on the binary case there are general q-ary definitions of codes, Hamming distance, covering radius and so on. As such our techniques may have generalizations as well. A good code to look at may be the q-ary Hamming code since it is known to have covering radius one.

Our completeness and approximation results require m to be polynomially bounded in the length of the code. The problem of finding the covering radius when $m = 2^n$ is trivial as $t_{2^n}(C) = n$ for any code C. Furthermore, let $K_1(n, 1)$ be the size of the smallest code with 1-covering radius 1, then for $m \geq K_1(n, 1)$, the m-covering radius can be found in polynomial time. In this case the m-covering radius is at least $n - 1$. To see this, let S be a set of m vectors with 1-covering radius equal to 1. Then if $\mathbf{c}$ is any codeword, S contains a vector whose distance is at most 1 from the complement of $\mathbf{c}$, hence the set S has distance at least $n-1$ from $\mathbf{c}$. Given a code C, if $|C| \leq m$ then $t_m(C) = n$ since we can take $S = \{\overline{\mathbf{c}} : \mathbf{c} \in C\}$ and $\text{cov}(C, S) = n$. For $|C| > m$, any m-tuple S must not contain the complement of at least one codeword $\mathbf{c} \in C$. Hence $\text{dist}(\mathbf{c}, \mathbf{v}) \leq n-1$ for every $\mathbf{v} \in S$, and so $\text{cov}(C, S) \leq n-1$. So for $m \geq K_1(n, 1)$, $t_m(C) = n$ if $|C| \leq m$ and $t_m = n-1$ otherwise. For all n, $K_1(n, 1) \leq 2^n - n$ since the set of size $2^n - n$ that contains every vector except for $n-1$ of the vectors of weight $n-1$ and the all one vector has 1-covering radius 1. Therefore for large m the problem of finding the m-covering radius can be done in polynomial time. This leads to the question of how the complexity varies with respect to m when m is not polynomially bounded in the length of the code.

Bibliography

[1] A. Barg. Complexity issues in coding theory. In V. Pless and W. Huffman, editors, *Handbook of Coding Theory*, volume 1, pages 649–754. Elsevier Science B.V., 1998.

[2] J. Buss and J. Goldsmith. Nondeterminism within p. *SIAM Journal on Computing*, 22:560–572, 1993.

[3] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes*. Elsevier Science B.V., 1997.

[4] S. Cook. The complexity of theorem proving procedures. In *Proceedings of the Third ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[5] R. Downey and M. Fellows. Fixed parameter tractability and completeness i: Basic theory. *SIAM Journal on Computing*, 24:873–921, 1995.

[6] M. Frances and A. Litman. On covering problems of codes. *Theory of Computing Systems*, 30(2):113–119, 1997.

[7] M. Garey and D. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completeness*. Freeman, 1979.

[8] W. Peterson D. Gorenstein and N. Zierler. Two-error correcting bose-chaudhury codes are quasi-perfect. *Information and Control*, 3:291–294, 1960.

[9] I. Honkala. Personal communication with A. Klapper.

[10] I. Honkala and A. Klapper. Multicovering bounds from relative covering radii. *SIAM Journal on Discrete Math*, pages 228–234, 2002.

[11] A. Klapper. The multicovering radii of codes. *IEEE Transactions on Information Theory*, 43:1372–1377, 1997.

[12] A. Klapper. On the existence of secure keystream generators. *Journal of Cryptology*, 14:1–15, 2001.

[13] A. Klapper. Improved multicovering bounds from linear inequalities and supercodes. *IEEE Transactions on Information Theory*, 50:532–536, 2004.

[14] D. Kleitman and J. Spencer. Families of k-independent sets. *Discrete Mathematics*, 6:255–262, 1973.

[15] I. Krasikov and S. Litsyn. On spectra of bch codes. *IEEE Transactions on Information Theory*, 41:786–788, 1995.

[16] R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[17] L. Levin. Universal sorting problems. *Problems of Information Transmission*, 9:265–266, 1973. Translation of original that appeared in Problemy Peredaci Informacii.

[18] F. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes.* Elsevier Science B.V., 1977.

[19] E. Berlekamp R. McEliece and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24:384–386, 1978.

[20] A. McLoughlin. The complexity of computing the covering radius of a code. *IEEE Transactions on Information Theory*, 30:800–804, 1984.

[21] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential time. In *Proceedings of the 13th Annual Symposium on Switching and Automata Theory*, pages 125–129, 1972.

[22] I. Dumer D. Micciancio and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49:22–37, 2003.

[23] V. Pless and W. Huffman, editors. *Handbook of Coding Theory*, volume 1. Elsevier Science B.V., 1998.

[24] V. Pless and W. Huffman, editors. *Handbook of Coding Theory*, volume 2. Elsevier Science B.V., 1998.

[25] N. Robertson and P. Seymour. Graph minors xiii: The disjoint paths problem. *Journal of Combinatorial Theory Series B*, 63:65–110, 1995.

[26] A. Vardy. The inherent intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43:1757–1766, 1997.

[27] R. Downey M. Fellows A. Vardy and G. Whittle. The parameterized complexity of some fundamental problems in coding theory. *SIAM Journal on Computing*, 29:545–570, 1999.

[28] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theory of Computer Science*, 3:23–33, 1976.

Andrew E. Mertz

Biographical Information

Born in Galax, Virgina on December 26 1974

Education

2005	Expected PhD in Computer Science, University of Kentucky • Concentration: Error Correcting Codes • Advisor: Andrew Klapper, University of Kentucky
2000	Master of Arts in Mathematics, University of Kentucky
1997	Bachelor of Science in Physics with Honors, New Mexico Tech
1997	Bachelor of Science in Mathematics with Honors, New Mexico Tech

Work Experience

2004-Present	Assistant Professor, Department of Mathematics and Computer Sciences, Eastern Illinois University, Charleston, IL
1997-2004	Teaching and Research Assistant, University of Kentucky
1995-1997	Optics Technician, Aerotherm, White Sands Missile Range, NM

Journal Articles

On the Complexity of Multicovering Radii, by A. Mertz, IEEE Transactions on Information Theory, vol. 50, pp. 1804–1808, August 2004

Refereed Conferences

The Multicovering Radii of Linear Codes with 1-Covering Radius One, by A. Mertz, International Symposium on Information Theory 2003.

The Multicovering Radii of the Even Weight Codes, by A. Mertz, International Symposium on Information Theory 2001.

Awards

1997-2004	Lyman T. Johnson Academic Achievement Fellowship
1996	Inducted into the physics honor society, Sigma Pi Sigma
1992-1997	Presidential Scholarship, New Mexico Tech

Academic Service

Reviewed papers for IEEE Transactions on Information Theory
Reviewed papers for Crypto '99, International Cryptology Conference
Served on the nomination committee for Sigma Pi Sigma, New Mexico Tech chapter

Professional Memberships

Institute of Electrical and Electronics Engineers
International Association for Cryptologic Research
American Mathematical Society
American Institute of Physics