MASTER

Lower bound on the minimum distance of cyclic codes

Hendradi, Erwin

Award date:
2007

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Lower bound on the Minimum Distance of Cyclic Codes

by
Erwin Hendradi

Supervisors:

dr. G.R. Pellikaan (TU/e)
dr.ir. H.D.L. Hollmann (Philips Research Labs.)

Eindhoven, August 2007

Abstract

The minimum distance of cyclic codes is an important parameter that determines how many errors that code can correct. In general, it is not easy to determine the true minimum distance of cyclic codes. Let C be an q-ary cyclic code of length n. The easiest way to compute the minimum distance of C is compute the distance between two codewords in C and take the minimum value. But this method is inefficient, since it costs a lot of memory, when working with a large C.

In 1960, R.C.Bose, D.K.Chaudhuri and A. Hocquenghem invented an algorithm to determine a lower bound of the cyclic code by using the set of zeros of C. And in 1972, C.R.P. Hartmann and K.K. Tzeng, generalized the BCH bound. In 1982, C. Roos generalized the HT bound. In 1986, J.H. van Lint and R.M. Wilson introduced the Shift bound to determine a lower bound on the minimum distance of C.

We implemented these bounds using C++. For the BCH, HT, and Roos bounds, the algorithms follow directly from the definition of the bound itself. For the Shift bound, we implemented a backtracking algorithm to compute the Shift bound. We give an estimation on the complexity of the algorithm only for a special case. To speed up the computation process of our backtracking algorithm, we apply the branch-and-bound technique and the Greedy algorithm. And then compare the results with the Square Root bound for the Quadratic Residue codes. We consider these bounds for all binary cyclic codes of length 45 and 73.

Acknowledgements

This report is the result of two years of training under the guidance of my supervisors, Henk Hollmann and Ruud Pellikaan. I would like to thank Henk Hollmann for letting me stay at Philips Research Labs. during the internship phase of my master program. The internship project also included as part of my master project. Ruud Pellikaan has gone through numerous drafts of this report and the report of the internship project, and his constructive comments have led to many improvements and additions. Also I thank them for their endless patience in their interaction with me, for the fact that they always had time for me and that they constantly challenged me to keep on learning.

I also would like to thank Prof. Arjeh Cohen for letting me to continue my study at TU/e. I also would like to thank Hans Cuypers for the constant support throughout all these years, especially in coordinating my study program.

Table of contents

1
Introduction

The theory of error-detecting and correcting codes is a branch of engineering and mathematics which deals with the reliability on transmission and storage of data. Noise of any form of interference frequently causes data to be distorted. This is an undesirable but inevitable situation. To solve this problem, add redundancy to the original message in such a way that it is possible for the receiver to detect the error and correct it, recovering the original message. An effectiveness of a code for error-detection or error-correction is measured by the minimum distance of a code.

Let C be an q-ary cyclic code of length n. In general, it is not easy to determine the true minimum distance of cyclic codes. The easiest way to compute the minimum distance of C is compute the distance between two codewords in C and take the minimum value. But this method is inefficient, since it costs a lot of memory, when working with a large C.

In 1960, R.C.Bose, D.K.Chaudhuri and A. Hocquenghem invented an algorithm to determine a lower bound of the cyclic code by using the set of zeros of C. They determine the lower bound for the minimum distance of a cyclic code by looking at the largest consecutive element set in the set of zeros of C.

In 1972, C.R.P. Hartmann and K.K. Tzeng, generalized the BCH bound. If the BCH bound only looking at single consecutive element set, then the HT bound looking at several consecutive element sets in the set of zeros of C. In 1982, C. Roos generalized the HT bound. In 1986, J.H. van Lint and R.M. Wilson introduced the Shift bound to determine a lower bound on the minimum distance of C.

As far as we concerns, we did not find any exact algorithm on how to compute the Shift bound. So, our contribution is a development on the algorithm to compute the Shift bound as well as implements the other bounds in our program. We use the high level language C++ to implements these bounds.

For the BCH, HT, HTR and Roos bounds, the algorithms follow directly from the definition of the bound themselves. For the Shift bound, we implemented a backtracking algorithm to compute the Shift bound. We also give an estimation on the complexity of the algorithm only for a special case, since in general it is quite difficult.

To speeding-up the computation process and improving the efficiency of our program, first we modified the Shift bound problem and then we apply the *Branch-And-Bound* procedure. However, in the case of large n, this was not enough. So to make the computation even faster, we implemented the Greedy algorithm.

We provide a comparison of the results with the Square Root bound for the Quadratic Residue codes. We also consider these bounds for all binary cyclic codes of length 45 and 73.

2

Linear codes

In this chapter, we will review our basic knowledge on error-correcting codes, particularly about linear codes and its one most important class, cyclic codes. We borrowed and adapted notations and definitions from [11], [13], [19], and [21].

2.1. Block codes

Let n be fixed, and let Q be an alphabet of cardinality q. The set of Q-ary n-tuples is denoted by Q^n.

Definition 2.1. The *Hamming distance* $d(\mathbf{x}, \mathbf{y})$ between $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ in Q^n is given by

$$d(\mathbf{x}, \mathbf{y}) = |\{1 \leq i \leq n \mid x_i \neq y_i\}|.$$

In other words, $d(\mathbf{x}, \mathbf{y})$ denotes the number of coordinates, where $\mathbf{x}$ and $\mathbf{y}$ differ.

Definition 2.2. The weight $\mathrm{wt}(\mathbf{x})$ of $\mathbf{x}$ is defined by

$$\mathrm{wt}(\mathbf{x}) = d(\mathbf{x}, \mathbf{0}),$$

where $\mathbf{0} = (0, \ldots, 0)$.

Remark 2.3. Note that the function $d(\mathbf{x}, \mathbf{y})$ is a *metric* and defines a distance in Q^n, since it is always non-negative and satisfies

1. $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$,
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}, \mathbf{y} \in Q^n$,
3. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$ for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in Q^n$.

A q-ary *block code* of C of length n is any nonempty subset of Q^n. The elements of C are called *codewords*. If $|C| = 1$ or $C = Q^n$, the code is called *trivial*.

If we use a channel with the property that an error in position i does not influence other positions and a symbol in error can be each of the remaining $q - 1$ symbols with equal probability, then the Hamming-distance is a good way to measure the error content of a received message.

Definition 2.4. The *minimum distance* d of a code C, where $|C| \geq 2$, is given by

$$d(C) = \min\{\, d(\mathbf{x}, \mathbf{y}) \;:\; \mathbf{x} \in C, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y} \,\}.$$

The *minimum weight* of C is

$$\min\{\mathrm{wt}(\mathbf{x}) \,|\, \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\}.$$

Define the distance of $\mathbf{x}$ not in C by

$$d(\mathbf{x}, C) = \min\{d(\mathbf{x}, \mathbf{c}) \,|\, \mathbf{c} \in C\}.$$

Next, we are going to introduce the counterpart of minimum distance called the *covering radius*. The covering radius determines how far a received word can be from a closest codeword.

Definition 2.5. If $C \subset Q^n$, then the *covering radius* $\rho(C)$ of C is

$$\max\{d(\mathbf{x}, C) \,|\, \mathbf{x} \in Q^n\}.$$

Definition 2.6. The *sphere* with radius ρ and center $\mathbf{x}$ is defined to be the set

$$B_\rho(\mathbf{x}) = \{\mathbf{y} \in Q^n \,|\, d(\mathbf{x}, \mathbf{y}) \leq \rho\}.$$

If ρ is the largest integer such that spheres $B_\rho(\mathbf{c})$ with $\mathbf{c} \in C$ are disjoint, then $d = 2\rho + 1$ or $d = 2\rho + 2$. The covering radius is the smallest ρ such that spheres $B_\rho(\mathbf{c})$ with $\mathbf{c} \in C$ cover the set Q^n. If these numbers are equal, then the code C is called *perfect*.

Remark 2.7. A code $C \subset Q^n$ with minimum distance $2e+1$ is called a *perfect code* if every $\mathbf{x} \in Q^n$ has distance $\leq e$ to exactly one codeword.

2.2. Linear codes

2.2.1 Definitions

From now on, Q will have the structure of the Galois field $\mathbb{F}_q$, the finite field with $q = p^r$ (p prime) elements. The set of words $\mathbb{F}_q^n$ can be associated with an n-dimensional vector space over $\mathbb{F}_q$. The elements of $\mathbb{F}_q^n$ are vectors, and are also called *words*.

Now that Q^n has the structure of the vector space $\mathbb{F}_q^n$, we can define the most important general class of codes.

Definition 2.8. A *linear code* C of length n is linear subspace of $\mathbb{F}_q^n$. If C has dimension k and minimum distance d, then C its parameter are denoted by $[n, k, d]$.

Note that a q-ary (n, M, d) code C has cardinality M, while a q-ary $[n, k, d]$ code C is linear and it has cardinality q^k.

2.2.2 Properties of linear codes

One way to describe a linear code is by means of k independent basis vectors.

Definition 2.9. A *generator matrix* G of a $[n, k, d]$ code C is a $k \times n$ matrix of which the rows are a basis of C. In other words,

$$C = \{\mathbf{a}G \,|\, \mathbf{a} \in \mathbb{F}_q^k\}.$$

If G is of the form $G = (I_k, P)$, where I_k is the $k \times k$ identity matrix, then the first symbols of a codeword are called *information symbols*. The last $n - k$ coordinates are added to the k information symbols to make error-correction possible.

A second way to describe a linear code is by means of $n - k$ linearly independent equations.

Definition 2.10. A *parity check matrix* H of an $[n, k, d]$ code C is an $(n - k) \times n$ matrix, satisfying

$$\mathbf{c} \in C \text{ if and only if } H\mathbf{c}^t = \mathbf{0}^t.$$

In other words, C is the null space (solution space) of the $n - k$ linearly independent equations $H\mathbf{x}^t = \mathbf{0}^t$.

2.2.3 Minimum distance of linear codes

To determine the minimum distance of a q-ary (n, M, d) code C, we have to compute the distance between all $\binom{M}{2}$ pairs of codewords. But to determine the minimum distance of a linear code, we just need to find the smallest weight of all non-zeros codewords.

Theorem 2.11. *The minimum distance of a linear code C is equal to the minimum non-zero weight in C.*

Proof.

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= d(\mathbf{x} - \mathbf{y}, \mathbf{0}) \\ &= \mathrm{wt}(\mathbf{x} - \mathbf{y}) \end{aligned}$$

and if $\mathbf{x}, \mathbf{y} \in C$ and $\mathbf{x} \neq \mathbf{y}$, then $\mathbf{x} - \mathbf{y} \in C$ and $\mathbf{x} - \mathbf{y} \neq \mathbf{0}$. □

Corollary 2.12. *If a code C has minimum distance d, then C can be used to detect up to $d - 1$ errors or to correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors in any codeword. Here $\lfloor x \rfloor$ represents the greatest integer less than or equal to x.*

2.3. Bounds on Codes

We assume that q is fixed and define an $(n, *, d)$ code as a code with length n and the minimum distance d. We are interested in the maximal number of codewords, i.e. the largest M which can be put in place of the $*$.

Definition 2.13. $A(n, d) = \max\{M \,|\, \text{an}(n, M, d) \text{ code exists }\}$

An (n, M, d) code is called maximal, if $M = A(n, d)$.

Given a channel with certain error probability p. The average number of errors in a received word is np and hence d must grow at least as fast as $2np$ is we wish to correct these errors.

Definition 2.14. $$\alpha(\delta) = \limsup_{n\to\infty} \frac{1}{n} \log_q A(n, \delta n).$$

We are interested in the inverse function $\alpha^{-1}(R)$, with given rate R. The function A and α are not known in general. And in this section we will discuss bounds for both of them only.

We define

$$V_q(n, r) = |B_r(\mathbf{x})| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

To study the function α, we need a generalization of the *entropy function*. We define the entropy function H_q on $[0, \theta]$, where $\theta = \frac{q-1}{q}$, by

$$\begin{aligned} H_q(0) &= 0 \\ H_q(x) &= x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x), \text{ for } 0 < x \leq \theta. \end{aligned}$$

Note that $H_q(x)$ increases from 0 to 1 as x runs from 0 to θ.

Lemma 2.15. *Let* $0 \leq \lambda \leq \theta$, $q \geq 2$. *Then*

$$\lim_{n\to\infty} \frac{1}{n} \log_q V_q(n, \lfloor \lambda n \rfloor) = H_q(\lambda).$$

The proof of Lemma 2.15 was taken from [19].

Proof. For $r = \lfloor \lambda n \rfloor$ the last term of the sum of the right-hand side of $V_q(n, r)$ is the largest. Hence

$$\binom{n}{\lfloor \lambda n \rfloor}(q-1)^{\lfloor \lambda n \rfloor} \leq V_q(n, \lfloor \lambda n \rfloor) \leq (1 + \lfloor \lambda n \rfloor)\binom{n}{\lfloor \lambda n \rfloor}(q-1)^{\lfloor \lambda n \rfloor}.$$

By taking logarithms, dividing by n, and then proceeding as in the proof of Theorem 1.4.5 in [19] the result follows. □

2.3.1 Gilbert bound

Theorem 2.16 (the Gilbert-Varshamov bound). *For* $n \in \mathbb{N}$, $d \in \mathbb{N}$, $d \leq n$, *we have*

$$A(n, d) \geq \frac{q^n}{V_q(n, d-1)}.$$

The proof of Theorem 2.16 was taken from [19].

Proof. Let C be a maximal (n, M, d) code, then C is not contained in any $(n, M+1, d)$ code. This implies that there is no word in Q^n with distance d or more to all words of C. In other words, the spheres $B_{d-1}(\mathbf{c})$, with $c \in C$, cover Q^n. Therefore the sum of their volumes, i.e. $|C|V_q(n, d-1)$ exceeds $q^n = |Q|^n$. □

The Gilbert-Varshamov bound is a lower bound, which is telling us that there exist codes with *good parameters*.

Now, we look at the corresponding bound for α.

Theorem 2.17 (Asymptotic Gilbert bound). *If* $0 \leq \delta \leq \theta$, *then*

$$\alpha(\delta) \geq 1 - H_q(\delta).$$

Proof. By Theorem 2.16 and Lemma 2.15, we have

$$\begin{aligned} \alpha(\delta) &= \limsup_{n\to\infty} \frac{1}{n} \log_q A(n, \delta n) \geq \lim_{n\to\infty} \{1 - \frac{1}{n} \log_q V_q(n, \delta n)\} \\ &= 1 - H_q(\delta). \end{aligned}$$

□

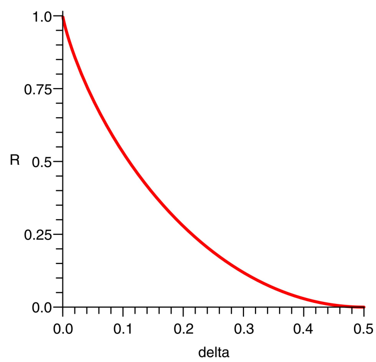


Figure 2.1: Asymptotic Gilbert bound with x-axis is δ and y-axis is $\alpha(\delta)$.

2.3.2 Upper bounds

In this sub-section, we give several upper bounds for $A(n, d)$ that are relatively easy to derive.

Theorem 2.18 (Singleton bound). *For* q, n, $d \in \mathbb{N}$, $q \geq 2$ *we have*

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

Proof. Let (n, M, d) be a code. By puncturing $d - 1$ times, we may obtain an $(n - d + 1, M, 1)$ code, i.e. the M punctured words are different. Hence $M \leq q^{n-d+1}$ □

As immediate result,

Corollary 2.19. *For an* $[n, k]$ *code over* $\mathbb{F}_q$ *we have* $k \leq n - d + 1$.

A code achieving this bound is called an MDS code.

Theorem 2.20 (Asymptotic Singleton bound). *For* $0 \le \delta \le 1$, *we have*

$$\alpha(\delta) \le 1 - \delta.$$

Theorem 2.21 (Plotkin bound). *For* $q,\ n,\ d \in \mathbb{N}$, $q \ge 2$ *and* $\theta = 1 - \frac{1}{q}$, *we have*

$$A(n,d) \le \frac{d}{d - \theta n}, \ \textit{if } d > \theta n.$$

Theorem 2.22 (Asymptotic Plotkin bound).
$$\begin{array}{rcll} \alpha(\delta) & = & 0, & \textit{if } \theta \le \delta \le 1 \\ \alpha(\delta) & \le & 1 - \frac{\delta}{\theta} & \textit{if } 0 \le \delta < \theta. \end{array}$$

Theorem 2.23 (Hamming bound). *If* $q,\ n,\ e \in \mathbb{N}$, $q \ge 2$, $d = 2e + 1$, *then*

$$A(n,d) \le \frac{q^n}{V_q(n,e)}.$$

Proof. The spheres $B_e(\mathbf{c})$, where $\mathbf{c}$ runs through an $(n, M, 2e+1)$ code, are disjoint. Therefore, $M \cdot V_q(n,e) \le q^n$. □

And its asymptotic form is as follows;

Theorem 2.24 (Asymptotic Hamming bound).
$$\alpha(\delta) \le 1 - H_q\left(\frac{1}{2}\delta\right).$$

Proof.
$$A(n, \lceil \delta n \rceil) \le A(n, 2\left\lceil \frac{1}{2}\delta n \right\rceil - 1) \le \frac{q^n}{V_q(n, \lceil \frac{1}{2}\delta n \rceil - 1)}.$$

□

The best known upper bound for $\alpha(\delta)$ is due to R.J. McEliece, E.R.Rodemich, H.C.Rumsey, and L.R.Welch. We only give the asymptotic form. For detail information, see [19] and [11].

Theorem 2.25 (The McEliece-Rodemich-Rumsey-Welch bound I). *For any* (n, M, d) *code,*

$$\alpha(\delta) \le H_2\left(\frac{1}{2} - \sqrt{\delta(1-\delta)}\right).$$

Theorem 2.26 (The McEliece-Rodemich-Rumsey-Welch bound II). *For any* (n, M, d) *code,*

$$\alpha(\delta) \le \min\left\{P(u,\delta) \mid 0 \le u \le 1 - 2\delta\right\},$$

where

$$P(u,\delta) = 1 + g(u^2) - g(u^2 + 2\delta u + 2\delta),$$

and

$$g(x) = H_2\left(\frac{1}{2} - \frac{1}{2}\sqrt{1-x}\right).$$

Notice that $P(1-2\delta,\delta) = H_2\left(\frac{1}{2} - \sqrt{\delta(1-\delta)}\right)$, so

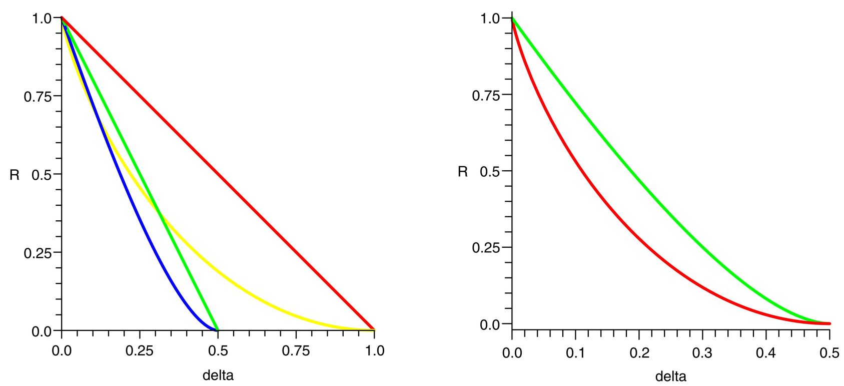
$$\alpha(\delta) \leq H_2\left(\frac{1}{2} - \sqrt{\delta(1-\delta)}\right),$$

and Theorem 2.26 is never weaker than Theorem 2.25. In fact, it turns out that for $\delta \geq 0.273$, $\alpha(\delta)$ is actually equal to

$$H_2\left(\frac{1}{2} - \sqrt{\delta(1-\delta)}\right),$$

and in this range Theorem 2.25 and Theorem 2.26 coincide. For $\delta < 0.273$, Theorem 2.26 is slightly stronger.

Figure 2.2(a) is plots for all asymptotic form of upper bounds. The best lower bound, i.e. Gilbert-Varshamov bound and upper bound, i.e. McEliece-Rodemich-Rumsey-Welch I bound are plotted in Figure 2.2(b). These bounds will be our tools for determining a good code in our study on the minimum distance of cyclic codes.



(a) (b)

Figure 2.2: Plot with x-axis is $\delta = \frac{d}{n}$ and y-axis is $\alpha(\delta) = \frac{k}{n}$.

3

Cyclic codes

In this chapter we will discuss an important class of linear codes. These codes are called *cyclic*. It is our main interest in this final project. We borrowed and adapted notations and definitions from [11], [13], [19] and [21].

3.1. Definitions

We start with the definition of a ring.

Definition 3.1. A *ring* $\mathcal{R}$ is an additive abelian group, together with a multiplication satisfying

$$\begin{aligned} ab &= ba; \\ a(b+c) &= ab + ac; \\ a(bc) &= (ab)c, \end{aligned}$$

and which contains an identity element 1 such that

$$1a = a.$$

Our definition of a ring is also called a commutative ring with identity.

Definition 3.2. An *ideal* $\mathcal{I}$ of a ring $\mathcal{R}$ is a subgroup of $\mathcal{R}$ such that if $a \in \mathcal{I}$, then so is ba for all $b \in \mathcal{R}$.

Definition 3.3. The *polynomial ring* $\mathbb{F}_q[x]$ is the set of all polynomials $f(x)$ with coefficients in $\mathbb{F}_q$.

Definition 3.4. Let $\mathcal{R}$ be a ring and let $\mathcal{I}$ be an ideal in $\mathcal{R}$. Then $\mathcal{R}/\mathcal{I}$ is the *factor ring* or *residue class ring* of $\mathcal{R}$ modulo $\mathcal{I}$. If $\mathcal{R} = \mathbb{F}_q[x]$ and $\mathcal{I} = (x^n - 1)$ is the ideal generated by $x^n - 1$, then $\mathbb{C}_{q,n}$ is a residue class ring, where

$$\mathbb{C}_{q,n} = \mathbb{F}_q[x]/(x^n - 1).$$

The residue class ring $\mathbb{C}_{q,n}$ is represented by the set of all polynomial remainders obtained by long division of polynomial in $\mathbb{F}_q[x]$ by $x^n - 1$. Note that, $\mathbb{C}_{q,n}$ can be represented by the set of polynomials of degree less than n.

Definition 3.5. Let the cyclic shift $\sigma(\mathbf{c})$ of a word $\mathbf{c} = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ be defined as

$$\sigma(\mathbf{c}) = (c_{n-1}, c_0, \ldots, c_{n-2}).$$

Let C be a code of length n in $\mathbb{F}_q^n$. A code C is called *cyclic* if C is linear and for each $\mathbf{c} \in C$, the cyclic shift $\sigma(\mathbf{c})$ is also in C.

Definition 3.6. Consider the map φ between $\mathbb{F}_q^n$ and $\mathbb{C}_{q,n}$

$$\varphi(\mathbf{c}) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1}.$$

Then $\varphi(\mathbf{c})$ is also denoted by $c(x)$.

So, instead of writing $\mathbf{c}$ is in C, we shall write $c(x)$ is in C. Multiplying $c(x)$ with x gives the polynomial corresponding to the cyclic shift. After multiplying $c(x)$ with x, we have to reduce $xc(x)$ modulo $x^n - 1$, i.e replace $xc(x)$ by its remainder after division by $x^n - 1$. So, instead of considering the set of all q-ary polynomials in x, denoted by $\mathbb{F}_q[x]$, we work with the set of the residues of these polynomials modulo $x^n - 1$.

A cyclic shift in $\mathbb{F}_q^n$ corresponds to a multiplication by x in $\mathbb{C}_{q,n}$. And since C is linear by definition, with $c(x)$ in a cyclic code C, not only $xc(x)$ is in C, but also $x^2c(x)$, $x^3c(x)$, etc., and all their linear combinations are in C as well. The most important tool in the description of a cyclic code is the isomorphism between $\mathbb{F}_q^n$ and $\mathbb{C}_{q,n}$. From now on, we identify $\mathbb{F}_q^n$ with $\mathbb{C}_{q,n}$.

Proposition 3.7. *The map φ is an isomorphism of vector spaces. Ideals in the ring $\mathbb{C}_{q,n}$ correspond one-to-one to cyclic codes in $\mathbb{F}_q^n$.*

The proof of Proposition 3.7 was taken from [13].

Proof. The map φ is linear and it maps the i-th standard basis vector of $\mathbb{F}_q^n$ to the coset x^{i-1} in $\mathbb{C}_{q,n}$ for $i = 1, \ldots, n$. Hence φ is an isomorphism of vector spaces.

Let ψ be the inverse map of φ.

1. Let $\mathcal{I}$ be an ideal in $\mathbb{C}_{q,n}$. Let $C = \psi(\mathcal{I})$. Then C is a linear code, since ψ is a linear map. Let $\mathbf{c} \in C$. Then $c(x) = \varphi(\mathbf{c}) \in \mathcal{I}$ and $\mathcal{I}$ is an ideal. So $xc(x)$ is also in I. But,

$$xc(x) = c_{n-1} + c_0 x + c_1 x^2 + \ldots + c_{n-2}x^{n-1},$$

since $x^n = 1$. So, $\psi(xc(x)) = (c_{n-1}, c_0, c_1, \ldots, c_{n-1}) \in C$. Hence, C is cyclic.

2. Conversely, if C is a cyclic code in $\mathbb{F}_q^n$, and let $\mathcal{I} = \varphi(C)$, then $\mathcal{I}$ is closed under addition of its elements, since C is a linear code and φ is a linear map. If $\mathbf{a} \in \mathbb{F}_q^n$ and $\mathbf{c} \in C$, then

$$a(x)c(x) = \varphi(a_0\mathbf{c} + a_1\sigma(\mathbf{c}) + a_{n-1}\sigma^{n-1}(\mathbf{c})) \in \mathcal{I}.$$

Hence $\mathcal{I}$ is an ideal in $\mathbb{C}_{q,n}$.

□

3.2. Generator polynomial

From the previous section, we know that cyclic codes have a one-to-one relation with an ideal in the ring $\mathbb{C}_{q,n}$. So, one way to describe cyclic codes is by its generator polynomial that generates the corresponding ideal. A cyclic code C considered as an ideal in $\mathbb{C}_{q,n}$ is generated by one element, but this element is not unique. An ideal that consists of all multiples of a fixed polynomial $g(x)$ by elements of $\mathcal{R}$ is called a *principal ideal*.

Note that, material on this section was taken from [11], [13], and [21].

Definition 3.8. Let C be a cyclic code of length n. Let $g(x)$ be the monic polynomial of minimal degree such that $g(x)$ generates C. Then $g(x)$ is called the *generator polynomial* of C.

Let C be a nonzero ideal in $\mathbb{C}_{q,n}$, i.e. a cyclic code of length n.

Proposition 3.9. *Let $g(x)$ be a polynomial in $\mathbb{F}_q[x]$. Then $g(x)$ is a generator polynomial of a cyclic code in $\mathbb{F}_q$ of length n if and only if $g(x)$ is monic and divides $x^n - 1$.*

Proof. We will proof this proposition in two directions.

(⇒) We need to show that there is a unique monic polynomial $g(x)$ of minimal degree in C. Suppose $f(x), g(x) \in C$ both are monic and have the minimal degree. But then $f(x)-g(x) \in C$ has lower degree. This is a contradiction unless $f(x) = g(x)$.

Now, we need to show that $g(x)$ is factor of $x^n - 1$. Write $x^n - 1 = h(x)g(x) + r(x)$ in $\mathbb{F}[x]$, where $\deg(r(x)) < \deg(g(x))$. In $\mathbb{C}_{q,n}$, $r(x) = -h(x)g(x) \in C$, this contradicts unless $r(x) = 0$.

(⇐) We need to show that $C = \langle g(x)\rangle$. Suppose $c(x) \in C$. Write $c(x) = q(x)g(x) + r(x)$ in $\mathbb{C}_{q,n}$, where $\deg(r(x)) < \deg(g(x))$. But $r(x) = c(x) - q(x)g(x) \in C$ since the code is linear, so $r(x) = 0$. Therefore, $c(x) \in \langle g(x)\rangle$.

□

Theorem 3.10. *Any $c(x) \in C$ can be written uniquely as $c(x) = f(x)g(x)$ in $\mathbb{F}[x]$, where $f(x) \in \mathbb{F}[x]$ has degree $< n - r$, $r = \deg(g(x))$. The dimension of C is $n - r$. Thus the message $f(x)$ becomes the codeword $f(x)g(x)$.*

Proof. Let $c(x)$ be a polynomial in $\langle g(x)\rangle$. Then $c(x) = q(x)g(x)$ in $\mathbb{C}_{q,n}$ for some polynomial $q(x)$, and hence $c(x) = q(x)g(x) + d(x)(x^n - 1)$ in $\mathbb{F}[x]$ for some polynomial $d(x)$. Since $g(x)|(x^n - 1)$,

$$c(x) = \left(q(x) + \frac{(x^n - 1)d(x)}{g(x)}\right) g(x), \text{ in } \mathbb{F}[x].$$

Hence every element of $\langle g(x)\rangle$ is of form

$$f(x)g(x) \text{ with } f(x) \in \mathbb{F}[x] \text{ and } \deg(f(x)) < n - r.$$

Moreover, the code consists of multiples of $g(x)$ by polynomials of degree $< n - r$, evaluated in $\mathbb{F}[x]$ not in $\mathbb{C}_{q,n}$. There are $n - r$ linearly independent multiples of $g(x)$, namely $g(x)$, $xg(x)$, ..., $x^{n-r-1}g(x)$. The corresponding vectors are the rows of the generator matrix G of C. Thus the code has dimension $n - r$. □

3.3. Factors of $x^n - 1$

The generator polynomial of a cyclic code of length n over $\mathbb{F}_q$ must be a factor of $x^n - 1$. For the existence of an integer m such that $q^m \equiv 1 \bmod n$, it is necessary and sufficient to assume that n and q are relatively prime.

Theorem 3.11 (Fermat-Euler Theorem). *If a and m are relatively prime, then $a^{\varphi(m)} \equiv 1 \bmod m$, where $\varphi(m)$ is the number of positive integers $\leq m$ that are relatively prime to m, for any integer m. Later on, $\varphi(m)$ is called the Euler-totient function*

By Theorem 3.11, there is a smallest integer m such that n divides $q^m - 1$. This integer m is called the *multiplicative order of q modulo n.*

Then $x^n - 1$ divides $x^{q^m} - 1$. Thus, the zeros of $x^n - 1$, which are called n-th *roots of unity*, are in the extension field $\mathbb{F}_{q^m}$. Let ω be a primitive element in $\mathbb{F}_{q^m}$. If n divides $q^m - 1$ or equivalently $q^m = 1 \bmod n$, for some positive integer m, then 1, α, α^2, ..., α^{n-1} are n mutually distinct zeros of $x^n - 1$, where $\alpha = \omega^{(q^m-1)/n}$. Thus we obtain a complete factorization of $x^n - 1$ into linear factors over $\mathbb{F}_{q^m}$,

$$x^n - 1 = \prod_{i=0}^{n-1} x - \alpha^i.$$

Since α is a zero of $x^n - 1$ and also generates the other zeros of $x^n - 1$, it is called a *primitive n-th root of unity*. $\mathbb{F}_{q^m}$ is called the *splitting field* of $x^n - 1$.

Definition 3.12. The operation of multiplying by q partitions the integers mod n into sets called the **cyclotomic cosets** mod n.

The *cyclotomic coset* containing s consist of

$$C_s = \{s, qs, q^2 s, \ldots, q^{m_s - 1} s\},$$

where m_s is the smallest positive integer such that $q^{m_s} s \equiv s \bmod n$ and s is the smallest number in the coset. The subscripts s are called the *cyclotomic coset representatives* modulo n.

In other words, the integers modulo n are partitioned into cyclotomic cosets,

$$\{0, 1, 2, \ldots, n-1\} = \bigcup_s C_s,$$

where s runs through a set of cyclotomic coset representatives modulo n. Then the *minimal polynomial* of α^s is

$$m_s(x) = \prod_{i \in C_s} (x - \alpha^i).$$

This is a monic polynomial with coefficients from $\mathbb{F}_q$, and is the lowest degree polynomial having α^s as a root. Therefore the complete factorization of $x^n - 1$ into irreducible polynomials over $\mathbb{F}_q$ is as follows,

$$x^n - 1 = \prod_s m_s(x),$$

where s runs through a set of cyclotomic coset representatives modulo n. For additional information about minimal polynomial, see [11], [13], and [21].

3.4. The zeros of a cyclic code

Let $g(x)$ be the generator polynomial of a cyclic code in $\mathbb{F}_q$ of length n. By Proposition 3.9, $g(x)$ divides $x^n - 1$, so its zeros are n-th roots of unity if n is not divisible by the characteristic of $\mathbb{F}_q$. Instead of describing a cyclic code by its generator polynomial $g(x)$, we can describe it by the set of zeros of $g(x)$ in the smallest extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ that contains n-th roots of unity where m is a positive integer, such that n divides $q^m - 1$.

From now we choose a fixed $\alpha \in \mathbb{F}_{q^m}^*$ of order n.

Definition 3.13. A subset I of $\mathbb{Z}_n$ is called a *defining set* of a cyclic code C if

$$C = \{ c(x) \in \mathbb{C}_{q,n} \,|\, c(\alpha^i) = 0 \text{ for all } i \in I\}.$$

The set of zeros of C is called the *complete defining set* and is defined as follows,

$$Z(C) = \{i \in \mathbb{Z}_n \,|\, c(\alpha^i) = 0 \text{ for all } c(x) \in C\}.$$

Let $f(x)$ be a q-ary polynomial dividing $x^n - 1$ and let α^i be a zero of $f(x)$. Then α^{iq} is a zero of $f(x)$ and by induction $\alpha^{iq^2}, \ldots, \alpha^{iq^{m-1}}$ are also zeros of $f(x)$. These exponents can be reduced modulo n, since $\alpha^n = 1$. The elements α^{iq^j} are called *cyclotomic conjugates* of α^i.

A generator polynomial of a cyclic code is the product of some minimal polynomials and the corresponding defining set of a cyclic code is the union of the corresponding cyclotomic cosets.

Proposition 3.14. *The relation between the generator polynomial $g(x)$ of a cyclic code C and the set of zeros $Z(C)$ is given by*

$$g(x) = \prod_{i \in Z(C)} (x - \alpha^i).$$

The dimension of C is equal to $n - |Z(C)|$.

Remark 3.15. Consider $\mathbb{C}_{n,q}$ to be the group algebra of a cyclic group G of order n. The mappings $\sigma_a : i \mapsto a \cdot i$, where a is an integer prime to n, form a group $\mathcal{G}$ of *automorphism* of G. An *automorphism* of a group G is a mapping σ onto itself which preserves multiplication, $\sigma(ab) = \sigma(a)\sigma(b)$. Thus $\mathcal{G}$ permutes the coordinate places $\mathbb{C}_{n,q}$, and sends cyclic codes into cyclic codes. $\mathcal{G}$ is a multiplicative abelian group, isomorphic to the multiplicative group of integers less than and prime to n, and has order $\varphi(n)$, where φ is the Euler φ-function. And the mapping $i \mapsto a \cdot i$, where a is prime to n, permutes the cyclotomic cosets.

Let $G = \mathbb{Z}_n$ and $\mathcal{G} = \mathbb{Z}_n^*$, where $\mathbb{Z}_n^*$ is a set of invertible elements in $\mathbb{Z}_n$.

For instance, if I_1 and I_2 are defining sets for the cyclic code C_1 and C_2, respectively, and

$$I_2 = \{a \cdot i \,|\, i \in I_1\},$$

for some a with $\gcd(a, n) = 1$, then C_1 and C_2 are *equivalent codes.*

Example 3.16. Let C be the binary cyclic code of length 31. The cyclotomic coset representatives modulo 31 are $\{0, 1, 3, 5, 7, 11, 15\}$. Let $C_{1,5,7}$ be a cyclic code of length 31 with defining set $\{1, 5, 7\}$. Hence the complete defining set of $C_{1,5,7}$ is given by

$$Z(C_{1,5,7}) = \{1, 2, 4, 5, 7, 8, 9, 10, 14, 16, 18, 19, 20, 25, 28\}.$$

Also let $C_{3,11,15}$ be a cyclic code of length 31 with defining set $\{3, 11, 15\}$. Hence the complete defining set of $C_{3,11,15}$ is given by

$$Z(C_{3,11,15}) = \{3, 6, 11, 12, 13, 15, 17, 21, 22, 23, 24, 26, 27, 29, 30\}.$$

As we can see, $Z(C_{3,11,15}) = 3 \cdot Z(C_{1,5,7})$, and $\gcd(3, 31) = 1$. By Remark 3.15, $C_{1,5,7}$ and $C_{3,11,15}$ are equivalent codes.

The complete factorization of $1 + x^{31}$ in $\mathbb{F}_2[x]$ is given by,

$$\begin{aligned} x^{31} - 1 &= (1+x)(1+x^2+x^5)(1+x^3+x^5)(1+x+x^2+x^3+x^5)(1+x+x^2+x^4+x^5) \\ &\quad (1+x+x^3+x^4+x^5)(1+x^2+x^3+x^4+x^5). \end{aligned}$$

If α be a zero of $1 + x^2 + x^5$, then α is an element of $\mathbb{F}_{2^5}$ of order 31. Hence,

$$\begin{aligned} m_1(x) &= 1 + x^2 + x^5 \\ m_3(x) &= 1 + x^2 + x^3 + x^4 + x^5 \\ m_5(x) &= 1 + x + x^2 + x^4 + x^5 \\ m_7(x) &= 1 + x + x^2 + x^3 + x^5 \\ m_{11}(x) &= 1 + x + x^3 + x^4 + x^5 \\ m_{15}(x) &= 1 + x^3 + x^5 \end{aligned}$$

Let $C_{1,5,7}$ be a binary cyclic code of length 31 with defining set $\{1, 5, 7\}$. Hence the generator polynomial of $C_{1,5,7}$ is given by,

$$\begin{aligned} g_0(x) &= m_1(x) \cdot m_5(x) \cdot m_7(x) \\ &= 1 + x^3 + x^8 + x^9 + x^{13} + x^{14} + x^{15}. \end{aligned}$$

And also let $C_{3,11,15}$ be a binary cyclic code of length 31 with defining set $\{3, 11, 15\}$. Hence the generator polynomial of $C_{3,11,15}$ is given by,

$$\begin{aligned} g_1(x) &= m_3(x) \cdot m_{11}(x) \cdot m_{15}(x) \\ &= 1 + x + x^2 + x^6 + x^7 + x^{12} + x^{15}. \end{aligned}$$

By Remark 3.15, binary cyclic codes of length 31 with generator $g_0(x)$ is equivalent with a binary cyclic code of length 31 with generator $g_1(x)$. Note that, the order of α^i is 31 for $i \neq 0$. Hence $C_{1,5,7}$ and $C_{3,11,15}$ are equivalent.

3.5. Mattson-Solomon polynomial

There are several ways of representing cyclic codes other than the standard way which was already discussed in Section 3.1. We shall now introduce a discrete analog of the Fourier transform, which in coding theory is also referred to as the *Mattson-Solomon polynomial.*

Definition 3.17. Let $\alpha \in \mathbb{F}_{q^m}^*$ be primitive n-th root of unity. The Mattson-Solomon (MS) polynomial $A(Z)$ of

$$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

is defined by

$$A(Z) = \sum_{i=1}^{n} A_i Z^{n-i}, \text{ where } A_i = a(\alpha^i) \in \mathbb{F}_{q^m}$$

The Mattson-Solomon polynomial $A(Z)$ is the *discrete Fourier transform* of $a(x)$. Therefore, we need to compute inverse discrete Fourier transform in order to get the coefficient of $a(x)$ in terms of the $A(Z)$.

Lemma 3.18. *Let $\beta \in \mathbb{F}_{q^m}$ be a zero of $x^n - 1$. Then*

$$\sum_{i=1}^{n} \beta^i = \begin{cases} n, & \text{if } \beta = 1; \\ 0, & \text{if } \beta \neq 1. \end{cases}$$

Proof. It is easy to prove this lemma, by considering two cases.

Case 1 If $\beta = 1$, then $\sum_{i=1}^{n} \beta^i = n$.

Case 2 If $\beta \neq 1$, then apply the sum of a geometric series to $\sum_{i=1}^{n} \beta^i$. This yields

$$\sum_{i=1}^{n} \beta^i = \frac{\beta^n - 1}{\beta - 1}.$$

Hence, $\sum_{i=1}^{n} \beta^i = 0$, since β is a zero of $x^n - 1$.

□

Proposition 3.19. *The inverse transform is given by*

$$a_i = \frac{1}{n} A(\alpha^i).$$

Proof. By definition of $A(\alpha^i)$,

$$A(\alpha^i) = a(\alpha)(\alpha^i)^{n-1} + a(\alpha^2)(\alpha^i)^{n-2} + \ldots + a(\alpha^{n-1})(\alpha^i) + a(1),$$

where

$$\begin{aligned} a(\alpha)(\alpha^i)^{n-1} &= a(\alpha)\alpha^{-i} \\ a(\alpha^2)(\alpha^i)^{n-2} &= a(\alpha^2)(\alpha^2)^{-i} \\ &\vdots \\ a(\alpha^{n-1})(\alpha^i) &= a(\alpha^{n-1})(\alpha^{n-1})^{-i} \\ a(1) &= a_0 + a_1 + \ldots + a_{n-1}. \end{aligned}$$

Thus,

$$\begin{aligned} a(\alpha)(\alpha^i)^{n-1} &= a_i + a_{i+1}\alpha + \ldots + a_{i-1}\alpha^{n-1} \\ a(\alpha^2)(\alpha^i)^{n-2} &= a_i + a_{i+1}\alpha^2 + \ldots + a_{i-1}\alpha^{2(n-1)} \\ &\vdots \\ a(\alpha^{n-1})(\alpha^i) &= a_i + a_{i+1}\alpha^{n-1} + \ldots + a_{i-1}\alpha^{(n-1)(n-1)} \\ a(1) &= a_i + a_{i+1} + \ldots + a_{i-1}. \end{aligned}$$

Apply Lemma 3.18, and we can conclude that

$$\begin{aligned} A(\alpha^i) &= a_i n + a_{i+1}\textstyle\sum_{j=1}^{n-1}(\alpha)^j + \ldots + a_{i-1}\sum_{j=1}^{n-1}(\alpha^{n-1})^j \\ &= a_i n. \end{aligned}$$

So,

$$\begin{aligned} A(\alpha^i) &= a_i n \\ a_i &= \tfrac{1}{n}A(\alpha^i) \end{aligned}$$

And this proves the assertion. □

Proposition 3.20. *$A(Z)$ is the Mattson-Solomon polynomial of a codeword $c(x)$ of the cyclic code C if and only if $A_j = 0$ for all $j \in Z(C)$ and $A_{jq} = A_j^q$ for all $j = 1, \ldots, n$.*

Proof. We will proof this proposition in two directions.

(⇒) Let $A(Z)$ be the MS polynomial for codeword $c(x) \in C$, where C is a cyclic code. Hence by definition,

$$A(Z) = \sum_{i=1}^{n} A_i Z^{n-i}, \text{ where } A_i = c(\alpha^i) \in \mathbb{F}_{q^m}.$$

For all $j = 1, \ldots, n$,

$$\begin{aligned} A_j &= c(\alpha^j) \\ &= c_0 + c_1\alpha^j + c_2(\alpha^j)^2 + \ldots + c_{n-1}(\alpha^j)^{n-1} \end{aligned}$$

and

$$\begin{aligned} (A_j)^q &= c_0^q + (c_1\alpha^i)^q + \ldots + (c_{n-1}(\alpha^j)^{n-1})^q \\ &= c_0 + c_1\alpha^{jq} + \ldots + c_{n-1}(\alpha^{jq})^{n-1} \\ &= A_{jq}, \end{aligned}$$

since $c_i^q = c_i$ for $i = 1, \ldots, n$. Thus for all $j \in Z(C)$, we have $A_j = c(\alpha^j) = 0$.

(⇐) Let $A(Z)$ be a polynomial over $\mathbb{F}_{q^m}$ with $A_{jq} = A_j^q$ for all $j = 0, 1, \ldots, n-1$ and $A_j = 0$ for all $j \in Z(C)$. Let

$$A(Z) = \sum_{i=1}^{n} A_i Z^{n-i}.$$

Let $c(x) \in \mathbb{F}_{q^m}[x]/(x^n - 1)$ with

$$c_j = \frac{1}{n}A(\alpha^j).$$

Then

$$c_j^q = \left(\frac{1}{n}A(\alpha^j)\right)^q = \left(\frac{1}{n}\right)^q (A(\alpha^j))^q = \frac{1}{n}\sum_{i=1}^{n} A_i^q \alpha^{jq(n-i)}$$

Multiplication by q modulo n is a permutation of $\mathbb{Z}_n$, since $\gcd(n, q) = 1$. Hence,

$$c_j^q = \frac{1}{n} \sum_{i=1} A_i \alpha^{j(n-i)} = \frac{1}{n} A(\alpha^j) = c_j.$$

Hence, $c_j \in \mathbb{F}_q$. Therefore, $c(x) \in \mathbb{F}_q[x]/(x^n - 1)$. Furthermore, $c(\alpha^i) = A_i = 0$, for all $i \in Z(C)$, where $Z(C)$ is the complete defining set of C. Hence, $c(x) \in C$.

□

Now we use the MS polynomial in terms of cyclic codes.

Lemma 3.21. *Let C be a cyclic code over $\mathbb{F}_q$ generated by*

$$g(x) = \prod_{k \in K} (x - \alpha^k),$$

where $\alpha \in \mathbb{F}_{q^m}$ is a primitive n-th root of unity. Suppose $\{1, 2, \ldots, d-1\} \subset K$ and $\mathbf{c} \in C$. Then the degree of the Mattson-Solomon polynomial A of a word $\mathbf{c}$ is at most $n - d$.

Proof. $c(\alpha^j) = 0$ for $1 \leq j \leq d - 1$ since $c(x)$ is divisible by $g(x)$. The result follows from Definition 3.17. □

Suppose the vector $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$, $a_i \in \mathbf{F}_q$, has non-zero components

$$a_{i_1}, a_{i_2}, \ldots, a_{i_w}$$

and no others, where $w = \mathrm{wt}(\mathbf{a})$. We associate with $\mathbf{a}$ the following elements of $\mathbb{F}_{q^m}$

$$X_1 = \alpha^{i_1}, X_2 = \alpha^{i_2}, \ldots, X_w = \alpha^{i_w},$$

called the *locators* of $\mathbf{a}$, and the following elements of $\mathbb{F}_q$,

$$Y_1 = a_{i_1}, Y_2 = a_{i_2}, \ldots, Y_w = a_{i_w},$$

giving the values of the non-zero components. Thus $\mathbf{a}$ is completely specified by the list (X_1, Y_1), (X_2, Y_2), ..., (X_w, Y_w). If $\mathbf{a}$ is a binary vector, then the Y_i's are 1. Note that,

$$a(\alpha^j) = A_j = \sum_{i=1}^{w} Y_i X_i^j.$$

Definition 3.22. The *locator polynomial* of the vector $\mathbf{a}$ is

$$\begin{aligned} \sigma(z) &= \prod_{i=1}^{w} (1 - X_i z) \\ &= \sum_{i=0}^{w} \sigma_i z_i, \end{aligned}$$

The roots of $\sigma(z)$ are the reciprocals of the locators. Thus the coefficients σ_i are the *elementary symmetric function* of the X_i:

$$
\begin{aligned}
\sigma_0 &= 1\\
\sigma_1 &= -(X_1 + X_2 + \ldots + X_w)\\
\sigma_2 &= X_1X_2 + X_1X_3 + \ldots + X_{w-1}X_w\\
&\;\;\vdots\\
\sigma_w &= (-1)^w X_1X_2\cdots X_w.
\end{aligned}
$$

Theorem 3.23. *If there are r n-th roots of unity which are zeros of the Mattson-Solomon polynomial A of a word $\mathbf{c}$, then* $\mathrm{wt}(\mathbf{c}) = n - r$.

Proof. This is immediate consequences of Proposition 3.19 □

Corollary 3.24. *If $\mathbf{c}$ has Mattson-Solomon polynomial $A(Z)$, then* $\mathrm{wt}(\mathbf{c}) \geq n - \deg(A(Z))$.

We will use these results to prove the BCH bound and its generalization the Hartmann-Tzeng bound.

3.6. Parity check and the minimum distance

In the previous chapter, we described a cyclic code by its generator polynomial. Cyclic codes are linear codes. Therefore, they are given by a set of homogeneous linear equations, i.e. by the null space of a matrix.

Let C be a cyclic code of length n. Let $g(x)$ be the generator polynomial of C and from Proposition 3.9, $g(x)$ divides $x^n - 1$. Then

$$
\begin{aligned}
h(x) &= \frac{x^n - 1}{g(x)}\\
&= \sum_{i=0}^{k} h_i x^i, \text{ where } h_k \neq 0,
\end{aligned}
$$

is called the *check polynomial* of C. If

$$c(x) = \sum_{i=0}^{n-1} c_i x^i = f(x)g(x)$$

is any codeword of C, then

$$
\begin{aligned}
c(x)h(x) &= \sum_{i=0}^{n-1} c_i x^i \cdot \sum_{j=0}^{k} h_j x^j\\
&= f(x)g(x)h(x)\\
&= 0 \text{ in } \mathbb{C}_{q,n}.
\end{aligned}
$$

The coefficient of x^j in this product is

$$\sum_{i=0}^{n-1} c_i h_{j-i}, \text{ for } j = 0, 1, 2, \ldots, n-1, \tag{3.1}$$

where the subscripts are taken modulo n. Thus the equations 3.1 are parity check equations satisfied by the code.

Let C be a linear code with parameter $[n, k, d]$. Suppose the matrix H is an $m \times n$ matrix with entries in $\mathbb{F}_q$. If C be the null space of H, then C is the set of all $\mathbf{c} \in \mathbb{F}_q^n$ such that $H\mathbf{c}^t = \mathbf{0}^t$. Hence, we get $n - m$ homogeneous linear equations. They are called *parity check equations*. The dimension of C is at least $n - m$. If there are dependent rows in the matrix H, where $k < n - m$, then delete few row until we get an $(n - k) \times n$ matrix H' with independent rows and with the same null space as H. So $\operatorname{rank}(H')$ is equal to $n - k$.

Definition 3.25. A parity check matrix H of an $[n,k,d]$ code C is an $(n - k) \times n$ matrix, satisfying

$$\mathbf{c} \in C \Leftrightarrow H\mathbf{c}^t = \mathbf{0}^t.$$

In other words, C is the null space of matrix H of rank $n - k$.

The parity check matrix of a linear code can be used to detect errors during the transmission. Suppose that the minimum distance of C is equal to d and H is the parity check matrix of code C. Suppose that the codeword $\mathbf{c}$ is transmitted and $\mathbf{r} = \mathbf{c} + \mathbf{e}$ is the received codeword. Then $\mathbf{e}$ is called the *error vector* and $\operatorname{wt}(\mathbf{e})$ is called the number of errors that occurs during the transmission.

Theorem 3.26. *Consider $\mathbb{F}_q^n$ with $\gcd(q, n) = 1$. Let m satisfy $q^m \equiv 1 \bmod n$ and let ω be a primitive element in $\mathbb{F}_{q^m}$. Then $\alpha = \omega^{(q^m-1)/n}$ is a primitive n-th root of unity.*

Let $I = \{i_1, i_2, \ldots, i_l\}$ be a subset of $\{0, 1, \ldots, n - 1\}$. Let I be a defining set of the q-ary, cyclic code $C(I)$ of length n. Then $C(I)$ can be described in the following way:

$$C(I) = \{\mathbf{c} \in \mathbb{F}_q^n \mid H\mathbf{c}^t = \mathbf{0}^t\},$$

where

$$H = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \ldots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \ldots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & & & \vdots \\ 1 & \alpha^{i_l} & \alpha^{2i_l} & \ldots & \alpha^{(n-1)i_l} \end{pmatrix}$$

Definition 3.27. We denote by $C^*(I)$, the code over $\mathbb{F}_{q^m}$ with H as parity check matrix.

An important parameter of a code C, besides its length and dimension, is the minimum distance between its codewords. As already discussed in the previous chapter, the minimum distance determines how many errors a code C can correct, see Section 2.1. In this chapter, we will discuss the parity check matrix of a cyclic code, and its relation with the minimum distance of the code. For additional reading, see [13] and [21].

Theorem 3.28. *A linear code C has minimum distance d if and only if d is the maximum number such that any $d - 1$ columns of its parity check matrix are linearly independent.*

Proof. Let C be a linear code and $\mathbf{u}$ be a codeword such that $wt(\mathbf{u}) = d(\mathcal{C}) = d$. Since $\mathbf{u} \in \mathcal{C}$ if and only if $H\mathbf{u}^t = 0$ and $\mathbf{u}$ has d non-zeros components, some d columns of H are linearly independent. Any $d - 1$ columns of H must be linearly independent, or else there would exist a non-zero codeword in $\mathcal{C}$ with weight $d - 1$. □

Let us explain Theorem 3.28 in more detail way. Let $\mathbf{h}_1, \ldots, \mathbf{h}_n$ be the columns of H. Let $\mathbf{c}$ be a nonzero codeword of weight w.

Definition 3.29. Let the *support* of codeword $\mathbf{c}$ be denoted by $\mathrm{supp}(\mathbf{c})$ is defined as follows

$$\mathrm{supp}(\mathbf{c}) = \{j_1, j_2 \ldots, j_w\},$$

where $1 \leq j_1 < j_2 < \ldots < j_w \leq n$, such that c_{j_1}, c_{j_2}, …, c_{j_w} are not equal to 0.

Since H is the parity check matrix of code C, then $H\mathbf{c}^t = 0$. We can re-write $H\mathbf{c}^t = 0$ in the following term;

$$c_{j_1}\mathbf{h}_{j_1} + \ldots + c_{j_w}\mathbf{h}_{j_w} = 0,$$

for all $i = 1, \ldots, w$. Thus, the columns $\mathbf{h}_{j_1}, \ldots, \mathbf{h}_{j_w}$ are dependent. Conversely, if $\mathbf{h}_{j_1}, \ldots, \mathbf{h}_{j_w}$ are dependent, then there exist constant $a_1, \ldots, a_w$, not all zero, such that

$$a_1\mathbf{h}_{j_1} + \ldots + a_w\mathbf{h}_{j_w} = 0.$$

Let $\mathbf{c}$ be the word defined by $c_j = 0$ if $j \neq j_i$ for all i, and $c_j = a_i$ if $j = j_i$ for some i. Then $H\mathbf{c}^t = \mathbf{0}^t$. Hence $\mathbf{c}$ is a nonzero codeword of weight at most w.

Let H be the parity check matrix of a code C. As consequences, the minimum distance of C is equal to 1 if and only if H has a zero column. If H has no zero column, then the minimum distance of a code C is at least 2. Theorem 3.28 is an important tool to find the minimum distance of linear codes. Since cyclic codes are linear, then we can use Theorem 3.28 to determine its minimum distance.

Let C be a cyclic code with generator polynomial $g(x)$ and check polynomial $h(x) = (x^n - 1)/g(x)$.

Theorem 3.30. *The dual code $C^\perp$ is cyclic and has generator polynomial*

$$g^\perp(x) = x^{\deg(h(x))}h(x^{-1}).$$

3.7. Idempotents

The subject of this section mainly taken from [11].

Definition 3.31. A polynomial $\theta(x)$ of $\mathbb{C}_{q,n}$ is an *idempotent* if

$$\theta(x) = \theta(x)^2.$$

Theorem 3.32. *A cyclic code $C = \langle g(x) \rangle$ contains a unique idempotent $\theta(x)$ such that $C = \langle \theta(x) \rangle$. Also $\theta(x) = p(x)g(x)$ for some polynomial $p(x)$, and*

$$\theta(\alpha^i) = 0 \Leftrightarrow g(\alpha^i) = 0.$$

Proof. Let $g(x)$ be the generator polynomial of C and $h(x)$ the parity check polynomial of C, where $g(x)$ and $h(x)$ are relatively prime. As a consequence of the *Euclidean* algorithm, there exist polynomial $p(x)$ and $q(x)$ such that

$$p(x)g(x) + q(x)h(x) = 1, \text{ in } \mathbb{F}_q[x].$$

Let $\theta(x) = p(x)g(x)$. Then,

$$p(x)g(x)\,(p(x)g(x) + q(x)h(x)) = p(x)g(x),$$

i.e.

$$\theta(x)^2 + 0 = \theta(x), \text{ in } \mathbb{C}_{q,n},$$

so $\theta(x)$ is an idempotent. An n-th root of unity is a zero of either $g(x)$ or $h(x)$, but not both. Since $p(x)g(x) + q(x)h(x) = 1$ in $\mathbb{F}_q[x]$, hence $p(x)$ and $h(x)$ are relatively prime. So if there is an n-th root of unity which is a zero of $p(x)$, it must also be a zero of $g(x)$. Since $p(x)$ does not introduce any new zeros, hence $\theta(x)$ and $g(x)$ generate the same code. To show that $\theta(x)$ is the unique idempotent which generates C, suppose $\vartheta(x)$ is another idempotent that generates C, then from the following theorem, $\vartheta(x)\theta(x) = \theta(x) = \vartheta(x)$. $\square$

Theorem 3.33. $c(x) \in C \Leftrightarrow c(x)\theta(x) = c(x)$.

Proof. If $c(x) = c(x)\theta(x)$, then clearly $c(x) \in C$. Conversely, if $c(x) \in C$, then $c(x) = b(x)\theta(x)$, and $c(x)\theta(x) = b(x)\theta(x)^2 = b(x)\theta(x) = c(x)$. $\square$

Lemma 3.34. *$\theta(x)$ is an idempotent if and only if $\theta(\alpha^i) = 0$ or 1 for $i = 0, 1, 2, \ldots, n-1$.*

To proof Lemma 3.34, we need the following theorem from the finite field theory;

Lemma 3.35. *If n, r, s are integers with $n \geq 2$, $r \geq 1$, $s \geq 1$, then*

$$n^s - 1 | n^r - 1 \Leftrightarrow s | r.$$

Proof. Write $r = as + b$, where $0 \leq b < s$. Then

$$\frac{n^r - 1}{n^s - 1} = n^b \cdot \frac{n^{as} - 1}{n^s - 1} + \frac{n^b - 1}{n^s - 1}.$$

Term $n^{as} - 1$ is always divisible by $n^s - 1$. Term $n^b - 1$ is less than 1 and so is an integer if and only if $b = 0$. $\square$

Theorem 3.36. *$\mathbb{F}_{q^r}$ contains a subfield isomorphic to $\mathbb{F}_{q^s}$ if and only if s divides r.*

Proof. If $s|r$, then $\mathbb{F}_{q^r}$ contains a subfield isomorphic to $\mathbb{F}_{q^s}$. Conversely, let β be a primitive element of $\mathbb{F}_{q^s}$. Then

$$\beta^{q^s - 1} = 1, \text{ and } \beta^{q^r - 1} = 1.$$

So, $q^s - 1$ divides $q^r - 1$, and s divides r by Lemma 3.35. $\square$

Theorem 3.37 (Fermat Theorem). *Every element β of a field $\mathbb{F}$ of order q^m satisfies the identity*

$$\beta^{q^m} = \beta,$$

or equivalently is a root of the equation

$$x^{q^m} = x.$$

Thus,

$$x^{q^m} - x = \prod_{\beta \in \mathbb{F}} (x - \beta).$$

Theorem 3.38. *If $\beta \in \mathbb{F}_{q^r}$, then β is in $\mathbb{F}_{q^s}$ if and only if $\beta^{q^s} = \beta$. In any field if $\beta^2 = \beta$, then β is 0 or 1.*

Proof. The first statement is an immediate consequence of Theorem 3.37. The second statement is obvious. □

Proof of Lemma 3.34 Let $\theta(x)$ be an idempotent. Then by Theorem 3.38, $\theta(\alpha^i)^2 = \theta(\alpha^i)$. So $\theta(\alpha^i) = 0$ or 1 for $i = 0, 1, 2, \ldots, n-1$. Conversely, let

$$\theta(x) = \sum_{i=0}^{n-1} \epsilon_i x^i.$$

Since $\theta(\alpha^i)$ is 0 or 1, hence $\theta(\alpha^{2j}) = \theta(\alpha^j)^2 = \theta(\alpha^j)$. By the *inversion* formula,

$$\epsilon_i = \sum_{j=0}^{n-1} \theta(\alpha^j)\alpha^{-ij} = \sum_{s} \sum_{j \in C_s} \alpha^{-ij},$$

where s runs through a subset of the cyclotomic cosets. Thus, $\epsilon_i = \epsilon_{2i}$, and $\theta(x)$ is an idempotent. □

4

Lower Bounds for the Minimum Distance of Cyclic Codes

In this chapter, we will discuss lower bounds on the minimum distance of cyclic codes, due to the BCH bound, Hartmann-Tzeng bound, Roos bound, and the AB-method. We borrowed and adapted notations and definitions from [11], [13], [19] and [21].

4.1. The BCH bound

A very general class of cyclic codes with a guaranteed minimum distance is given by BCH codes. They are named after R.C.Bose, D.K.Chaudhuri and A. Hocquenghem, the inventors of these codes.

Definition 4.1. A cyclic code of length n over $\mathbb{F}_q$ with *generator polynomial* $g(x)$ is a BCH code of *designed minimum distance* δ, if, for some integer $b \geq 0$,

$$g(x) = \text{l.c.m}\{m_b(x), m_{b+1}(x), \ldots, m_{b+\delta-2}(x)\}.$$

In other words, $g(x)$ is the lowest degree monic polynomial over $\mathbb{F}_q$ having $\{b, b+1, \ldots, b+\delta-2\}$ as its defining set. Therefore,

$$\mathbf{c} \in C \text{ if and only if } c(\alpha^b) = c(\alpha^{b+1}) = \ldots = c(\alpha^{b+\delta-2}) = 0.$$

Which means this code has $\delta - 1$ consecutive elements in its defining set.

Theorem 4.2. *Let C be a cyclic code with* designed minimum distance δ*. Then the minimum distance of the code is at least δ.*

We will give three different proofs of Theorem 4.2. The first proof was taken from [11] and [13].

First Proof. Let $Z(C)$ contain the consecutive elements $\{b \leq i \leq b+\delta-2\}$ for certain b. Then the parity check matrix H of a $\mathbb{F}_{q^m}$-linear code C^* that has C as its subfield sub-code is

$$H = (\alpha^{ij} | b \leq i \leq b+\delta-2, 0 \leq j \leq n-1).$$

Let

$$H' = \begin{pmatrix} \alpha_{i_1}^b & \dots & \alpha_{i_t}^b \\ \vdots & & \\ \alpha_{i_1}^{b+\delta-2} & \dots & \alpha_{i_t}^{b+\delta-2} \end{pmatrix}$$

be a square sub-matrix of size $t = \delta - 1$ of H. Then H' is a *Vandermonde* matrix. Therefore

$$\det(H') = \alpha_{i_1}^b \dots \alpha_{i_t}^b \prod_{1 \le r < s \le t} (\alpha_{i_s} - \alpha_{i_r}) \neq 0.$$

Since the α_i are nonzero and mutually distinct. So any $\delta - 1$ columns of H are independent. Hence by Theorem 3.28, the minimum distance of C is at least δ. □

The second proof is the application of the Mattson-Solomon polynomial in cyclic codes. As immediate result of Lemma 3.21.

Second Proof. Let $c(x)$ be any nonzero codeword in C. Let $c(x)$ be a nonzero codeword in C. By hypothesis, $c(\alpha^j) = 0$, for $b \le j \le b + \delta - 2$. Let $A(Z)$ be the MS-polynomial of $c(x)$. Then

$$A(Z) = c(\alpha)Z^{n-1} + \dots + c(\alpha^{b-1})Z^{n-b+1} + c(\alpha^{b+\delta-1})Z^{n-b-\delta+1} + \dots + c(\alpha^n).$$

Let

$$\begin{aligned} \hat{A}(Z) &= Z^{b-1}A(Z) - (c(\alpha)Z^{b-2} + \dots + c(\alpha^{b-1}))(Z^n - 1) \\ &= c(\alpha^{b+\delta-1})Z^{n-\delta} + \dots + c(\alpha^n)Z^{b-1} + c(\alpha)Z^{b-2} + \dots + c(\alpha^{b-1}). \end{aligned}$$

Clearly, the number of n-th roots of unity which are zeros of $A(Z)$ is the same as the number which are zeros of $\hat{A}(Z)$. This number is at most $\deg(\hat{A}(Z)) \le n - \delta$. Thus the weight of $\mathbf{c}$ is at least δ by Theorem 3.23. □

The third proof was from [8] on BCH bound. Instead of the parity check matrix approach, they use the locator polynomial to prove the BCH bound. They used this approach to proof the generalization of the BCH bound.

Third Proof. Let $c(x)$ be a code polynomial of weight $w < \delta$. Since C is a cyclic code, we may assume without loss of generality that

$$c(x) = 1 + c_1x^{t_1} + c_2x^{t_2} + \dots + c_{w-1}x^{t_{w-1}},$$

where $c_i \neq 0$, $c_i \in \mathbb{F}_q$ and t_i are mutually distinct positive integers less than n. Let

$$X_i = \alpha^{t_i},$$

and

$$S_j = c_1X_1^j + c_2X_2^j + \dots + c_{w-1}X_{w-1}^j.$$

Then

$$\begin{aligned} S_j &= c(\alpha^j) - 1 \\ &= -1, \end{aligned}$$

for all j such that $g(\alpha^j) = 0$, in other words $S_j = -1$ if j is an element of the complete defining set of C. Now let

$$\begin{aligned}\sigma(x) &= \prod_{i=1}^{w-1}(x - X_i^a) \\ &= x^{w-1} + \sigma_1 x^{w-2} + \ldots + \sigma_{w-2}x + \sigma_{w-1}.\end{aligned}$$

Now $0 < t_i < n$ and $\gcd(n, a) = 1$. Hence $X_i \neq 1$ and $X_i^a \neq 1$. So, if we substitute $x = 1$, then $\sigma(1) \neq 0$.

In the equation

$$\prod_{i=1}^{w-1}(x - X_i^a) = x^{w-1} + \sigma_1 x^{w-2} + \ldots + \sigma_{w-2}x + \sigma_{w-1},$$

substitute $x = X_i^a$. And we get

$$X_i^{a(w-1)} + \sigma_1 X_i^{a(w-2)} + \ldots + \sigma_{w-2}X_i^a + \sigma_{w-1} = 0.$$

Multiply both side with $c_i X_i^b$, and we get

$$c_i X_i^b X_i^{a(w-1)} + c_i X_i^b \sigma_1 X_i^{a(w-2)} + \ldots + c_i X_i^b \sigma_{w-2} X_i^a + c_i X_i^b \sigma_{w-1} = 0.$$

Summing on $i = 1, \ldots, w-1$ gives,

$$\begin{aligned}S_{b+(w-1)a} + \sigma_1 S_{b+(w-2)a} + \ldots + \sigma_{w-2}S_{b+a} + \sigma_{w-1}S_b &= \sum_{i=1}^{w-1} c_i X_i^b \sigma(X_i^a) \\ &= 0.\end{aligned}$$

Since $S_{b+ia} = -1$ for all $i = 0, 1, \ldots, \delta - 2$ and $w < \delta$, the above equation implies that $\sigma(1) = 0$, which is a contradiction. Therefore, there does not exist any codeword of weight less than δ. Hence the minimum distance of C is greater than or equal to δ. $\square$

Definition 4.3. For a subset I of $\mathbb{Z}_n$. Let $d_{BCH}(I)$ be the largest number δ such that I contain a subset of the form $\{b + i \cdot a \mid 0 \leq i \leq \delta - 2\}$ with $\gcd(a, n) = 1$.

Proposition 4.4 (The BCH bound). *Let $Z(C)$ be the complete defining set of a cyclic code of length n. Then the minimum distance of C is at least $d_{BCH}(Z(C))$. Let C be a cyclic code of length n. Then $d_{BCH}(Z(C))$ is denoted by $d_{BCH}(C)$.*

Proof. Immediate result of Theorem 4.2 and Remark 3.15. $\square$

We deduce that the minimum distance of a cyclic code of length n over $\mathbb{F}_q$ with defining set $\{b, b+1, \ldots, b+\delta-2\}$ is greater than or equal to the designed distance δ.

Remark 4.5. If $b = 1$, then these codes are called *narrow-sense* BCH codes. If $n = q^m - 1$, (so if α is a primitive element of $\mathrm{GF}(q^m)$), then the BCH code is called *primitive*. BCH codes with $n = q - 1$, i.e. $m = 1$, and $\alpha \in \mathrm{GF}(q)$, are called *Reed-Solomon* codes.

4.2. The Hartmann-Tzeng bound

The BCH bound considers only one set of $\delta - 1$ consecutive elements in the complete defining set. But in reality, there exist many cyclic codes whose complete defining sets possess more than one set of $\delta - 1$ consecutive elements. It has been shown by C.R.P. Hartmann and K.K. Tzeng [8] that when considerations are given to these multiple sets of $\delta - 1$ consecutive elements, much improvement over the BCH bound can be obtained.

Hartmann-Tzeng (HT) presented the bound for the minimum distance for cyclic codes generated by polynomials with more than one set of consecutive elements in its complete defining set. We borrowed and adapted notations and definitions from [8].

Theorem 4.6. *Let $g(x) \in \mathbb{F}_q[x]$ be the generator polynomial of a cyclic code, C, of length n. If*

$$g(\alpha^{b+i_1a_1+i_2a_2}) = 0$$

for $i_1 = 0, 1, 2, \ldots, \delta - 2$ and $i_2 = 0, 1, \ldots, s$ where $\gcd(n, a_1) = 1$ and $\gcd(n, a_2) = 1$, then the minimum distance of C is at least $\delta + s$.

The proof of this theorem is basically an extended version of the proof of the BCH bound. Instead of considering one set of consecutive roots, this theorem considers multiple set of consecutive roots. And the proof is taken from [8].

Proof. By Theorem 4.2, the minimum distance of C is greater than or equal to δ. Let $c(x)$ be a codeword polynomial of weight w such that $\delta \leq w < \delta + s$. Since C is a cyclic code, $c(x)$ can be written as follows;

$$c(x) = 1 + \sum_{i=1}^{w-1} c_i x^{t_i},$$

where $c_i \neq 0$, $c_i \in \mathbb{F}_q$ and t_i are distinct positive integers less than n. Let

$$X_i = \alpha^{t_i},$$

and

$$S_j = \sum_{i=1}^{w-1} c_i X_i^j.$$

Then

$$\begin{aligned} S_j &= c(\alpha^j) - 1 \\ &= -1, \end{aligned}$$

for all j such that $g(\alpha^j) = 0$. Now let

$$\begin{aligned}
\sigma_1(x) &= \prod_{i_1=1}^{\delta-2}(x - X_{i_1}^{a_1}) \\
&= x^{(\delta-2)} + \sigma_1^{(1)}x^{\delta-3} + \ldots + \sigma_{\delta-3}^{(1)}x + \sigma_{\delta-2}^{(1)}, \\
\sigma_2(x) &= \prod_{i_2=\delta-1}^{w-1}(x - X_{i_2}^{a_2}) \\
&= x^{(w-\delta+1)} + \sigma_1^{(2)}x^{w-\delta} + \ldots + \sigma_{w-\delta}^{(2)}x + \sigma_{w-\delta+1}^{(2)}, \\
\sigma(x) &= \sigma_1(x)\sigma_2(x).
\end{aligned}$$

Since $t_i \neq 0$, $\gcd(n, a_1) = 1$, $\gcd(n, a_2) = 1$, then $X_i \neq 1$, $X_{i_1}^{a_1} \neq 1$, and $X_{i_2}^{a_2} \neq 1$. These yields $\sigma(1) \neq 0$.

Before we continue, we need to do some trick here. First, in equation $\sigma_1(x)$, substitute x with $X_i^{a_1}$ for $i = 1, 2, \ldots, \delta - 2$. Hence we get

$$\sigma_1(X_i^{a_1}) = X_i^{a_1(\delta-2)} + \sigma_1^{(1)}X_i^{a_1(\delta-3)} + \ldots + \sigma_{\delta-3}^{(1)}X_i^{a_1} + \sigma_{\delta-2}^{(1)} = 0.$$

Also in equation $\sigma_2(x)$, substitute x with $X_i^{a_2}$ for $i = \delta - 1, \ldots, w$. Then

$$\sigma_2(X_i^{a_2}) = X_i^{a_2(w-\delta+1)} + \sigma_1^{(2)}X_i^{a_2(w-\delta)} + \ldots + \sigma_{w-\delta}^{(2)}X_i^{a_2} + \sigma_{w-\delta+1}^{(2)} = 0.$$

Note that, $\sigma(X_i) = \sigma_1(X_i^{a_1})\sigma_2(X_i^{a_2}) = 0$ for $i = 1, \ldots, w$. Multiply both sides with $c_iX_i^b$ and then summing the result on $i = 1, 2, \ldots, w$. We get

$$\begin{aligned}
&\left(S_{b+(\delta-2)a_1+(w-\delta+1)a_2} + \sigma_1^{(1)}S_{b+(\delta-3)a_1+(w-\delta+1)a_2} + \ldots + \sigma_{\delta-2}^{(1)}S_{b+(w-\delta+1)a_2}\right) \\
&\quad +\sigma_1^{(2)}\left(S_{b+(\delta-2)a_1+(w-\delta)a_2} + \sigma_1^{(1)}S_{b+(\delta-3)a_1+(w-\delta)a_2} + \ldots + \sigma_{\delta-2}^{(1)}S_{b+(w-\delta)a_2}\right) \\
&\quad + \ldots + \sigma_{w-\delta-1}^{(2)}\left(S_{b+(\delta-2)a_1} + \sigma_1^{(1)}S_{b+(\delta-3)a_1} + \ldots + \sigma_{\delta-2}^{(1)}S_b\right) \\
&= \sum_{i=1}^{w-1} c_iX_i^b\sigma_1(X_i^{a_1})\sigma_2(X_i^{a_1}) \\
&= 0
\end{aligned}$$

Since $S_{b+i_1a_1+i_2a_2} = -1$ for $i_1 = 0, 1, \ldots, \delta - 2$ and $i_2 = 0, 1, \ldots, s$ and $\delta \leq w < \delta + s$, we have $\sigma(1) = 0$ which is contradiction. Therefore, there dos not exist any codeword of weight less than $\delta + s$. Hence $d \geq \delta + s$. □

Definition 4.7. For a subset I of $\mathbb{Z}_n$. Let $d_{HT}(I)$ be the largest number γ such that there exists a subset of I of the form $\{b+i_1\cdot a_1+i_2\cdot a_2 \mid 0 \leq i_1 \leq \delta-2, 0 \leq i_2 \leq s\}$ with $\gcd(a_1, n) = \gcd(a_2, n) = 1$ and $\gamma = \delta + s$. Let C be a cyclic code of length n. Then $d_{HT}(Z(C))$ is denoted by $d_{HT}(C)$.

Theorem 4.8 (The HT bound). *Let $Z(C)$ be the complete defining set of a cyclic code of length n. Then the minimum distance of C is at least $d_{HT}(Z(C))$.*

Proof. As immediate consequence of Definition 4.8 and Theorem 4.6. □

Proposition 4.9. *Let I be a subset of $\mathbb{Z}_n$. Then $d_{HT}(I) \geq d_{BCH}(I)$.*

Proof. Take $A = \{b + i \cdot a_1 \,|\, i = 0, 1, \ldots, \delta - 2\}$ and $B = \{0\}$, where $a_1 = 1$ in the HT bound, then we get the BCH bound. □

Remark 4.10. In the Hartmann-Tzeng (HT) bound, if $a_1 = 1$, there are $s + 1$ of $\delta - 1$ consecutive elements in the complete defining set, $Z(C)$, of a q-ary cyclic code C of length n.

C. Roos improved the HT bound as follows;

Proposition 4.11. *Let $g(x) \in \mathbb{F}_q[x]$ be the generator polynomial of a cyclic code, C, of length n. If*

$$g(\alpha^{b+i_1a_1+i_2a_2}) = 0$$

for $i_1 = 0, 1, 2, \ldots, \delta - 2$ and $i_2 = 0, 1, \ldots, s$ where $\gcd(n, a_1) = 1$ and $\gcd(n, a_2) < \delta$, then the minimum distance of C is at least $\delta + s$.

The proof of Proposition 4.11 can be found in [15]. He showed that the HT bound can be strengthened by choosing a_2 with $\gcd(n, a_2) < \delta$ instead of $\gcd(n, a_2) = 1$.

Definition 4.12. For a subset I of $\mathbb{Z}_n$. Let $d_{HTR}(I)$ be the largest number γ such that there exists a subset of I of the form $\{b + i_1 \cdot a_1 + i_2 \cdot a_2 \,|\, 0 \leq i_1 \leq \delta - 2, 0 \leq i_2 \leq s\}$ with $\gcd(a_1, n) = 1$, $\gcd(a_2, n) < \delta$ and $\gamma = \delta + s$. Let C be a cyclic code of length n. Then $d_{HTR}(Z(C))$ is denoted by $d_{HTR}(C)$.

Theorem 4.13 (The HT-Roos bound). *Let $Z(C)$ be the complete defining set of a cyclic code of length n. Then the minimum distance of C is at least $d_{HTR}(Z(C))$.*

Proof. As immediate consequence of Definition 4.12 and Proposition 4.11. □

The following example was taken from [15]. This example shows the improvement of HT bound.

Example 4.14. Let C be the cyclic code of length 51 with defining set $\{1, 5, 9\}$. The complete defining set

$$Z(C) = \{1, 2, 4, 5, 7, 8, 9, 10, 13, 14, 15, 16, 18, 20, 21, 26, 28, 29, 30, 32, 33, 36, 40, 42\}.$$

By the BCH bound with $b = 7$ and $a = 1$, the minimum distance of C is $d_{BCH} \geq 5$. By the HT bound with $b = 1$, $a_1 = 1$, $a_2 = 14$, $s = 2$, and $\delta = 3$, hence $d_{HT} = 5$. Based on the HT bound, the minimum distance of C is at least 5. By the HT-Roos bound with $b = 7$, $a_1 = 1$, $a_2 = 6$, $s = 1$, and $\delta = 5$, we get $d_{HTR} = 6$. Based on the HT-Roos bound, the minimum distance of C is at least 6.

4.3. The Roos bound

In this section, we will discuss a lower bound on the minimum distance of a cyclic code based on the paper by C.Roos [16].

Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ be any matrix over a finite field $\mathbb{F}$ with n columns $\mathbf{y}_i$. Let C be the linear code over the field $\mathbb{F}$ with $\mathbf{Y}$ as parity check matrix:

$$C = \{\mathbf{c} \in \mathbb{F}^n \,|\, \mathbf{Y}\mathbf{c}^t = \mathbf{0}\}.$$

The minimum distance of C will be denoted as $d_{\mathbf{Y}}$.

Let $\mathbf{X}$ be any $m \times n$ matrix, with nonzero columns $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{mi}) \in \mathbb{F}^m$ for $1 \leq i \leq n$, we define the matrix $\mathbf{X} * \mathbf{Y}$ as

$$\mathbf{X} * \mathbf{Y} = \begin{pmatrix} x_{11}\mathbf{y}_1 & x_{12}\mathbf{y}_2 & \ldots & x_{1n}\mathbf{y}_n \\ x_{21}\mathbf{y}_1 & x_{22}\mathbf{y}_2 & \ldots & x_{2n}\mathbf{y}_n \\ . & . & \ldots & . \\ x_{m1}\mathbf{y}_1 & x_{m2}\mathbf{y}_2 & \ldots & x_{mn}\mathbf{y}_n \end{pmatrix}.$$

Theorem 4.15. *If $d_{\mathbf{Y}} \geq 2$ and every $m \times (m + d_{\mathbf{Y}} - 2)$ sub matrix of $\mathbf{X}$ has full rank, then $d_{\mathbf{X}*\mathbf{Y}} \geq d_{\mathbf{Y}} + m - 1$.*

Proof. The proof is by contradiction. Suppose that $d_{\mathbf{X}*\mathbf{Y}} < d_{\mathbf{Y}} + m - 1$, then there exist $d_{\mathbf{Y}} + m - 2$ columns of the matrix $\mathbf{X} * \mathbf{Y}$ which are linearly dependent over field $\mathbb{F}$. Let us denote the i-th column of $\mathbf{X} * \mathbf{Y}$ as $\mathbf{z}_i$, and let $\{\mathbf{z}_i \,|\, i \in I\}$ be such a set of $d_{\mathbf{X}} + m - 2$ columns. Then there exist element $\lambda_i \in \mathbb{F}$ not all zeros such that

$$\sum_{i \in I} \lambda_i \mathbf{z}_i = \mathbf{0}.$$

This implies

$$\sum_{i \in I} \lambda_i x_{r,i} \mathbf{y}_i = \mathbf{0}, \tag{4.1}$$

for $r = 1, 2, \ldots, m$.

Now consider the sub-matrix of $\mathbf{X}$ consisting of the columns $\mathbf{x}_i$, $i \in I$. This sub-matrix has size $m \times (d_{\mathbf{B}} + m - 2)$, and by hypothesis, it will contain a non-singular $m \times m$ sub-matrix.

Let J be an m-subset of I such that the columns $\mathbf{x}_j$, $j \in J$, form such a non-singular square sub-matrix of $\mathbf{X}$, and let $\det(J)$ denote the determinant of this matrix. For $j \in J$ and $i \in I$, let $\det_{i,j}(J)$ denote the determinant which is obtained by replacing column $\mathbf{x}_j$ in $\det(J)$ by $\mathbf{x}_i$.

Multiplication both members of 4.1 by the cofactor of the element $x_{r,j}$ in the determinant $\det(J)$, and then taking the sum over r yields the following identity:

$$\sum_{i \in I} \lambda_i \det_{i,j}(J) \mathbf{y}_i = \mathbf{0}, j \in J. \tag{4.2}$$

It is clear that $\det_{i,j}(J)$ vanishes if $i \in J \setminus \{j\}$. Hence the sum 4.2 contains at most $(d_{\mathbf{X}} + m - 2) - (m - 1) = d_{\mathbf{X}} - 1$ nonzero terms. However, since any $d_{\mathbf{X}} - 1$ columns in the matrix $\mathbf{X}$ are linearly independent, it follows that every term in this sum must vanish. So we have

$$\lambda_i \det_{i,j}(J) = 0, i \in I, j \in J. \tag{4.3}$$

If $i = j \in J$ in 4.3, then $\lambda_i = 0$ for each $j \in J$. Therefore, at most $d_{\mathbf{X}} - 2$ of the elements λ_i are nonzero. Using again that any $d_{\mathbf{X}} - 2$ columns in the matrix $\mathbf{X}$ are linearly independent, we deduce from 4.1 that

$$\lambda_i x_{r,i} = 0, i \in I, r = 1, 2, \ldots, m. \tag{4.4}$$

We assumed that some λ_i is nonzero. From 4.4, it follows that the corresponding column $\mathbf{x}_i$ in $\mathbf{X}$ must vanish element-wise. This is a contradiction. So, $d_{\mathbf{X}*\mathbf{Y}} \geq d_{\mathbf{Y}} + m - 1$. $\square$

Now, we will discuss the application of Theorem 4.15 to cyclic codes. Let C be a cyclic code of length n over $\mathbb{F}_q$, and let m be the multiplicative order of q modulo n. Let $B = \{j_1, j_2, \ldots, j_l\}$ be any subset of $\mathbb{Z}_n$. We shall say that B is a *consecutive set* of length l if there exist a nonzero integer j such that $B = \{j, j+1, \ldots, j+l-1\}$.

If $A = \{i_1, i_2, \ldots, i_t\} \subseteq \mathbb{Z}_n$ and α is a primitive n-th root of unity in $\mathbb{F}_{q^m}$, then

$$H_A = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2\cdot i_1} & \ldots & \alpha^{(n-1)\cdot i_1} \\ 1 & \alpha^{i_2} & \alpha^{2\cdot i_2} & \ldots & \alpha^{(n-1)\cdot i_2} \\ \vdots & & & & \vdots \\ 1 & \alpha^{i_t} & \alpha^{2\cdot i_t} & \ldots & \alpha^{(n-1)\cdot i_t} \end{pmatrix},$$

is a $t \times n$ matrix over $\mathbb{F}_{q^m}$. Clearly, H_A is the parity check matrix for the cyclic code C over $\mathbb{F}_q$ having A as defining set of zeros.

Recall Definition 3.27, let $C^*(A)$ be the cyclic code over $\mathbb{F}_{q^m}$ with H_A as parity check matrix, and let this code have minimum distance d_A. Since C is a subfield sub-code of $C^*(A)$, hence the minimum distance of C is at least d_A.

Definition 4.16. Let N and M be subset of $\mathbb{Z}_n$. Define $N + M = \{x + y \,|\, x \in N, y \in M\}$. If $M = \{i_1, i_2, \ldots, i_t\}$, where $i_1 < i_2 < \ldots < i_t$, then $\overline{M}$ is defined as the consecutive set with i_1 as the first element and i_t as last element.

Theorem 4.17. *Let C be cyclic code of length n. If A is defining set of C with minimum distance d_A and if B is a subset of $\mathbb{Z}_n$ such that $|\overline{B}| \leq |B| + d_A - 2$, then the minimum distance of C is at least $\delta \geq |B| + d_A - 1$.*

Proof. Using the notation of Theorem 4.15, define $\mathbf{Y} = H_A$ and $\mathbf{X} = H_B$. Then $\mathbf{X} * \mathbf{Y} = H_{B+A}$. Since A is non-empty, $d_A \geq 2$. Hence, by Theorem 4.17 follows from Theorem 4.15, if in the matrix H_B every $|B| \times (|B| + d_A - 2)$ sub-matrix has full rank. To complete the proof, we need to show that the matrix H_B has full rank if $|\overline{B}| \leq |B| + d_A - 2$ for some consecutive set $\overline{B}$ containing B. Note that H_B is a sub-matrix of $H_{\overline{B}}$, and that in the matrix $H_{\overline{B}}$ every $|\overline{B}| \times |\overline{B}|$ is non-singular, since the determinant of such a matrix is of Vandermonde type. Hence it is clear that every $|B| \times |\overline{B}|$ sub-matrix of H_B has full rank. Since $|\overline{B}| \leq |B| + d_A - 2$, this implies that also every $|B| \times (|B| + d_A - 2)$ sub-matrix of H_B has full rank. □

As immediate result,

Corollary 4.18. *Let A, B, and $\overline{B}$ be as in Theorem 4.17. If A is consecutive, then $|\overline{B}| < |B| + |A|$ implies $\delta \geq |B| + |A|$.*

Proof. If A is consecutive set, then $d_A = |A| + 1$. The result follows from Theorem 4.17, by substitute d_A into $\delta \geq |B| + d_A - 1$. This yields $\delta \geq |B| + |A|$. □

Definition 4.19. Let I be a subset of $\mathbb{Z}_n$. Let $d_{Roos}(I)$ be the largest number γ such that there exist non-empty subsets A and $B = \{i_1, \ldots, i_t\}$ of $\mathbb{Z}_n$ and let $\bar{B}$ be a consecutive set such that its first element is i_1 and its last element is i_t with $B \subseteq \overline{B}$, $A + B \subseteq Z(C)$, where $A + B = \{a + b \,|\, a \in A,\, b \in B\}$, and $|\overline{B}| \leq |B| + d_A - 2 = \gamma - 1$. Let C be a cyclic code of length n. Then $d_{Roos}(C)$ is denoted by $d_{Roos}(I)$.

Theorem 4.20 (The Roos bound). *The minimum distance of a cyclic code C of length n is at least $d_{Roos}(C)$.*

Proof. This is an immediate consequence of Definition 4.19 and Theorem 4.17. □

Proposition 4.21. *Let I be a subset of $\mathbb{Z}_n$. Then $d_{Roos}(I) \geq d_{HT}(I)$.*

Proof. Let A and B be non-empty consecutive subsets of $\mathbb{Z}_n$ of size $\delta - 1$ and s, respectively. To be precise, the HT bound is the Roos bound with $A = \{b+i\cdot a_1 \,|\, 0 \leq i \leq \delta-2\}$ and $B = \{j \cdot a_2 \,|\, 0 \leq j \leq s\}$. Now $d_A \geq 2$, and since A is not empty. By Theorem 4.17, $d_{Roos}(J) \geq |B| + d_A - 2$. Hence $d_{Roos}(J) \geq d_{HT}(J)$. □

Remark 4.22. The BCH bound follows from Corollary 4.18 by taking for B the set $\{1\}$. Similarly, the HT bound follows by taking for B a consecutive set. Observe also that Corollary 4.18 improves HT bound by allowing in the set B the occurrence of $|A|-1$ *holes*, instead of B being consecutive.

The following example was taken from [16],

Example 4.23. Let C be the binary cyclic code of length 21 with defining set $\{1, 3, 7, 9\}$. The complete defining set of C is

$$Z(C) = \{1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 14, 15, 16, 18\}.$$

Now take $A = \{2, 3, 4\}$. Then $d_A \geq 4$. Let $B = \{4j \,|\, j = 0, 1, 3, 5\} = \{0, 4, 12, 20\}$. Then

$$A + B = \{2, 6, 14, 1\} \cup \{3, 7, 15, 2\} \cup \{4, 8, 16, 3\},$$

which gives $A+B \subseteq Z(C)$, in other words $A+B$ is in the set of zeros of C. So, we have $|\bar{B}| = 6 \leq |B| + d_A - 2$. Thus by the Roos bound, the minimum distance of C is $d_{Roos} \geq |B| + d_A - 1 = 7$.

Note that, by the BCH bound, the minimum distance of C is $d_{BCH} = 5$ and by the HT bound with $b = 6$, $a_1 = 1$, $a_2 = 16$, $s = 1$, and $\delta = 5$, the minimum distance of C is $d_{HT} = 6$.

Theorem 4.24. *Let C be a q-ary cyclic code of length n with $I \subseteq \mathbb{Z}_n$ as defining set and let d_{BCH}, d_{HT}, d_{HTR}, and d_{Roos} be the lower bounds on the minimum distance of C by the BCH bound, HT bound, HT and Roos bound, and Roos bound, respectively. Then*

$$d_{BCH}(I) \leq d_{HT}(I) \leq d_{HTR}(I) \leq d_{Roos}(I).$$

Proof. As a consequence of Theorem 4.9, $d_{BCH}(I) \leq d_{HT}(I)$. As consequence of Proposition 4.11, $d_{HT}(I) \leq d_{HTR}(I)$. And as a consequence of Theorem 4.21, $d_{HT}(I) \leq d_{Roos}(I)$. □

4.4. AB method

In this section, we will discuss a method of estimating the minimum distance of a cyclic code. The method is due to J.H. van Lint and R.M. Wilson [20]. Consider a product operation of matrices,

$$\mathbf{A} * \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{b}_1 & a_{12}\mathbf{b}_2 & \dots & a_{1n}\mathbf{b}_n \\ a_{21}\mathbf{b}_1 & a_{22}\mathbf{b}_2 & \dots & a_{2n}\mathbf{b}_n \\ \cdot & \cdot & \dots & \cdot \\ a_{m1}\mathbf{b}_1 & a_{m2}\mathbf{b}_2 & \dots & a_{mn}\mathbf{b}_n \end{pmatrix},$$

where $\mathbf{A}$ is a matrix of size $m \times n$ with entries a_{ij}, for $1 \leq i \leq m$, $1 \leq j \leq n$ and $\mathbf{B}$ is a matrix with columns $\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_n$. Note that, if $\mathbf{a} = (a_1, a_2, \dots, a_n)$ and $\mathbf{b} = (b_1, b_2, \dots, b_n)$, then $\mathbf{a} * \mathbf{b} = (a_1b_1, a_2b_2, \dots, a_nb_n)$. Therefore, $\mathbf{A} * \mathbf{B}$ is a matrix with its rows all the products $\mathbf{a} * \mathbf{b}$, where $\mathbf{a}$ is a row of matrix $\mathbf{A}$ and $\mathbf{b}$ is a row of $\mathbf{B}$.

Theorem 4.25. *If a linear combination, with nonzero coefficients, of the columns of* $\mathbf{A} * \mathbf{B}$ *is* $\mathbf{0}$*, then*

$$\operatorname{rank}(\mathbf{A}) + \operatorname{rank}(\mathbf{B}) \leq n.$$

The proof of Theorem 4.25 is due to [19].

Proof. If the coefficients in the linear combination are λ_j for $j = 1, \ldots, n$, then multiply column j by $\mathbf{B}$ by λ_j for $j = 1, \ldots, n$. This yields a matrix $\mathbf{B}'$ with has the same rank as $\mathbf{B}$. The condition of the theorem states that every row of $\mathbf{A}$ has inner product 0 with every row of $\mathbf{B}'$. Since this implies that $\operatorname{rank}(A) + \operatorname{rank}(\mathbf{B}')$ is at most n. And this proves the theorem. □

Next, we are going to give a theorem to find the minimum distance of a large number of cyclic codes. If $\mathbf{c}$ is a codeword in a cyclic code, then the *support* J of $\mathbf{c}$ is the set of coordinate positions j such that $c_j \neq 0$.

Definition 4.26. If M is a matrix with n columns and $J \subseteq \{1, 2, \ldots, n\}$, then M_J is the sub-matrix of M consisting of the columns indexed by elements of J.

The following theorem is an immediate corollary of Theorem 4.25.

Theorem 4.27. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices with entries from the field* $\mathbb{F}$*, and let* $\mathbf{A} * \mathbf{B}$ *be a parity check matrix for the code* C *over* $\mathbb{F}$*. Let* $\mathbf{c}$ *be a codeword. If* J *is the support of a codeword* $\mathbf{c}$*, then*

$$\operatorname{rank}(\mathbf{A}_J) + \operatorname{rank}(\mathbf{B}_J) \leq |J|.$$

In particular, C *has minimum distance* $\geq \delta$ *if* $\operatorname{rank}(\mathbf{A}_J) + \operatorname{rank}(\mathbf{B}_J) > |J|$ *for every subset* J *of* $\{1, 2, \ldots, n\}$ *for which* $|J| < \delta$*.*

Proof. This is an immediate corollary of Theorem 4.25. □

For additional reading and proofs of Theorem 4.25 and Theorem 4.27, see [20].

Now, we would like to apply those theorems for the analysis of the minimum distance of cyclic codes. If $I = \{i_1, \ldots, i_l\} \subseteq Z_n$ and α is a primitive n-th root of unity such that a cyclic code C of length n

$$c(x) \in C \Leftrightarrow \forall j \in I \,:\, c(\alpha^j) = 0,$$

then I is a *defining set* for C. If I is the maximal defining set for C, then I is called *complete*.

Definition 4.28. Let α be a primitive n-th root of unity. Let us denote by $M(I)$ be the matrix of size l by n that has $1, \alpha^{i_k}, \alpha^{2i_k}, \ldots, \alpha^{(n-1)i_k}$ as its k-th row, that is

$$M(I) = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \ldots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \ldots & \alpha^{(n-1)i_2} \\ . & . & . & \ldots & . \\ 1 & \alpha^{i_l} & \alpha^{2i_l} & \ldots & \alpha^{(n-1)i_l} \end{pmatrix}.$$

We consider the matrix $M(I)$ as a parity check matrix for a cyclic code C^* over $\mathbb{F}_{q^m}$. Now, let $\mathbf{A} = M(I_1)$ and $\mathbf{B} = M(I_2)$. If I_1 and I_2 are subsets of $\mathbb{Z}_n$, then every row of $M(I_1) * M(I_2)$ is also a row of $M(I_1 + I_2)$, see Definition 4.16 for notation $I_1 + I_2$.

Lemma 4.29. *If* $|I_2| = \delta - 1$, *then* $M(i, i+1, \ldots, i+\delta-2)_{I_2}$ *has rank* $\delta - 1$.

As a consequence of Lemma 4.29, we have the following corollary,

Corollary 4.30. *If* $i_1 < i_2 < \ldots < i_k = i_1 + t - 1$, *and* $|J| = t$, *then* $M(\{i_1, \ldots, i_k\})_J$ *has rank* k.

Hence, by Corollary 4.30, if $\gcd(b, n) = 1$ and $S = \{i_1, i_2, \ldots, i_k\} \cap \{bi, b(i+1), \ldots, b(i+t-1)\}$ and $|J| \geq t$, then $\text{rank}(M(I)_J) \geq |S|$. If R is the defining set of a cyclic code, we try to find suitable sets I_1 and I_2 such that $I_1 + I_2 \subseteq R$.

How the above theory can be implemented to determine a lower bound of a cyclic codes is answered in the following lemma.

Remark 4.31. For $J \subset \{1, 2, \ldots, n\}$,

$$\text{rank}(M(i_1, \ldots, i_k)_J) = \text{rank}(M(i_1 + j, \ldots, i_k + j)_J).$$

Lemma 4.32. *Let* C *be a code with defining set* R, *and let* $\mathbf{c} \in C$ *be a codeword with support contained in* I *such that* $\mathbf{c}$ *does not belong to the code with defining set* $R \cup \{j\}$. *Then for any set* $\{i_1, \ldots, i_k\} \subset R$ *we have*

$$\text{rank}(M(i_1, \ldots, i_k, j)_I) = 1 + \text{rank}(M(i_1, \ldots, i_k)_I).$$

The proof of Lemma 4.32 can be found in [20]. Using Lemma 4.32, we determine a lower bound of the minimum distance of cyclic codes based on its defining set. The following example was taken from Example 6 of [20].

Example 4.33. Let C be the binary cyclic code of length 51 with generator $g(x) = m_1(x)m_3(x)m_{19}(x)$. The complete defining set of code C is

$$R = \{i \,|\, i = 1, 2, 3, 4, 6, 8, 12, 13, 16, 19, 24, 25, 26, 27, 32, 35, 38, 39, 43, 45, 47, 48, 49, 50\}.$$

We wish to show that C has minimum distance of $d \geq 9$.

(1) Suppose we add zero to the set R, then we have nine consecutive elements in R. By the BCH bound, we get that the minimum distance of the even weight subcode of C is at least 10.

(2) The set R contains $A = \{i \,|\, i = 1, 2, 3, 4\}$ and $B = \{j \,|\, j = 0, 23, 46\}$. Hence by the HT bound, we get the minimum distance $d \geq 7$. We will show this by contradiction. Suppose $|J| = 7$ and let J be the support of a codeword in C of minimum weight. Note that, if we add 5 into R, then we have to add $5 \cdot 2^5 \bmod 51 = 7$ into R. It means that in R, we have eight consecutive elements. And by the BCH bound, it yields that J is not the support of the codeword with $R \cup \{5\}$ as defining set.

We will apply Lemma 4.32 and Remark 4.31;

$$\begin{aligned}
\text{rank}(M(1,2,3,4,24)_J) &= \text{rank}(M(2,3,4,5,25)_J) \\
&= 1 + \text{rank}(M(2,3,4,25)_J) \\
&= 1 + \text{rank}(M(3,4,5,26)_J) \\
&= 2 + \text{rank}(M(3,4,26)_J) \\
&= 2 + \text{rank}(M(4,5,27)_J) \\
&= 3 + \text{rank}(M(4,27)_J) \\
&= 5.
\end{aligned}$$

If we take $A = \{i \,|\, i = 1, 2, 3, 4, 24\}$ and $B = \{j \,|\, j = 0, 23, 46\}$, then rank of $M(B)_I$ is equal to 3 by Lemma 4.29. Hence, we have $\text{rank}(M(A)_J) + \text{rank}(M(B)_J) = 5 + 3 = 8 > |J| = 7$. This is a contradiction with Theorem 4.27. It means that there is a codeword in C of minimum weight strictly greater than 7. If a codeword has weight 8, then it is an element of the even weight subcode. We know that the even weight subcode has minimum distance $d \geq 10$ by (1). So, the minimum distance is at least 9.

Example 4.33 shows us a method to determine the lower bound of minimum distance of cyclic codes. This method called *shifting*.

4.5. Algorithms computing the bounds

In this section, we discuss the algorithm that we implemented in C++ to compute the bounds. Let C be a q-ary cyclic code of length n with complete defining set $Z(C)$.

4.5.1 The BCH bound

Recall the Definition 4.1 that a cyclic code C of length n and designed distance δ is the largest possible cyclic code having zeros α^b, α^{b+1}, $\ldots$, $\alpha^{b+\delta-2}$, where $\alpha \in \mathbb{F}_{q^m}$ is a primitive n-th root of unity, b is a non-negative integer, and m is the multiplicative order q modulo n. The minimum distance of C is at least δ.

The algorithm to compute the lower bound on the minimum distance of cyclic codes based on the BCH bound is follows directly from Definition 4.3 and Theorem 4.4. The following algorithm has been implemented in C++ and tested on q-ary cyclic codes of various length.

Algorithm 1 BCH bound

procedure BCH$(n, q, Z(C))$
    $d_{BCH} \leftarrow 0$;
    for $b \in \mathbb{Z}_n^+$ **do**
        for a, where $\gcd(a, n) = 1$ **do**
            $i \leftarrow 0$;
            repeat
                $tmp \leftarrow_n b + i \cdot a$;
                $i \leftarrow i + 1$;
            until $tmp \notin Z(C)$ AND $i \leq n$;
            $\delta \leftarrow i + 1$;
            if $\delta - 1 > d_{BCH}$ **then**
                $d_{BCH} \leftarrow \delta - 1$;
            end if
        end for
    end for
    return d_{BCH} ▷ The BCH bound
end procedure

4.5.2 The HT bound

Given $Z(C)$, compute $tmp = b + i \cdot a_1 + j \cdot a_2$ such that $tmp \in Z(C)$ for $i = 0, 1, 2, 3, \ldots, \delta - 2$ and $j = 0, 1, 2, \ldots, s$ with $\gcd(n, a_1) = 1$ and $\gcd(n, a_2) = 1$, where $b \in \mathbb{Z}_n^+$. The algorithm to compute lower bound on the minimum distance of cyclic codes based on the HT bound follows directly from Definition 4.7 and Theorem 4.8. The following algorithm has been implemented in C++ and tested on q-ary cyclic codes of various length.

Algorithm 2 HT bound

procedure HT$(n, q, Z(C))$
    $d_{HT} \leftarrow 0$;
    for $b \in \mathbb{Z}_n^+$ **do**
        for a_1, where $\gcd(a_1, n) = 1$ **do**
            $i \leftarrow 0$;
            repeat
                $tmp \leftarrow_n b + i \cdot a_1$;
                $i \leftarrow i + 1$;
            until $tmp \notin Z(C)$ AND $i \leq n$;
            $\delta \leftarrow i + 1$;
            while $\delta > 2$ **do**
                for a_2, with $\gcd(a_2, n) = 1$ **do**
                    $vtmp \leftarrow \emptyset$;
                    while $vtmp \subseteq Z(C)$ **do**
                        $j \leftarrow 0$
                        for $i \leftarrow 0, 1, 2, \ldots, \delta - 2$ **do**
                            $tmp \leftarrow_n b + i \cdot a_1 + j \cdot a_2$;
                            $vtmp \leftarrow vtmp \cup \{tmp\}$;
                        end for
                        $j \leftarrow j + 1$;
                    end while
                    $s \leftarrow j - 1$;
                    if $\delta + s > d_{HT}$ **then**
                        $d_{HT} \leftarrow \delta + s$;
                    end if
                end for
                $\delta \leftarrow \delta - 1$;
            end while
        end for
    end for
    return d_{HT} ▷ The HT bound
end procedure

4.5.3 The HT-Roos bound

The HT-Roos bound is an improvement of the HT bound by C. Roos in [15]. The original HT bound considers that a_1 and a_2 must relatively prime to n. In the HT-Roos bound considers that $\gcd(n, a_1) = 1$, but $\gcd(n, a_2) < \delta$. The algorithm to compute lower bound on the minimum distance of cyclic codes based on the HT-Roos bound follows directly from the Definition 4.12 and Theorem 4.13. The following algorithm has been implemented in C++ and tested on q-ary cyclic codes of various length.

Algorithm 3 HTR bound

```
procedure HTR(n, q, Z(C))
    d_HTR ← 0;
    for b ∈ Z_n^+ do
        for a_1, where gcd(a_1, n) = 1 do
            i ← 0;
            repeat
                tmp ←_n b + i · a_1;
                i ← i + 1;
            until tmp ∉ Z(C) AND i ≤ n ;
            δ ← i + 1;
            while δ > 2 do
                for a_2, with gcd(a_2, n) < δ do
                    vtmp ← ∅;
                    while vtmp ⊆ Z(C) do
                        j ← 0
                        for i ← 0, 1, 2, ..., δ − 2 do
                            tmp ←_n b + i · a_1 + j · a_2;
                            vtmp ← vtmp ∪ {tmp};
                        end for
                        j ← j + 1;
                    end while
                    s ← j − 1;
                    if δ + s > d_HTR then
                        d_HTR ← δ + s;
                    end if
                end for
                δ ← δ − 1;
            end while
        end for
    end for
    return d_HTR                                        ▷ The HT-Roos bound
end procedure
```

4.5.4 The Roos bound

Let A be subset of $\mathbb{Z}_n$, and let H_A be a parity check matrix for C having A as defining set of zeros. The minimum distance of C will be denoted as d_A.

Let $B = \{i_1, i_2, \ldots, i_t\}$ be another subset of $\mathbb{Z}_n$ and let $\bar{B}$ be a consecutive set with the first element of $\bar{B}$ is equal to i_1 and the last element of $\bar{B}$ is equal to i_t, i.e. $\bar{B} = \{i_1, i_1 + 1, \ldots, i_1 + t - 1 = i_t\}$.

By Theorem 4.17, if $B \subseteq \mathbb{Z}_n$ and there exist a consecutive set $\bar{B}$ with $|\bar{B}| \leq |B| + d_A - 2$, then the minimum distance of C is at least $|B| + d_A - 1$.

In general, to compute d_A is not an easy task. We need to build the parity check matrix H_A and then compute the rank of H_A. The rank of the parity check matrix determined how many columns of the matrix that are linearly independent. Let d_A be the maximum number such that $d_A - 1$ columns of H_A are linearly independent. By Theorem 3.28, the minimum distance of C with defining set A is d_A.

Our implementation on the Roos bound is based on Corollary 4.18, which is assuming that A is a consecutive defining set. By the BCH bound, $d_A = |A| + 1$. Given $Z(C)$, compute consecutive element set A. And for each A, compute B such that $A + B \subseteq Z(C)$ with $|\bar{B}| \leq |B| + d_A - 2$, where $\bar{B}$ is a consecutive element set with its first element is the first element of B and its last element is the last element of B, in other words, $\bar{B}$ is a consecutive element set such that $B \subseteq \bar{B}$. By the Roos bound, the minimum distance of C is $d_{ROOS} \geq |B| + d_A - 1$. Note that, since we take A such that A is consecutive, this means d_{ROOS} is the lower bound of the Roos bound. The algorithm to compute the minimum distance of cyclic codes based on the Roos bound is directly follows from Definition 4.19 and Theorem 4.20.

Algorithm 4 Roos bound

```
procedure Roos(n, q, Z(C))
    d_ROOS ← 0;
    for b ∈ Z(C) do
        for a_1, where gcd(a_1, n) = 1 do
            i ← 0;
            d_A ← 0;
            A ← ∅;
            repeat
                tmp ←_n b + i · a_1;
                A ← A ∪ {tmp};
                i ← i + 1;
            until tmp ∈ Z(C) AND i ≤ n ;
            while A ≠ ∅ do
                d_A ← |A| + 1;
                for a_2, where gcd(a_2, n) = 1 do
                    B ← ∅;
                    J ← ∅;
                    j ← 0;
                    while j < n do
                        for a ∈ A do
                            tmp2 ←_n j · a_2;
                            tmp3 ←_n a + tmp2;
                            if tmp3 ∈ Z(C) then
                                B ← B ∪ {tmp2};
                                J ← J ∪ {j};
                            end if
                        end for
                        j ← j + 1;
                    end while
                    if J = ∅ then
                        d_ROOS ← d_A;
                    else
                        △ ← max{J} − min{J};
                        if △ ≤ |B| + d_A − 2 AND |B| + |A| > d_ROOS then
                            d_ROOS ← |B| + d_A − 1;
                        end if
                    end if
                end for
                Remove the last element of A;
            end while
        end for
    end for
    return d_ROOS;                                        ▷ The Roos bound
end procedure
```

5

The Shift bound

In this chapter, we discuss a different approach on determining a lower bound on the minimum distance of the cyclic codes due to T. Kasami [9], J.H. van Lint and R.M. Wilson [20]. In their paper [20], they proposed two methods on determining a lower bound on the minimum distance of cyclic codes, namely the AB method which is already discussed in Chapter 4 and the Shift bound, which will be discussed in this chapter. Also we give our contribution on an algorithm to compute the Shift bound.

5.1. Independent set

We describe the concept of an *independent set*.

Definition 5.1. Let Z be a subset of an abelian group $(G, +)$. Inductively we define a family of subsets of G, which we call *independent with respect to* Z, as follows:

1. $\emptyset$ is independent with respect to Z.
2. If A is independent with respect to Z, and $A \subseteq Z$, with $b \notin Z$, then $A \cup \{b\}$ is independent with respect to Z.
3. If A is independent with respect to Z, and $c \in G$, then $c + A$ is independent with respect to Z, where $c + A = \{c + a \mid a \in A\}$.

Remark 5.2. In the third item A is *shifted* to $c + A$.

Theorem 5.3. *Let $f(x) \in \mathbb{F}[x]/(x^n - 1)$ be a polynomial with coefficients in $\mathbb{F}$, and let $Z(f) = \{i \in \mathbb{Z}_n \mid f(\alpha^i) = 0\}$, where α is primitive n-th root of unity in an extension field of $\mathbb{F}$. Then the weight of $f(x)$ satisfies*

$$\mathrm{wt}(f(x)) \geq |A|,$$

for every subset A of $\mathbb{Z}_n$ that is independent with respect to $Z(f)$.

The proof was taken from [20].

Proof. Let $f(x)$ be a polynomial with coefficients in $\mathbb{F}$ and let be written as follows:

$$f(x) = \lambda_1 x^{i_1} + \ldots + \lambda_k x^{i_k},$$

where $\lambda_i \neq 0$ for all $1 \leq i \leq k$, hence $\mathrm{wt}(f(x)) = k$. Define the set of vectors

$$V(A) = \{(a^{i_1}, \ldots, a^{i_k}) \mid a = \alpha^i, i \in A\}.$$

To prove the theorem, we will show that if A is an independent set with respect to Z, then the vectors in the set $V(A)$ are distinct and linearly independent over $\mathbb{F}$. We will use induction to show it.

1. By definition of independent set, if $A = \emptyset$, then A is independent with respect to Z. So, the assertion is true if $A = \emptyset$.

2. Suppose the vectors in $V(A)$ are linearly independent with respect to Z and $A \subseteq Z$. Let $j \notin Z$ and $b = \alpha^j$. By definition, the set $A \cup \{j\}$ is an independent set with respect to Z. Note that, $f(\alpha^i) = 0$ for all $i \in A$ and $f(\alpha^j) \neq 0$ since $j \notin Z$.

   The inner product of the vector $(\lambda_1, \ldots, \lambda_k)$ with the vectors in $V(A)$ is equal to zero, but the inner product of the vector $(\lambda_1, \ldots, \lambda_k)$ with $(\alpha^{j \cdot i_1}, \ldots, \alpha^{j \cdot i_k})$ is equal to $f(\alpha^j)$ and not zero. Hence, vector $(\alpha^{j \cdot i_1}, \ldots, \alpha^{j \cdot i_k})$ is not in the span of $V(A)$, and thus the vectors in $V(A \cup \{j\})$ are linearly independent.

3. Suppose the vectors in $V(A)$ are linearly independent. Let $i \in \mathbb{Z}_n$ and $c = \alpha^i$. By the linear transformation of $V(A)$ with matrix $\mathrm{diag}(c^{i_1}, \ldots, c^{i_k})$, shows that $V(i + A)$ consists of linearly independent vectors.

□

Remark 5.4. Let I be a subset of $\mathbb{Z}_n$. The $\mathbb{F}_{q^m}$-linear code $C^*(I)$ is defined by

$$C^*(I) = \left\{ (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_{q^m}^n \;\middle|\; \sum_{k=0}^{n-1} c_k \alpha^{j \cdot k} = 0 \text{ for all } j \in I \right\}.$$

Let $C(I)$ be the $\mathbb{F}_q$-linear subfield subcode with $C(I) = C^*(I) \cap \mathbb{F}_q^n$. The codes $C^*(I)$ and $C(I)$ are cyclic with defining set I. The code $C(I \cup \{j\})$ is contained in $C(I)$ for every $j \in \mathbb{Z}_n$. Let I^* be the union of the cyclotomic cosets of $j \in I$. Then $C(I) = C(I \cup \{j\})$ for all $j \in I^*$. Hence $I \subseteq I^*$ and $C(I) = C(I^*)$. Then I^* is the complete defining set of $C(I)$, and we call I complete if $I = I^*$.

Definition 5.5. For a subset R of $\mathbb{Z}_n$, let $n(R)$ be the maximal size of a set which is independent with respect to R. Define the shift bound for a subset J of $\mathbb{Z}_n$ as follows :

$$d_{\text{shift}}(J) = \min\{n(R) \mid J \subseteq R \subseteq \mathbb{Z}_n \text{ and } R^* = R \neq \mathbb{Z}_n\},$$

where R^* follows from definition in Remark 5.4.

Theorem 5.6. *The minimum distance of $C(J)$ is at least $d_{\text{shift}}(J)$.*

Proof. This is an immediate consequence of Definition 5.5 and Theorem 5.3. □

To understand how the Theorem 5.3 works, we will illustrate how to find the shift bound given the zeros of the codewords, i.e. how to construct a sequence of independent sets. For simplicity, we will consider binary cyclic codes. The example was taken from [20], but the sequence of independence sets is a result of our program.

Example 5.7 (the Binary Golay Code). See Example 7 of [20]. Consider the binary cyclic code C of length 23 with generator $g(x) = m_1(x)$. The complete defining set is

$$Z(C) = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}.$$

Observe,

$$x^{23} - 1 = (x - 1)m_1(x)m_5(x).$$

Let $f(x) \in C$, and let $Z(f) = \{i \in \mathbb{Z}_{23} | f(\alpha^i) = 0\}$. If $m_5(x)$ divides $f(x)$, then either $f(x) = 0$ or $f(x) = m_1(x)m_5(x)$, which has weight 23. Therefore, we may assume that $m_5(x)$ does not divide $f(x)$, hence $Z(f)$ does not contain the zeros of $m_5(x)$. Construct a sequence A_0, A_1, A_2, ... of subsets of $\mathbb{Z}_{23}$ that are independent with respect to $Z(C)$. A sequence of independent sets is as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(C)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\begin{array}{llllll}
 & & & & & A_0 = \emptyset \\
a_0 = 0 & , & b_0 = 0 & \longrightarrow & & A_1 = \{0\} \\
a_1 = 1 & , & b_1 = 0 & \longrightarrow & & A_2 = \{1, 0\} \\
a_2 = 1 & , & b_2 = 5 & \longrightarrow & & A_3 = \{2, 1, 5\} \\
a_3 = 7 & , & b_3 = 22 & \longrightarrow & & A_4 = \{9, 8, 12, 22\} \\
a_4 = 4 & , & b_4 = 0 & \longrightarrow & & A_5 = \{13, 12, 16, 3, 0\} \\
a_5 = 13 & , & b_5 = 22 & \longrightarrow & & A_6 = \{3, 2, 6, 16, 13, 22\}
\end{array}$$

So, $n(Z(C)) \geq 6$. In fact the algorithm of Section 5.2 gives that equality holds.

Let C_0 be a sub-code of C by adding 0 into $Z(C)$, i.e. $Z_{C_0} = Z(C) \cup \{0\}$. Construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_0)$, $b_i \notin Z(C_0)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\begin{array}{llllll}
 & & & & & A_0 = \emptyset \\
a_0 = 0 & , & b_0 = 5 & \longrightarrow & & A_1 = \{5\} \\
a_1 = 1 & , & b_1 = 5 & \longrightarrow & & A_2 = \{6, 5\} \\
a_2 = 19 & , & b_2 = 5 & \longrightarrow & & A_3 = \{2, 1, 5\} \\
a_3 = 11 & , & b_3 = 5 & \longrightarrow & & A_4 = \{13, 12, 16, 5\} \\
a_4 = 11 & , & b_4 = 10 & \longrightarrow & & A_5 = \{1, 0, 4, 16, 10\} \\
a_5 = 2 & , & b_5 = 7 & \longrightarrow & & A_6 = \{3, 2, 6, 18, 12, 7\} \\
a_6 = 6 & , & b_6 = 5 & \longrightarrow & & A_7 = \{9, 8, 12, 1, 18, 13, 5\}
\end{array}$$

So, $n(Z(C_0)) \geq 7$. In fact the algorithm of Section 5.2 gives equality.

Let C_5 be a sub-code of C by the cyclotomic coset $\mathcal{C}_5$ into $Z(C)$, i.e. $Z_{C_5} = Z(C) \cup \mathcal{C}_5$. Construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_5)$, $b_i \notin Z(C_5)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$. And we get, $b_i = 0$ and $a_i = 1$, for $i = 0, \ldots, 22$. So, $n(Z(C_5)) = 23$.

By Definition 5.5,

$$\begin{aligned}
d_{shift}(C) &= \min\{n(Z(C)), n(Z(C_0)), n(Z(C_5))\}, \\
&= \min\{6, 7, 23\} \\
&= 6.
\end{aligned}$$

By Theorem 5.6, the minimum distance of C is at least 6.

It follows from theorem 5.3, that $\mathrm{wt}(f(x)) \geq |A_6| = 6$. If $\mathrm{wt}(f) = 6$, then $(x-1)|f(x)$, i.e. $0 \in Z(f)$. Next, construct a sequence of independent sets with respect to $Z(C) \cup \{0\}$ of size 7, then A_7 is an independent set with respect to $Z(f)$, which contradicts Theorem 5.3. It follows that $\mathrm{wt}(f) \geq 7$. In fact, the true minimum distance of this code is $d = 7$, and this code is called *the binary Golay code* and has parameters [23,12,7].

Theorem 5.8. $\quad d_{HT} \leq d_{shift}$.

Proof. We refer to [12] Proposition 2.8. □

Example 5.9. Let C be a binary cyclic code of length 23 with defining set $\{1\}$. From Example 5.7, we get $d_{shift} = 7$. Let $A = \{1,2,3,4\}$, hence $d_A = 5$. Let choose $a_2 = 1$ and $B = \{a_2 \cdot j \,|\, j = 0\}$, hence $\bar{B} = \{0\}$ which satisfy $|\bar{B}| \leq |B| + d_A - 2$. By the Roos bound, the minimum distance of C is $d_{Roos} \geq 1 + 5 - 1 = 5$.

In this case, we have $d_{shift} > d_{Roos}$.

We refer to Example 26.7 of M. van Eupen and J.H. van Lint [18].

Example 5.10. Let C be a ternary cyclic code of length 26 with the complete defining set $Z(C) = \{0, 13, 14, 16, 17, 22, 23, 25\}$. Construct a sequence of independent set with respect to $Z(C)$ as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(C)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$,

$$\begin{array}{llcl} & & & A_0 = \emptyset \\ a_0 = 0 & , \ b_0 = 1 & \longrightarrow & A_1 = \{1\} \\ a_1 = 25 & , \ b_1 = 1 & \longrightarrow & A_2 = \{0, 1\} \\ a_2 = 25 & , \ b_2 = 2 & \longrightarrow & A_3 = \{25, 0, 2\} \\ a_3 = 14 & , \ b_3 = 4 & \longrightarrow & A_4 = \{13, 14, 16, 4\} \\ a_4 = 9 & , \ b_4 = 1 & \longrightarrow & A_5 = \{22, 23, 25, 13, 1\} \end{array}$$

So, the maximum size of independent set with respect to $Z(C)$ is at least 5. Our algorithm gives that it is exactly 5. We also compute the maximum size of independent sets for all sub-codes of C. As summary, the minimum distance of C based on the Shift bound is $d_{shift} = 5$. Let $A = \{13, 14\}$, hence $d_A = |A| + 1$. Let choose $a_2 = 3$, and $B = \{a2 \cdot j \,|\, j = 0, 1, 3, 4\}$, hence $\bar{B} = \{0, 1, 2, 3, 4\}$, which satisfies $|\bar{B}| \leq |B| + d_A - 2$. By the Roos bound, the minimum distance of C is $d_{Roos} \geq 3 + 4 - 1 = 6$. In this case, we have $d_{shift} < d_{Roos}$.

Example 5.11. Let C and D be 7-ary cyclic codes of length 6 with defining sets $\{2, 4\}$ and $\{0, 2, 4\}$, respectively. Construct sequences of independent sets with respect to $Z(C)$ as follows; for each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(C)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$,

$$\begin{array}{llcl} & & & A_0 = \emptyset \\ a_0 = 0 & , \ b_0 = 0 & \longrightarrow & A_1 = \{0\} \\ a_1 = 2 & , \ b_1 = 0 & \longrightarrow & A_2 = \{2, 0\} \\ a_2 = 2 & , \ b_2 = 1 & \longrightarrow & A_3 = \{4, 2, 1\} \end{array}$$

So, $n(C) \geq 3$. Construct sequences of independent sets with respect to $Z(D)$ as follows; for each $i > 0$, $a_i + A_i \subseteq Z(D)$, $b_i \notin Z(D)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$,

$$\begin{array}{llcl} & & & A_0 = \emptyset \\ a_0 = 0 & , \ b_0 = 1 & \longrightarrow & A_1 = \{1\} \\ a_1 = 1 & , \ b_1 = 1 & \longrightarrow & A_2 = \{2, 1\} \end{array}$$

So, $n(D) \geq 2$. We have $n(C) > n(D)$, but $D \subseteq C$. Hence it is necessary to take the minimum in Definition 5.5.

The complete factorization of $x^6 - 1$ over $\mathbb{F}_7[x]$ is given by

$$(1+x)(2+x)(3+x)(4+x)(5+x)(6+x).$$

Let α be the primitive of 7-th of unity and it is the zero of minimal polynomial $2+x$. Then α^2 is the zero of minimal polynomial $3+x$ and α^4 is the zero of minimal polynomial $5+x$. Hence, the generator polynomial of C is $(x+3)(x+5) = x^2+x+1$, which has weight 3, and the generator polynomial of D is $(x+6)(x+3)(x+5) = x^3+6$, which has weight 2.

Another example that it is necessary to take the minimum in Definition 5.5.

Example 5.12. Let C be the binary cyclic code of length 21 with defining set $\{1,3,7,9\}$. Construct sequences of independent sets with respect to $Z(C)$ as follows; for each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(D)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$,

$$\begin{array}{lclcl}
 & & & & A_0 = \emptyset \\
a_0 = 0 & , & b_0 = 0 & \longrightarrow & A_1 = \{0\} \\
a_1 = 14 & , & b_1 = 0 & \longrightarrow & A_2 = \{14, 0\} \\
a_2 = 14 & , & b_2 = 0 & \longrightarrow & A_3 = \{7, 14, 0\} \\
a_3 = 1 & , & b_3 = 0 & \longrightarrow & A_4 = \{8, 15, 1, 0\} \\
a_4 = 1 & , & b_4 = 5 & \longrightarrow & A_5 = \{9, 16, 2, 1, 5\} \\
a_5 = 6 & , & b_5 = 5 & \longrightarrow & A_6 = \{15, 1, 8, 7, 11, 5\} \\
a_6 = 1 & , & b_6 = 0 & \longrightarrow & A_7 = \{16, 2, 9, 8, 12, 6, 0\} \\
a_7 = 16 & , & b_7 = 17 & \longrightarrow & A_8 = \{11, 18, 4, 3, 7, 1, 16, 17\} \\
a_8 = 11 & , & b_8 = 20 & \longrightarrow & A_9 = \{1, 8, 15, 14, 18, 12, 6, 7, 20\}
\end{array}$$

So, $n(Z(C)) \geq 9$. Our algorithm gives that it is exactly 9. Let C_0 be the sub-code of C by adding 0 into the defining set of C. Construct sequences of independent sets with respect to $Z(C)$ as follows; for each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(D)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$,

$$\begin{array}{lclcl}
 & & & & A_0 = \emptyset \\
a_0 = 0 & , & b_0 = 5 & \longrightarrow & A_1 = \{5\} \\
a_1 = 7 & , & b_1 = 5 & \longrightarrow & A_2 = \{12, 5\} \\
a_2 = 4 & , & b_2 = 5 & \longrightarrow & A_3 = \{16, 9, 5\} \\
a_3 = 20 & , & b_3 = 13 & \longrightarrow & A_4 = \{15, 8, 4, 13\} \\
a_4 = 20 & , & b_4 = 5 & \longrightarrow & A_5 = \{14, 7, 3, 12, 5\} \\
a_5 = 9 & , & b_5 = 13 & \longrightarrow & A_6 = \{2, 16, 12, 0, 14, 13\} \\
a_6 = 2 & , & b_6 = 10 & \longrightarrow & A_7 = \{4, 18, 14, 2, 16, 15, 10\} \\
a_7 = 14 & , & b_7 = 20 & \longrightarrow & A_8 = \{18, 11, 7, 16, 9, 8, 3, 20\}
\end{array}$$

So, $n(Z(C_0)) \geq 8$. Our algorithm gives that it is exactly 8. We $n(Z(C)) > n(Z(C_0))$, but $C_0 \subseteq C$. Hence by Definition 5.5, $d_{shift} = \min\{n(Z(C)), n(Z(C_0)), n(Z(C_5)), n(Z_{C_{0,5}})\} = 8$. In fact, the minimum distance of C is 8.

5.2. Algorithm to compute the Shift bound

5.2.1 Problem formulation

Now, we give an explanation on how to construct a sequence of independent sets. Let C be a cyclic code of length n with complete defining set $Z(C)$, where

$$Z(C) = \{i \in \mathbb{Z}_n \mid c(\alpha^i) = 0 \text{ for all } c(x) \in C\}.$$

Definition 5.13. Let

$$N(C) = \{i \in \mathbb{Z}_n \mid c(\alpha^i) \neq 0 \text{ for some } c(x) \in C\} = \mathbb{Z}_n \setminus Z(C).$$

Then

$$\mathbb{Z}_n = Z(C) \cup N(C).$$

Our goal is to determine the lower bound d_{shift} on the minimum distance for cyclic codes C using Theorem 5.6. As already explained in Section 5.1, we will use the concept of *independent set*, which is the same as shifting. Recall and rewrite Definition 5.1 of independent set in terms of cyclic codes.

Definition 5.14. Let $Z(C)$ be a subset of an abelian group $(G, +)$. Inductively we define a family of subsets of G, which we call *independent with respect to* $Z(C)$, as follows:

1. $\emptyset$ is independent with respect to $Z(C)$.
2. If A is independent with respect to $Z(C)$, and $A \subseteq Z(C)$, with $b \in N(C)$, then $A \cup \{b\}$ is independent with respect to $Z(C)$.
3. If A is independent with respect to $Z(C)$, and $c \in G$, then $c + A$ is independent with respect to $Z(C)$, where $c + A = \{c + a \mid a \in A\}$.

In order to determine the lower bound on the minimum distance of a cyclic codes, we need to find a sequence $(A_0, A_1, A_2, \ldots, A_i)$, where $A_0, A_1, A_2, \ldots, A_i$, with $|A_0| = 0$ and $|A_{k+1}| = |A_k| + 1$ for $k = 1, 2, \ldots, i$, are independent sets with respect to $Z(C)$ for each $i > 0$. Furthermore, we are searching for the maximum size of the sequence of independent sets.

Definition 5.15. Let C be a cyclic code of length n with complete defining set $Z(C)$. Let A be an independent set with respect to $Z(C)$. We define S_A as the set of shift elements of set A by

$$S_A = \{x \in \mathbb{Z}_n \mid x + A \subseteq Z(C)\}.$$

The idea of constructing a sequence of independent sets is as follows;

Step 1. If $A = \emptyset$, then by Definition 5.14 point 1, A is an independent set. And since $A = \emptyset \subseteq Z(C)$. If $b \in N(C)$, then by Definition 5.14 point 2, $A \cup \{b\} = \{b\}$ is an independent set with respect to $Z(C)$.

Step 2. If $A \neq \emptyset$ is an independent set, and $A \not\subseteq Z(C)$, then in order to extend the sequence, we can not directly apply Definition 5.14 point 2. There are two cases to consider;

1. If $S_A = \emptyset$, then there is no $a \in S_A$, such that $a + A \subseteq Z(C)$. So, we can not apply Definition 5.14 point 2 to extend A. In this case, A is maximal.
2. If $S_A \neq \emptyset$, then there is $a \in S_A$, such that $a + A \subseteq Z(C)$. By Definition 5.14 point 3, $a + A$ is an independent set with respect to $Z(C)$. And then we can apply Definition 5.14 point 2, such that $(a + A) \cup \{b\}$ is also an independent set with respect to $Z(C)$, where $b \in N(C)$.

Lemma 5.16. $S_{-a+X} = a + S_X$

Proof. Let $Y = -a + X$. Then

$$\begin{aligned} s \in S_Y &\Leftrightarrow s + Y \subseteq Z(C); \\ &\Leftrightarrow s + (-a + X) \subseteq Z(C); \\ &\Leftrightarrow (s - a) + X \subseteq Z(C); \\ &\Leftrightarrow s - a \in S_X; \\ &\Leftrightarrow s \in a + S_X. \end{aligned}$$

So, $S_Y = a + S_X$. And hence, $S_{-a+X} = a + S_X$ □

Lemma 5.17. *Let C be a cyclic code of length n with complete defining set $Z(C)$. Let A be an independent set with respect to $Z(C)$. If $a \in S_A$, $b \notin Z(C)$, then $\overline{A}$ and $\widetilde{A}$ are independent sets with respect to $Z(C)$, where $\overline{A} = (a + A) \cup \{b\}$ and $\widetilde{A} = A \cup \{b - a\}$. Furthermore, $S_{\overline{A}} = (S_A - a) \cap S_{\{b\}}$ and $S_{\widetilde{A}} = S_A \cap (S_{\{b\}} + a)$.*

Proof. If A is an independent set with respect to $Z(C)$, then by Definition 5.14 point 3, set $a + A$ is also an independent set with respect to $Z(C)$, with $a \in S_A$ such that $a + A \subseteq Z(C)$. And since $a + A$ is an independent set with respect to $Z(C)$, hence by Definition 5.14 point 2, for all $b \notin Z(C)$,

$$\overline{A} = (a + A) \cup \{b\}$$

is also an independent set with respect to $Z(C)$. We can write the above equation as

$$\overline{A} - a = A \cup \{b - a\} = \tilde{A}.$$

Hence $\widetilde{A}$ is also an independent set with respect to $Z(C)$. And by Lemma 5.16, we have that

$$S_{\tilde{A}} = a + S_{\bar{A}}.$$

And since $\tilde{A} = A \cup \{b - a\}$, obviously,

$$\begin{aligned} S_{\widetilde{A}} &= S_A \cap S_{\{b-a\}} \\ &= S_A \cap (S_{\{b\}} + a). \end{aligned}$$

□

Remark 5.18. There are several important points here to be made;

1. If A is an independent set with respect to $Z(C)$ with S_A is the set of shift element of A, then $\widetilde{A}$ is also an independent set with respect to $Z(C)$ with $S_{\widetilde{A}} = S_A \cap S_{\{b-a\}}$.

2. Because A is an independent set with respect to $Z(C)$, hence $\overline{A} = (a + A) \cup \{b\}$, $a \in S_A$, $b \notin Z(C)$ is the only way to make an independent set from A. Moreover, we can get $\overline{A}$ from $\widetilde{A}$, by $\overline{A} = \widetilde{A} + a$, where $a \in S_A$.

To construct a sequence of independent set with respect to $Z(C)$, we start with a set $A_0 = \emptyset$, and then we apply Lemma 5.17 iteratively. We summarize the statement in the following Table 5.1;

$$
\begin{array}{lll}
 & & A_0 = \emptyset \\
a_0 \in S_{A_0} & b_0 \notin Z(C) & A_1 = (a_0 + A_0) \cup \{b_0\} \\
a_1 \in S_{A_1} = S_{A_0} \cap S_{b_0} & b_1 \notin Z(C) & A_2 = (a_1 + A_1) \cup \{b_1\} \\
a_2 \in S_{A_2} = (S_{A_1} - a_1) \cap S_{b_1} & b_2 \notin Z(C) & A_3 = (a_2 + A_2) \cup \{b_2\} \\
a_3 \in S_{A_3} = (S_{A_2} - a_2) \cap S_{b_2} & b_3 \notin Z(C) & A_4 = (a_3 + A_3) \cup \{b_3\} \\
 & \vdots & \\
a_{i-2} \in S_{A_{i-2}} = (S_{A_{i-3}} - a_{i-3}) \cap S_{b_{i-3}} & b_{i-2} \notin Z(C) & A_{i-1} = (a_{i-2} + A_{i-2}) \cup \{b_{i-2}\} \\
a_{i-1} \in S_{A_{i-1}} = (S_{A_{i-2}} - a_{i-2}) \cap S_{b_{i-2}} & b_{i-1} \notin Z(C) & A_i = (a_{i-1} + A_{i-1}) \cup \{b_{i-1}\} \\
S_{A_i} = \emptyset & &
\end{array}
$$

Table 5.1: *Construction of a sequence of independent sets.*

Thus, the sequence $A_0, A_1, A_2, A_3, \ldots, A_{i-1}, A_i$ is a sequence of independent sets with respect to $Z(C)$.

Remark 5.19. From now on, instead of giving Table 5.1 in constructing a sequence of independent sets, we will use the following diagram to show a sequence of independent sets:

$$\emptyset \xrightarrow{\{a_0,b_0\}} \ldots \xrightarrow{\{a_{k-1},b_{k-1}\}} A_k = (a_{k-1} + A_{k-1}) \cup \{b_{k-1}\} \xrightarrow{\{a_k,b_k\}} \ldots \xrightarrow{\{a_{i-1},b_{i-1}\}} A_i,$$

for all $k = 1, 2, \ldots, i$.

To understand how to find sequence of independent set, we will illustrate the above explanation in an example. We will try to show the above result in the following example. See also Example 7 of [20] and Example 5.7.

Example 5.20. Let C be a binary cyclic code of length 23 with defining set $\{1\}$. Then the complete defining set is

$$Z(C) = \{1, 2, 4, 6, 8, 9, 12, 13, 16, 18\}.$$

Hence, the non-zeros of C is

$$N(C) = \mathbb{Z}_{23} \setminus Z(C) = \{0, 3, 5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\}.$$

We start with an empty set. By Definition 5.14 point 1, $A_0 = \emptyset$ is an independent set with respect to $Z(C)$.

Since $A_0 \subseteq Z(C)$, hence by applying Definition 5.14 point 2 to set A_0, we try to extend our sequence. Choose $b_0 = 0 \in N(C)$. Thus $A_1 = A_0 \cup \{b_0\} = \{0\}$.

Since $A_1 \nsubseteq Z(C)$, we must first apply Definition 5.14 point 3 to shift A_1. For $A_1 = \{0\}$, the set of shift elements set of A_1 is determine by

$$S_{A_1} = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}.$$

By Definition 5.14 point 3, if A_1 is an independent set, then $a_1 + A_1$ is also an independent set.

Choose $a_1 = 1 \in S_{A_1}$, hence $a_1 + A_1 = 1 + \{0\} = \{1\} \subseteq Z(C)$. Then we can apply Definition 5.14 point 2 to extend our sequence. Choose $b_1 = 0 \in N(C)$. Thus, we get $A_2 = (a_1 + A_1) \cup \{b_1\} = \{1, 0\}$.

With the same treatment as A_1, set $A_2 \nsubseteq Z(C)$, so first we must apply Definition 5.14 point 3 to shift A_2. For $A_2 = \{1, 0\}Z(C)$, the set of shift elements set of A_2 is

$$\begin{aligned} S_{A_2} &= (S_{A_1} - a_1) \cap S_{b_1} \\ &= \{0, 1, 2, 3, 5, 7, 8, 11, 12, 15, 17\} \cap \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\} \\ &= \{1, 2, 3, 8, 12\}. \end{aligned}$$

Choose $a_2 = 1 \in S_{A_2}$, hence $a_2 + A_2 = 1 + \{1, 0\} = \{2, 1\} \subseteq Z(C)$. Next, we apply Definition 5.14 point 2, and choose $b_3 = 5 \in N(C)$, hence $A_3 = (a_2 + A_2) \cup \{b_3\} = \{2, 1, 5\}$.

For A_3, the set of shift elements is

$$S_{A_3} = (S_{A_2} - a_2) \cap S_{b_2} = \{1, 7, 11\}.$$

Choose $a_3 = 7 \in S_{A_3}$, hence $a_3 + A_3 = \{9, 8, 12\} \subseteq Z(C)$. Choose $b_3 = 22 \in N(C)$, we get $A_4 = (a_3 + A_3) \cup \{b_3\} = \{9, 8, 12, 22\}$.

For A_4, the set of shift elements is

$$S_{A_4} = (S_{A_3} - a_3) \cap S_{b_3} = \{4, 17\}.$$

Choose $a_4 = 4 \in S_{A_4}$, hence $a_4 + A_4 = \{13, 12, 16, 3\} \subseteq Z(C)$. Let $b_4 = 0 \notin Z(C)$, then $A_5 = (a_4 + A_4) \cup \{b_4\} = \{13,\ 12,\ 16,\ 3,\ 0\}$.

For A_5, then the set of shift elements is

$$S_{A_5} = (S_{A_4} - a_4) \cap S_{b_4} = \{13\}.$$

Let $a_5 = 13 \in S_{A_5}$, hence $a_5 + A_5 = \{3, 2, 6, 16, 13\} \subseteq Z(C)$. Let $b_5 = 22 \in N(C)$, we get $A_6 = (a_5 + A_5) \cup \{b_5\} = \{3,\ 2,\ 6,\ 16,\ 13,\ 22\}$. For A_6, the set of shift elements is

$$S_{A_6} = (S_{A_5} - a_5) \cap S_{b_5} = \emptyset.$$

Since $S_{A_6} = \emptyset$, we can not continue. So the maximum size of independent set in the sequence is 6.

5.2.2 Backtracking algorithm

Recall the construction of a sequence of independent sets, as can be seen in Table 5.1. Let $\mathcal{S}$ be the solution space for the Shift bound problem. A solution for the Shift bound problem for the cyclic code C is a sequence $A_0, A_1, A_2, \ldots, A_{i-1}, A_i$, of independent sets with respect to $Z(C)$. Different choices of a_i's in S_{A_i} and b_i's in $N(C)$ lead to different independent sets. Different independent sets yields different sequences of independent sets. An algorithm that is suitable for this kind of problem is a backtracking algorithm. Because a backtracking algorithm systematically searches solutions through all possible options in the solution space.

Assume that a solution can be formulated as a vector of independent sets. Let $(A_1, \ldots, A_i)$ be a vector where each independent set A_i is selected from a finite set S_i, where S_i is a subset of $\mathcal{S}$. If S_i is the domain of A_i, then $\mathcal{S} = S_1 \cup S_2 \cup \ldots \cup S_m$ is the solution space of the problem, see Figure 5.1. A backtracking algorithm runs by traversing the domain of the vectors until it finds the solutions. When invoked, the algorithm starts with an empty vector. And at each stage, the

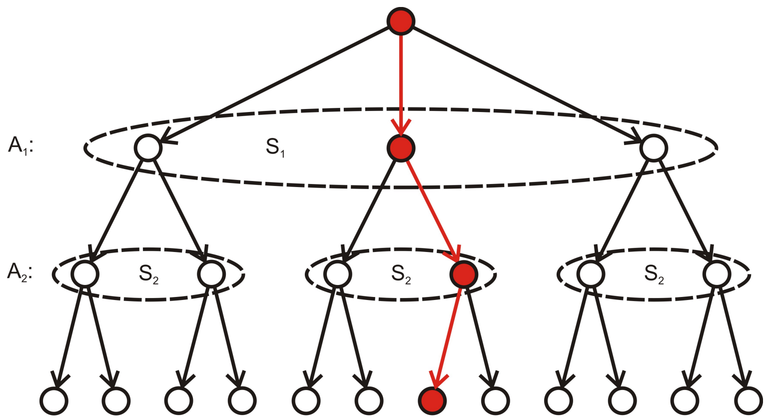


Figure 5.1: Solution space

algorithm extends the partial vector with a new independent set by applying Lemma 5.17. Upon reaching a partial vector $(A_1, A_2, \ldots, A_i)$, if this vector does not represents a partial solution, then the algorithm removes the trailing independent set from the vector and then proceeds by trying to extend the vector with new alternative independent set.

The following is a general backtracking algorithm as mentioned in [7].

Algorithm 5 A general backtracking algorithm

procedure BACKTRACK($\mathbf{x} = (A_0, A_1, A_2, \ldots, A_{i-1}, A_i)$)
  if $(A_0, A_1, A_2, \ldots, A_{i-1}, A_i)$ is solution **then**
    return $(A_0, A_1, A_2, \ldots, A_{i-1}, A_i)$
  end if
  for each v **do**
    if $(A_0, A_1, A_2, \ldots, A_{i-1}, A_i)$ is a sequence of independent sets **then**
      solution = backtrack($(A_0, A_1, A_2, \ldots, A_{i-1}, A_i, v)$)
      if solution $\neq \emptyset$ **then**
        return solution
      end if
    end if
  end for
end procedure

We introduce a notion *directed graph*.

Remark 5.21. We borrowed notation from [4], and [6]. A *directed graph* or *digraph* is a pair $\mathcal{D} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ the set of vertices and $\mathcal{A}$ consists of ordered pairs (u, v) of vertices. We call them *arcs* and denote them by $\vec{uv}$. We say the arc goes from u to v, leaves u, enters v, and call u the *tail* and v the *head* of the arc $\vec{uv}$. There may be more than one arc from a vertex u to a vertex v. For each subset $U \subseteq \mathcal{V}$, $\delta^{in}(U) = \{\vec{uv} \in \mathcal{A} \,|\, u \notin U, v \in U\}$, and $\delta^{out} = \{\vec{uv} \in \mathcal{A} \,|\, u \in U, v \notin U\}$.

Adapting Remark 5.21 into the Shift bound problem as follows;

Remark 5.22. Each node in the digraph represents an independent set with respect to $Z(C)$.

Remark 5.23. Each arc in the digraph is defined as the following relation (see also Figure 5.2);

$$\begin{gathered} A_{i+1} = (a_i + A_i) \cup \{b_i\}, \text{ where } a_i \in S_{A_i}, b_i \notin Z(C) \\ S_{A_{i+1}} = (S_{A_i} - a_i) \cap S_{b_i}, \end{gathered}$$

Table 5.2: *Definition of arc in the Shift bound problem.*

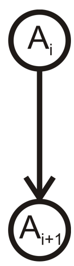


Figure 5.2: Arc

Let $\mathcal{D} = (\mathcal{V}, \mathcal{A})$ be a directed graph that has properties as mentioned in Remark 5.22 and Remark 5.23. Let $e = (u, v)$ be an arc in $\mathcal{A}$ where $u, v \in \mathcal{V}$. By Remark 5.23, u is a tail of e that contain A_i and v is a head of e that contain A_{i+1}. As consequence of Remark 5.22 and Remark 5.23, the Shift bound problem can be modeled as a *directed acyclic graph*. A *directed acyclic graph* or DAG is a directed graph with no directed cycle. It is also a generalization of *tree*.

Remark 5.24. We borrowed notations and definition from [17] and [10]. A *tree* is a computer data structure that emulates a tree structure with a set of linked nodes. Each node has zero or more *child* or *successor* node. A node that has a child is called the child's *parent* or *predecessor* node. The top most node in a tree is called the *root* node. Being the top most node, the root node will not have parent. Nodes at the bottom most level of the tree are called *leaf* nodes. Since they are at the bottom most level, they will not have any children. Since a tree does not contain a cycle and every child has *at most* one parent, hence it is a special case of a graph.

In the Shift bound problem, we will see that a node have predecessor more than one, except the root node. So instead of using the tree as a model for the Shift bound problem, we will use DAG as a model to the Shift bound problem. We also adapt terms that mentioned in Remark 5.24 for the DAG of the Shift bound problem.

Each node in DAG represents a *partial solution* to the Shift bound problem. By Remark 5.22, each node represents an independent set with respect to $Z(C)$. So, if $A_0 = \emptyset$ is an independent set with respect to $Z(C)$, then A_0 can be represented by the root node. Similarly, in the Shift bound, there is unique root node.

To solve the Shift bound problem, we must construct a sequence of independent sets A_0, A_1, ..., A_w for $w > 0$ such that $S_{a_w} = \emptyset$, by applying Lemma 5.17 recursively. When $S_{A_w} = \emptyset$ for $w > 0$, A_w is represented by a leaf node. So, a solution is a direct path from the *root* node to a *leaf* node. Therefore, the solution space $\mathcal{S}$ will consist of directed paths from the root node to leaf nodes. Note that, the objective of the Shift bound problem is searching for maximum size of directed path in the DAG.

The traversal of the DAG can be represented by a *depth-first* or *breadth-first search* traversal. In our implementation of the backtracking algorithm for the Shift bound problem, we use the *depth-first* for the traversal of the DAG.

For efficiency, it is unnecessary to store the entire graph in the algorithm. Instead, store only a directed path from a root node to the current working node. The backtracking algorithm creates and destroy the nodes dynamically as it explores the solution space $\mathcal{S}$. As we illustrated in Figure 5.1, the filled nodes are nodes that we store in the algorithm. The blank nodes are either unvisited nodes or visited nodes, which we do not store them in the algorithm.

Let $\mathbf{x} = (A_0, A_1, A_2, \ldots, A_{i-1}, A_i)$ be a vector solution where each element A_i is selected from a finite set S_i, where S_i is a subset of solution space $\mathcal{S}$, see Figure 5.1. It is clear that $\mathbf{x}$ is a partial solution. And it is represents a sequence of independent sets A_0, A_1, ..., A_i.

If $S_{A_i} = \emptyset$, then there is no $a_i \in S_{A_i} = \emptyset$ such that $(a_i + A_i) \subseteq Z(C)$. Since $A_i \not\subseteq Z(C)$, hence Definition 5.14 point 2 can not be applied. Moreover, the sequence $\mathbf{x}$ can not be extended. So, $\mathbf{x}$ is *maximal* sequence of independent sets. Maximum sequence is the largest sequence amongst these maximal sequences. And in order to solve the Shift bound problem, we need to find such sequence.

Let $d_{\max}$ be denoted the largest maximal sequence of independent sets. Let $\mathbf{x}_{\max}$ be the vector solution to store temporary maximum size of maximal sequence of independent sets. When a backtracking invoked, $d_{\max} = 0$. During the computation process, we will obtained maximal sequences of independent sets. Let $\mathbf{x} = (A_0, A_1, A_2, \ldots, A_{i-1}, A_i)$ be a maximal sequence of independent sets. If $|\mathbf{x}| > d_{\max}$, then update $d_{\max} = |\mathbf{x}|$ and $\mathbf{x}_{\max} = \mathbf{x}$.

When $\mathbf{x}$ is maximal, remove A_i from $\mathbf{x}$, and the backtracking algorithm will search for another alternative independent set $A_i^{new} = (a_{i-1}^{new} + A_{i-1}) \cup \{b_{i-1}^{new}\}$, where $a_{i-1}^{new} \in S_{A_{i-1}}$, $a_{i-1} \neq a_{i-1}^{new}$, and $b_{i-1}^{new} \notin Z(C)$. This process is called *backtrack*.

The algorithm will continue until all possible a_i's and b_i's to construct a sequence of independent sets considered. And when the algorithm terminates, $\mathbf{x}_{max}$ will be the largest sequence of independent sets.

We summarize the above explanation of the backtracking algorithm for finding the maximum size of sequence of independent sets in the following algorithm.

Algorithm 6 Backtracking algorithm for Shift bound

procedure $\textsc{Backtrack}(\mathbf{x} = (A_0, A_1, A_2, \ldots, A_{i-1}, A_i))$
  $A \leftarrow A_i$
  for $a \in S_A$ AND $b \notin Z(C)$ **do**
    $\overline{A} \leftarrow (a + A) \cup \{b\}$
    $\mathbf{x} \leftarrow \mathbf{x} \cup \{\overline{A}\}$
    $S_{\overline{A}} \leftarrow (S_A - a) \cap S_b$
    if $S_{\overline{A}} = \emptyset$ **then**
      if $|\mathbf{x}| > |\mathbf{x}_{max}|$ **then**
        $\mathbf{x}_{max} \leftarrow \mathbf{x}$
      end if
      backtrack($\mathbf{x} \leftarrow \mathbf{x} \setminus \overline{A}$)
    end if
    backtrack($\mathbf{x}$)
  end for
end procedure

Example 5.25. Figure 5.3 illustrates the solution space $\mathcal{S}$ of the Shift bound problem for the binary cyclic code of length 7 with defining set $\{1\}$. The complete defining set of this code is given by $Z(C) = \{1, 2, 4\}$, and the non-zeros set will be $N(C) = \mathbb{Z}_7 \setminus Z(C) = \{0, 3, 5, 6\}$.

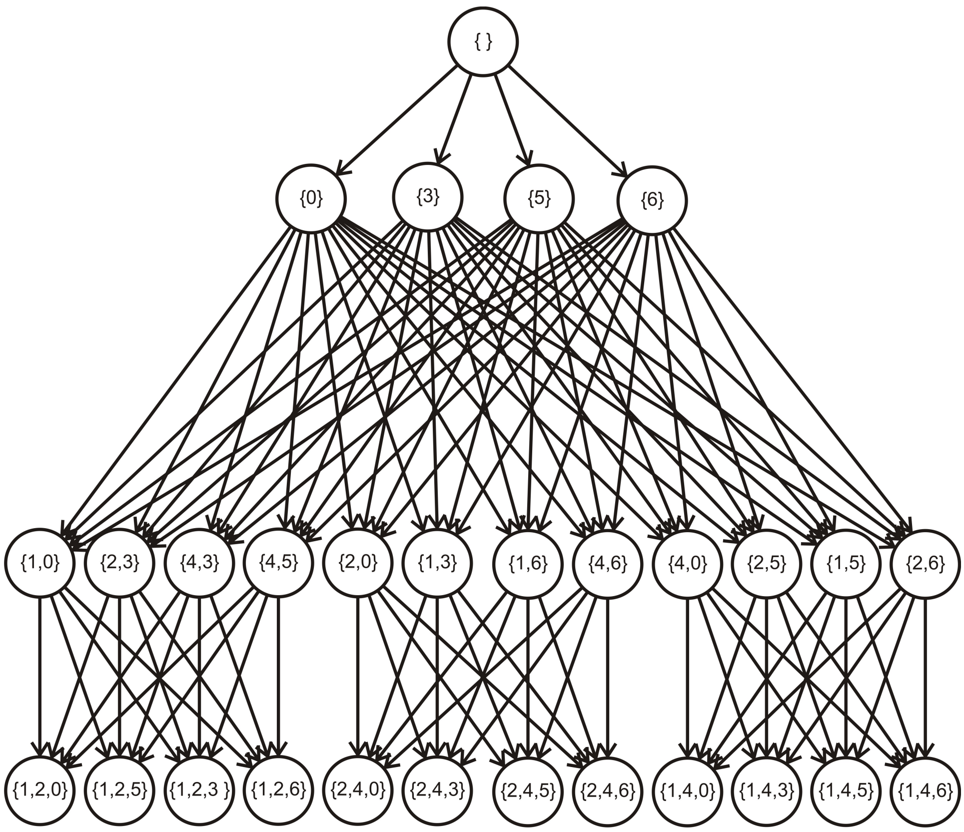


Figure 5.3: Directed acyclic graph of the Shift bound for Hamming code $[7,\ 4,\ 3]_2$.

5.2.3 Complexity

In this sub-section, we estimate the complexity of our implementation of a backtracking algorithm on the Shift bound problem. And to be able to tell about the complexity, we need to count the number of nodes in the solution space. Since it is hard in general case, we consider a special case.

Let C be a cyclic code of length n. And let the complete defining set of C be defined by

$$Z = \{0, 1, 2, \dots, \delta - 2\}.$$

Definition 5.26. Define *Lee* metric on $\mathbb{Z}_n$ by

$$d(i, j) = d_L(i, j) = \min\{|i - j|, n - |i - j|\},$$

where $i, j \in \mathbb{Z}_n$, $0 \leq i, j < n$.

Let $A \subseteq \mathbb{Z}_n$. We define the minimum distance as follows,

$$d(A) = d_L(A) = \min\{d(a, b) \mid a, b \in A, a \neq b\},$$

and the maximum distance as follows

$$\mu_d(A) = \max\{d(a, b) \mid a, b \in A\}.$$

Proposition 5.27. *Let $Z = \{0, 1, 2, \dots, \delta - 2\} \subseteq \mathbb{Z}_n$ and $A \subseteq \mathbb{Z}_n$, and $A \neq \emptyset$. Suppose $\delta \leq \frac{n}{2} + 2$.*

1. *A is independent set with respect to Z if and only if there is a $b \in A$ such that $\mu_d(A \setminus \{b\}) < \delta$.*
2. *$|S_A| = \delta - 1 - \mu_d(A)$.*

To prove Proposition 5.27, we need several definitions and lemmas.

Lemma 5.28. *If $A \subseteq B$, then $\mu_d(A) \leq \mu_d(B)$.*

Proof. $$\mu_d(A) = \max\{d(x, y) \mid x, y \in A\} \leq \max\{d(x, y) \mid x, y \in B\} = \mu_d(B),$$

since $A \subseteq B$. □

Lemma 5.29. $$\mu_d(Z) = \delta - 2.$$

Proof. By Definition 5.26,

$$\begin{aligned}
\mu_d(Z) &= \max\{d(a, b) \mid a, b \in Z, a \neq b\} \\
&= d(0, \delta - 2) \\
&= \min\{\delta - 2, n - \delta + 2\} \\
&= \delta - 2,
\end{aligned}$$

since $\delta \leq \frac{n}{2} + 2$ and $d(a, b) = \min\{b - a, n - b + a\} = b - a \leq \delta - 2$ for all $0 \leq a < b \leq \delta - 2$. □

Lemma 5.30. *$\mu_d(A) \leq \delta - 2$ if and only if $a + A \subseteq Z$ for some $a \in \mathbb{Z}_n$.*

Proof. We will prove this lemma in two directions;

$\Leftarrow$ This is a direct consequences of Lemma 5.28 and Lemma 5.29. Note that $\mu_d(a + A) = \mu_d(A)$.

$\Rightarrow$ Conversely, let $\mu_d(A) \leq \delta - 2$. Without loss of generality, assume that $A = \{a_1, a_2, \ldots, a_i\}$ with $1 = a_1 < a_2 < \ldots < a_{i-1} \leq \delta - 2 < a_i = \delta - 1$. Choose $a = -1 \in \mathbb{Z}_n$. Then after the a shift of A, $a + A = \{a_1, a_2, \ldots, a_i\}$ with $0 = a_1 < a_2 < \ldots < a_{i-1} \leq \delta - 3 < a_i = \delta - 2$, in which $a + A \subseteq Z$.

□

Lemma 5.31. *A is independent and is not maximal if and only if $\mu_d(A) \leq \delta - 2$.*

Proof. We will prove the lemma in two directions;

$\Leftarrow$ If $\mu_d(A) \leq \delta - 2$, then by Lemma 5.30 $a + A \subseteq Z$ for some $a \in \mathbb{Z}_n$. Apply Definition 5.14 point 2 to $a + A$. As a result, $\bar{A} = (a + A) \cup \{b\}$ for $b \notin Z$, which is independent set with respect to Z. Clearly that $|\bar{A}| \geq |A|$. Hence, A is not maximal.

$\Rightarrow$ Conversely, if A is independent set with respect to Z and is not maximal, then for some $a \in S_A$, $a + A \subseteq Z$. From Lemma 5.28 and 5.29, $\mu_d(a + A) \leq \delta - 2$.

□

Lemma 5.32. *A is independent and is maximal if and only if $\mu_d(A) > \delta - 2$ and $\mu_d(A \setminus \{b\}) \leq \delta - 2$ for some $b \in A$.*

Proof. We will prove the lemma in two directions;

$\Rightarrow$ If A is an independent set with respect to Z and is maximal, then there is no $a \in \mathbb{Z}_n$ such that $a + A \subseteq Z$. By Lemma 5.30, $\mu_d(A) > \delta - 2$. Since A is an independent set, hence $A = (a + \acute{A}) \cup \{b\}$ with $a + \acute{A}$ and $b \notin Z$. Hence $A \setminus \{b\} = a + \acute{A} \subseteq Z$. So, $\mu_d(A \setminus \{b\}) \leq \delta - 2$.

$\Leftarrow$ If $\mu_d(A \setminus \{b\}) \leq \delta - 2$, then by Lemma 5.28 and Lemma 5.29, $A \setminus \{b\} \subseteq Z$. By Definition 5.14, A is an independent set with respect to Z. Since $\mu_d(A) > \delta - 2$, then by Lemma 5.30, there is no $a \in \mathbb{Z}_n$ such that $a + A \subseteq Z$. So, A is maximal.

□

Lemma 5.33. *If $\mu_d(A) > \delta - 2$, then $S_A = \emptyset$.*

Proof. As a consequence of Lemma 5.32. If $\mu_d(A) > \delta - 2$, then A is an independent set and is maximal. If A is maximal, then $S_A = \emptyset$. □

Remark 5.34. If $\mu_d(A) \leq \delta - 2$, then after a shift of A, we may assume without loss of generality that $A = \{a_1, a_2, \ldots, a_i\}$ with $0 = a_1 < a_2 < \ldots < a_i \leq \delta - 2$. Then $\mu_d(A) = a_i$ and $S_A = \{0, 1, \ldots, \delta - 2 - a_i\}$. Hence $|S_A| = \delta - 1 - a_i = \delta - 1 - \mu_d(A)$.

Proof of Proposition 5.27. As a consequence of Lemma 5.32 and Remark 5.34. □

Now, we are going to estimate the number of nodes in the DAG for the backtracking algorithm on Shift bound problem. Recall the idea on constructing a sequence of independent sets. If A is an independent sets, then $\bar{A} = (a + A) \cup \{b\}$ where $a \in S_A$ and $b \notin Z$ is also an independent set. We are going to counting the number of nodes for the Shift bound on C with complete defining set $Z = \{0, 1, 2, \ldots, \delta - 2\}$.

We start with $A_0 = \emptyset$, the number of independent set of size 0 is 1. The number of independent set of size 1 is equal to the number of non-zeros of C times the number of ways to pick 0 element of Z or

$$(n - |Z|) \times \binom{\delta - 1}{0}.$$

The number of independent set of size 2 is equal to the number of non-zeros of C times the number of ways to pick 1 element of Z or

$$(n - |Z|) \times \binom{\delta - 1}{1}.$$

The number of independent set of size 3 is equal to the number of non-zeros of C times the number of ways to pick 2 elements of Z or

$$(n - |Z|) \times \binom{\delta - 1}{2}.$$

Without loss of generality, assume that $A = \{a_1, a_2, \ldots, a_i\}$ be an independent set with respect to Z of size i that is not maximal. As a consequence of Lemma 5.31, $0 = a_1 < a_2 < \ldots < a_i \leq \delta - 2$. And the number of independent set of size i that are leaves is equal to the number of non-zeros of C times the number of ways to pick $i - 1$ elements of Z or

$$(n - |Z|) \times \binom{\delta - 1}{i - 1}. \tag{5.1}$$

Without loss of generality, assume that $A = \{a_1, a_2, \ldots, a_i\}$ be an independent set with respect to Z of size i that is maximal. As a consequence of Lemma 5.32 and Lemma 5.33, $0 = a_1 < a_2 < \ldots < a_{i-1} \leq \delta - 2 < a_i < n + a_{i-1} - \delta - 2$. In this case, the number of independent sets is equal to

$$(n - |Z|) \times \binom{\delta - 1}{i - 1}.$$

So, let A_0, A_1, A_3, …, A_i be a sequence of independent sets and maximal, then the number of nodes in the tree equals to

$$1 + (n - |Z|) \times \sum_{j=1}^{i} \binom{\delta - 1}{j - 1}. \tag{5.2}$$

Note that,

$$\binom{\delta - 1}{i} \approx 2^{\lambda},$$

where $\lim_{n\to\infty} \frac{1}{n} \log_q V_q(n, \lfloor \lambda n \rfloor) = H_q(\lambda)$. See Section 2.3.

Example 5.35. Let C be a 5-ary cyclic code of length 4 with the complete defining set $Z = \{0, 1\}$. Hence, $\delta = 3$. Note that $N(C) = \mathbb{Z}_4 \setminus Z = \{2, 3\}$. The number of independent sets of size 1 is

$$|N(C)| \times \binom{\delta - 1}{0} = (4 - 2) \times \binom{2}{0} = 2.$$

The number of independent sets of size 2 is

$$|N(C)| \times \binom{\delta - 1}{1} = (4 - 2) \times \binom{2}{1} = 4.$$

And the number of independent sets of size 3, which in this case are maximal, is

$$|N(C)| \times \binom{\delta - 1}{2} = (4 - 2) \times \binom{2}{2} = 2.$$

So, indeed the total number of independent sets is

$$1 + |N(C)| \cdot \sum_{j=1}^{3} \binom{\delta - 1}{j - 1} = 1 + 4 \cdot (1 + 2 + 1) = 9.$$

See Figure 5.4 for the complete description of the solution space.

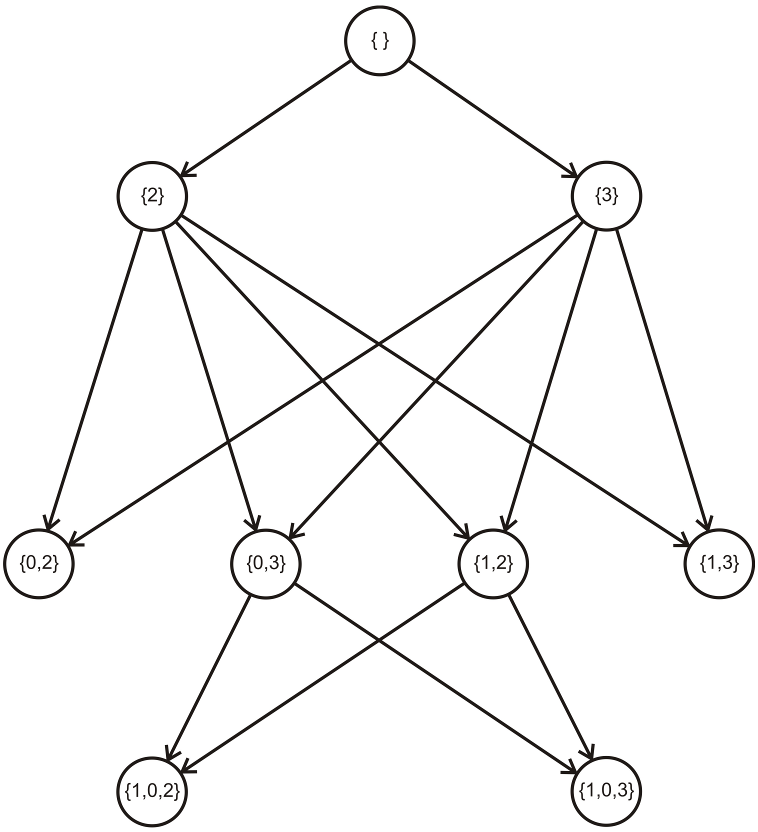


Figure 5.4: Directed graph of the Shift bound for a 5-ary cyclic code of length 4.

5.3. Improvements of the algorithm

5.3.1 Modification of the algorithm

The reason of using a backtracking algorithm for solving the Shift bound problem has been explained in the previous section. Now, we are going to explain the implementation of the backtracking algorithm for solving the Shift bound problem. We are going to explain about the modification that we made in order to implement the Shift bound problem into C++.

In the previous section, the successor of A, namely $\overline{A}$, was determined by the combination of $a \in S_A$ and $b \notin Z(C)$, which is quite a large number of successors. In this section, we reduce the number of unnecessary successors of A. The successor nodes for A are determined only by elements in the set C_A defined as follows;

Definition 5.36. Let define

$$C_A = N(C) - S_A = \{\, b - a \mid \text{ for all } b \in N(C), \text{ for all } a \in S_A\},$$

where $N(C)$ the set of non-zeros of C.

Edges will consist of the following properties; for each $c \in C_{A_i}$, and $a \in S_{A_i}$

$$\begin{aligned} \widetilde{A}_{i+1} &= A_i \cup \{c\} \\ S_{\widetilde{A}_{i+1}} &= S_{A_i} \cap S_c \end{aligned}$$

Table 5.3: *Arc properties.*

The following algorithm is our modified backtracking algorithm to find the maximum size of sequence of independent sets;

Algorithm 7 Backtracking algorithm for the modified Shift bound Part 1

procedure BACKTRACK($\mathbf{x} = (A_0, A_1, A_2, \ldots, A_{i-1}, A_i)$)
  $A = A_i$
  for $c \in C_A$ **do**
    $\widetilde{A} = A \cup \{c\}$
    $S_{\widetilde{A}} = S_A \cap S_c$
    $\overline{A} = \widetilde{A} - a$, for $a \in S_A$
    $\mathbf{x} = \mathbf{x} \cup \{\overline{A}\}$
    if $S_{\widetilde{A}} = \emptyset$ **then**
      if $|\mathbf{x}| > |\mathbf{x}_{max}|$ **then**
        $\mathbf{x}_{max} = \mathbf{x}$
      end if
      backtrack($\mathbf{x} = \mathbf{x} \setminus \overline{A}$)
    end if
    $S_{\overline{A}} = S_{\widetilde{A}} - a$

Algorithm 8 Backtracking algorithm for the modified Shift bound Part 2

$C_{\overline{A}} = \{\overline{b} - \overline{a} \,|\, \overline{a} \in S_{\overline{A}}, \overline{b} \notin Z(C)\}$
backtrack($\mathbf{x}$)
end for
end procedure

Example 5.37. Figure 5.5 illustrates the solution space $\mathcal{S}$ of the Shift bound problem for the binary cyclic code of length 7 with defining set $\{1\}$. The complete defining set of this code is given by $Z(C) = \{1, 2, 4\}$, and the non-zeros set will be $N(C) = \mathbb{Z}_7 \setminus Z(C) = \{0, 3, 5, 6\}$. But in this example we apply the Definition 5.36 to determine the successor of nodes.

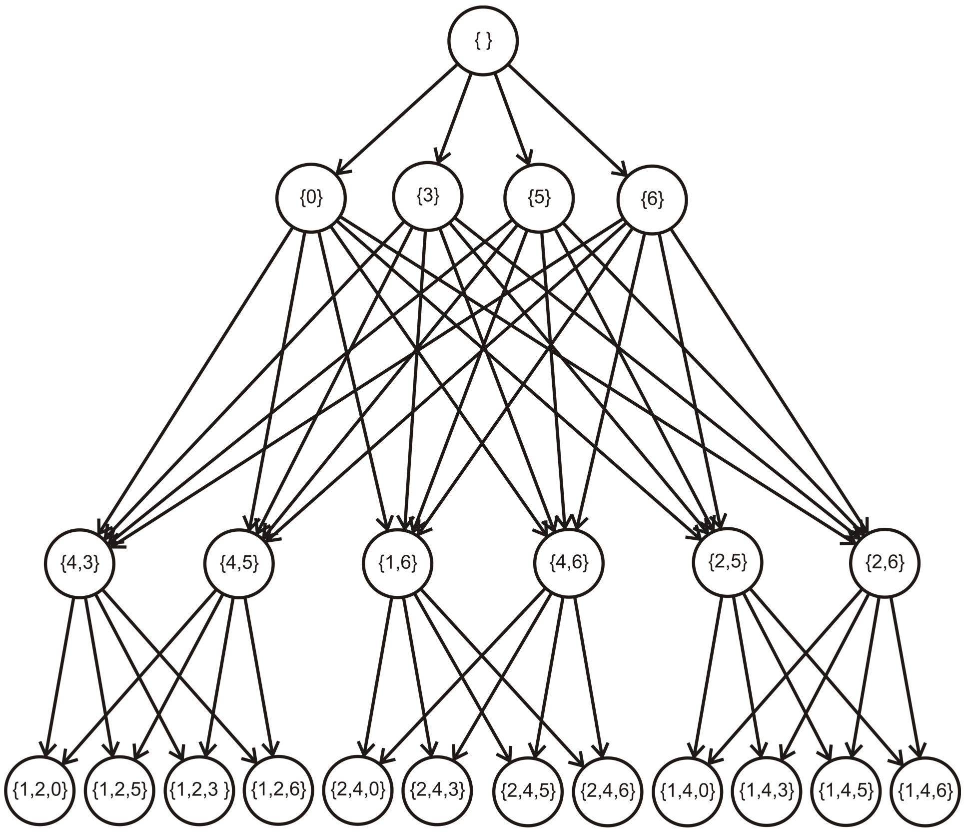


Figure 5.5: Directed graph of the modified Shift bound for Hamming code $[7, 4, 3]_2$.

5.3.2 Branch-And-Bound technique

While searching for the best solution, a backtracking algorithm visits all nodes in the solution space, i.e. it does the tree traversal. Sometimes we can determine that a given node in the solution space does not lead to the optimal solution, either because the given solution and all its successors are infeasible or because we have already found a solution that us guaranteed to be better than any successor of the given solution. In such cases, the given node and its successors need not be considered. In effect, we can *prune* the solution tree, hereby reducing the number of solutions to be considered, i.e reducing the number of nodes to be visited. A backtracking algorithm that prunes the search space is called a *branch-and-bound* algorithm.

Let A be an independent set with respect to $Z(C)$. Extend A by adding A with the element from $C_A = N(C) - S_A$. Hence $\tilde{A} = A \cup \{b - a\} = A \cup \{c\}$, where $c = b - a$ for all $b \in N(C)$ and $a \in S_A$. Since $a + \tilde{A} = (a + A) \cup \{b\}$, hence $a \notin S_{\tilde{A}}$. So,

$$|S_{\tilde{A}}| < |S_A|.$$

This means that A can be extended at most $|S_A|$ times.

If $S_A = \{a^{(1)}, a^{(2)}, \ldots, a^{(r)}\}$, then an independent set A can be extend at most

$$\hat{A} = A \cup \{b^{(1)} - a^{(1)}, b^{(2)} - a^{(2)}, \ldots, b^{(r)} - a^{(r)}\},$$

where $b^{(1)}$, $b^{(2)}$, $\ldots$, $b^{(r)} \in N(C)$.

So, let

$$m(A) = \max\{\,|\hat{A}\,|\,|\,\hat{A} \supseteq A, \hat{A} \text{ independent }\},$$

then

$$m(A) \leq |A| + |S_A|.$$

Let $\mathbf{x}$ be a vector of independent sets, where $\mathbf{x} = (A_0, A_1, \ldots, A_i)$, for $i > 0$. Let $\mathbf{x}_{max}$ be a vector of independent sets of maximum size. Note that, our working vector is $\mathbf{x}$. When backtracking algorithm gets to the independent set A, then compute $|A|$ and $|S_A|$, which yields $m(A)$. If $|\mathbf{x}_{max}| > m(A)$, then backtracking does not have to extend the sequence, because we will not find any sequence larger than $\mathbf{x}_{max}$. Instead, the algorithm should have backtrack, to find another alternative options.

5.3.3 Speeding-up the calculation process

Initially when our backtracking algorithm invoked, we start with an empty set. Let A_i be an independent set with respect to Z. Hence $A_{i+1} = (a_i + A_i) \cup \{b_i\}$, where $a_i \in S_{A_i}$ and $b_i \notin Z$ is also independent set with respect to Z. In the original algorithm, for all $a_i \in S_{A_i}$ and $b_i \notin Z$ are considered. In the modification algorithm, only $a_i \in S_{A_i}$ and $b_i \notin Z$ such that $b_i - a_i \in C_{A_i}$ are considered.

Without loss of generality, we may assume that $A_0 = \emptyset$ and choose $b_0 \notin Z(C)$ such that $A_1 = \{b_0\}$. For instance, in Example 5.37 we may choose $A_1 = \{0\}$.

Now, we introduce a new selection function, such that we do not have to consider all a_is and b_is in both algorithms. This new selection function is part of the so called *Greedy* algorithm.

The function is defined as follows; choose $a_i \in S_{A_i}$ and $b_i \notin Z(C)$ such that $\max\{|S_{A_{i+1}}|\}$, where $S_{A_{i+1}} = (S_{A_i} - a_i) \cap S_{\{b_i\}}$.

Example 5.38. Recall Example 5.35. If we apply the Greedy algorithm, then nodes that contain independent sets $\{0, 2\}$ and $\{1, 3\}$ will not be considered.

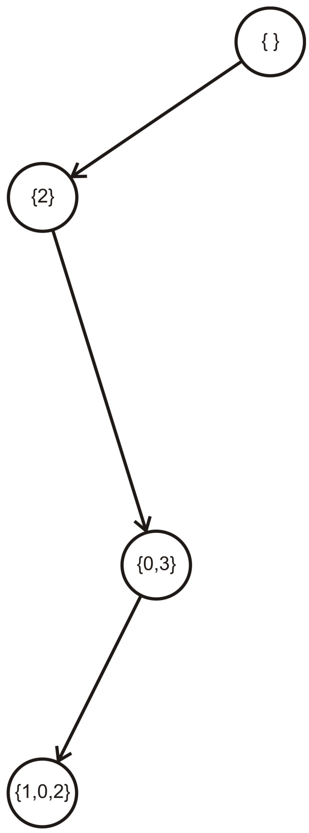


Figure 5.6: Directed graph of the 5-ary cyclic code of length 4 after Greedy algorithm implemented on the original algorithm.

Example 5.39. Let C be a ternary cyclic code of length 45 with defining set $\{1, 3, 5, 15\}$. In this example, we will show that our algorithm encounter a case where one really has to backtrack to find the optimal solution. The complete defining set of C is

$$Z(C) = \{1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 17, 19, 20, 23, 24, 25, 30, 31, 32, 34, 35, 38, 40\}.$$

Construct sequences of independent sets with respect to $Z(C)$ as follows; for each $i > 0$, $a_i + A_i \subseteq$

$Z(C)$, $b_i \notin Z(C)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$,

$$\begin{array}{lclcl}
 & & & & A_0 = \emptyset \\
a_0 = 0 & , & b_0 = 0 & \longrightarrow & A_1 = \{0\} \\
a_1 = 15 & , & b_1 = 0 & \longrightarrow & A_2 = \{15, 0\} \\
a_2 = 15 & , & b_2 = 0 & \longrightarrow & A_3 = \{30, 15, 0\} \\
a_3 = 1 & , & b_3 = 0 & \longrightarrow & A_4 = \{31, 16, 1, 0\} \\
a_4 = 3 & , & b_4 = 7 & \longrightarrow & A_5 = \{34, 19, 4, 3, 7\} \\
a_5 = 12 & , & b_5 = 0 & \longrightarrow & A_6 = \{1, 31, 16, 15, 19, 0\} \\
a_6 = 15 & , & b_6 = 0 & \longrightarrow & A_7 = \{16, 1, 31, 30, 34, 15, 0\} \\
a_7 = 1 & , & b_7 = 0 & \longrightarrow & A_8 = \{17, 2, 32, 31, 35, 16, 1, 0\} \\
a_8 = 3 & , & b_7 = 7 & \longrightarrow & A_9 = \{20, 5, 35, 34, 38, 19, 4, 3, 7\} \\
a_9 = 12 & , & b_7 = 18 & \longrightarrow & A_{10} = \{32, 17, 2, 1, 5, 31, 16, 15, 19, 18\}
\end{array}$$

So $n(Z(C)) = 10$. By backtrack to A_9, the algorithm then choose $a_9 = 27$ and $b_9 = 0$, which leads to larger sequence of independent sets.

$$\begin{array}{lclcl}
 & & & & A_0 = \emptyset \\
a_0 = 0 & , & b_0 = 0 & \longrightarrow & A_1 = \{0\} \\
a_1 = 15 & , & b_1 = 0 & \longrightarrow & A_2 = \{15, 0\} \\
a_2 = 15 & , & b_2 = 0 & \longrightarrow & A_3 = \{30, 15, 0\} \\
a_3 = 1 & , & b_3 = 0 & \longrightarrow & A_4 = \{31, 16, 1, 0\} \\
a_4 = 3 & , & b_4 = 7 & \longrightarrow & A_5 = \{34, 19, 4, 3, 7\} \\
a_5 = 12 & , & b_5 = 0 & \longrightarrow & A_6 = \{1, 31, 16, 15, 19, 0\} \\
a_6 = 15 & , & b_6 = 0 & \longrightarrow & A_7 = \{16, 1, 31, 30, 34, 15, 0\} \\
a_7 = 1 & , & b_7 = 0 & \longrightarrow & A_8 = \{17, 2, 32, 31, 35, 16, 1, 0\} \\
a_8 = 3 & , & b_7 = 7 & \longrightarrow & A_9 = \{20, 5, 35, 34, 38, 19, 4, 3, 7\} \\
a_9 = 27 & , & b_7 = 0 & \longrightarrow & A_{10} = \{2, 32, 17, 16, 20, 1, 31, 30, 34, 0\} \\
a_{10} = 30 & , & b_7 = 11 & \longrightarrow & A_{10} = \{32, 17, 2, 1, 5, 31, 16, 15, 19, 30, 11\}
\end{array}$$

We get $n(Z(C)) = 11$.

6

The Quadratic Residue Codes

In this chapter, we will discuss one type of cyclic code for which the word length n is an odd prime, and the field $\mathbb{F}_q$ satisfy condition such that q be a quadratic residue (mod n), i.e. $q^{\frac{n-1}{2}} \equiv 1 \bmod n$. We consider the Square Root bound of the quadratic residue codes and we compare the Square Root bound with the BCH, HT, HT-Roos, Roos and the Shift bounds. Let α be denote a primitive n-th root of unity in an extension field of $\mathbb{F}_q$.

6.1. Definition

We are going to define the quadratic-residue (QR) codes of prime length n over $\mathbb{F}_q$, where q is prime power which is a quadratic-residue modulo n.

Definition 6.1. Let R_0 be the set of the quadratic residues in $\mathbb{F}_n$, that is

$$R_0 = \{i^2 \mid i \in \mathbb{F}_n, i \neq 0\},$$

and let R_1 be the set of non-squares in $\mathbb{F}_n$, such that

$$R_1 = \mathbb{F}_n^* \setminus R_0.$$

Clearly, if i is a primitive element of the field $\mathbb{F}_n$, then $i^e \in R_0$ if and only if e is even, while $i^e \in R_1$ if and only if e is odd. Thus R_0 is a cyclic group generated by i^2. Since $q \in R_0$, the set R_0 is closed under multiplication by q. Thus R_0 is a disjoint union of cyclotomic cosets modulo n. Hence

$$g_0(x) = \prod_{r \in R_0} (x - \alpha^r),$$

and

$$g_1(x) = \prod_{r \in R_1} (x - \alpha^r).$$

have coefficients from $\mathbb{F}_q$, where α is a primitive n-th root of unity in some field containing $\mathbb{F}_q$.

Since $q \bmod n$ is in R_0, the polynomials $g_0(x)$ and $g_1(x)$ have coefficients in $\mathbb{F}_q$. Furthermore

$$x^n - 1 = (x-1)g_0(x)g_1(x).$$

Let $\mathbb{C}_{q,n}$ be the ring $\mathbb{F}_q[x]/(x^n - 1)$.

Definition 6.2. The *quadratic-residue* codes $\mathfrak{Q}$, $\overline{\mathfrak{Q}}$, $\mathfrak{N}$, $\overline{\mathfrak{N}}$ are cyclic codes of ring $\mathbb{C}_{q,n}$ with generator polynomials

$$g_0(x), \quad (x-1)g_0(x), \quad g_1(x), \quad (x-1)g_1(x)$$

respectively.

Clearly $\mathfrak{Q} \supset \overline{\mathfrak{Q}}$ and $\mathfrak{N} \supset \overline{\mathfrak{N}}$. In the binary case, $\overline{\mathfrak{Q}}$ is the even weight subcode of $\mathfrak{Q}$, and $\overline{\mathfrak{N}}$ is the even weight subcode of $\mathfrak{N}$.

The permutation $\pi_j : i \mapsto ij \bmod n$ acting on the position of the codewords maps the code generator $g_0(x)$ into itself if $j \in R_0$ resp. into the code generator $g_1(x)$ if $j \in R_1$. So the codes with generators $g_0(x)$ resp. $g_1(x)$ are equivalent. If $n \equiv -1 \bmod 4$, then $-1 \in R_1$ and in that case the transformation $x \to x^{-1}$ maps a codeword of the code with generator $g_0(x)$ into a codeword of the code with generator $g_1(x)$.

$\mathfrak{Q}$ and $\mathfrak{N}$ have dimension $\frac{1}{2}(n+1)$, and $\overline{\mathfrak{Q}}$ and $\overline{\mathfrak{N}}$ have dimension $\frac{1}{2}(n-1)$.

6.2. The Square Root (SQRT) bound on the minimum distance

Theorem 6.3. *If $\mathbf{c} = c(x)$ is a codeword in the quadratic-residue (QR) code with generator $g_0(x)$ and if $c(1) \neq 0$ and $\mathrm{wt}(\mathbf{c}) = \delta$, then the following hold.*

1. *$\delta^2 \geq n$,*
2. *if $n = 4k-1$, then 1 can be strengthened to $\delta^2 - \delta + 1 \geq n$,*
3. *if $n = 8k-1$ and $q = 2$, then $\delta \equiv 3 \bmod 4$.*

Proof. Since $c(1) \neq 0$, the polynomial $c(x)$ is not divisible by $(x-1)$. By a suitable permutation π_j we can transform $c(x)$ into a polynomial $\overline{c}(x)$ which is divisible by $g_1(x)$ and of course again not divisible by $(x-1)$.

1. Let $c(x)$ be a codeword of minimum nonzero weight δ in $\mathfrak{Q}$. If r_1 is a non-residue, $\overline{c}(x) = c(x^{r_1})$ is a codeword of minimum weight in $\mathfrak{N}$. Then $c(x)\overline{c}(x)$ must be in $\mathfrak{Q} \cap \mathfrak{N}$, i.e. is a multiple of

   $$\prod_{r_0 \in R_0} (x - \alpha^{r_0}) \prod_{r_1 \in R_1} (x - \alpha^{r_1}) = \prod_{j=1}^{n-1} (x - \alpha^j) = \sum_{j=0}^{n-1} x^j.$$

   Thus $c(x)\overline{c}(x)$ has weight n. Since $c(x)$ has weight δ, the maximum number of nonzero coefficients in $c(x)\overline{c}(x)$ is δ^2, so that $\delta^2 \geq n$.

2. If $n = 4k-1$, we may take $r_1 = -1$. Now in the product $c(x)c(x^{-1})$ there are δ terms equal to 1, so the maximum weight of the product is $\delta^2 - \delta + 1$.

3. Let

$$\begin{aligned} c(x) &= \sum_{i=1}^{d} x^{l_i}, \\ \bar{c}(x) &= \sum_{i=1}^{d} x^{-l_i}. \end{aligned}$$

If $l_i - l_j = l_k - l_l$, then $l_j - l_i = l_l - l_k$. Hence, if terms in the product $c(x)\bar{c}(x)$ cancel, then they cancel four at time. Therefore, $n = \delta^2 - \delta + 1 - 4 \cdot a$ for some $a \geq 0$.

□

To say something about the minimum distance of QR codes, we need a powerful tool such as the *idempotent* of a cyclic code and the analysis on the automorphism group of the extended binary QR codes.

Theorem 6.4. *Let C be a binary QR code of length n and let choose α be a primitive n-th root of unity so that the idempotent of $\mathfrak{Q}$, $\bar{\mathfrak{Q}}$, $\mathfrak{N}$, $\bar{\mathfrak{N}}$ are*

$$\begin{aligned} \theta(x) &= \sum_{r \in R_0} x^r, \\ \vartheta(x) &= 1 + \sum_{r \in R_1} x^r, \\ \bar{\theta}(x) &= \sum_{r \in R_1} x^r, \\ \bar{\vartheta}(x) &= 1 + \sum_{r \in R_0} x^r, \end{aligned}$$

respectively.

Proof. Since 2 is a quadratic residue modulo n, hence

$$\begin{aligned} (\theta(x))^2 &= \theta(x), \\ (\vartheta(x))^2 &= \vartheta(x), \\ (\bar{\theta}(x))^2 &= \bar{\theta}(x), \\ (\bar{\vartheta}(x))^2 &= \bar{\vartheta}(x), \end{aligned}$$

so these polynomials are idempotent. Thus $\theta(\alpha^i) = 0$ or 1 by Lemma 3.34. For any quadratic residue s,

$$\theta(\alpha^s) = \sum_{r \in R_0} \alpha^{rs} = \sum_{r_1 \in R_0} \alpha^{r_1} = \theta(\alpha),$$

independent of s. Similarly,

$$\theta(\alpha^t) = \sum_{r \in R_0} \alpha^{rt} = \sum_{r \in R_0} \alpha^{-r} = \theta(\alpha^{-1}),$$

for any non-residue t. Since $\theta(\alpha) + \theta(\alpha^{-1}) = 1$, either

$$\theta(\alpha^s) = 0 \text{ for all } s \in R_0 \text{ and } \theta(\alpha^t) = 1 \text{ for all } t \in R_1, \tag{6.1}$$

or

$$\theta(\alpha^s) = 1 \text{ for all } s \in R_0 \text{ and } \theta(\alpha^t) = 0 \text{ for all } t \in R_1$$

depending on the choice of α. Choose α such that 6.1 holds. Then $\theta(x)$ is the idempotent of $\mathfrak{Q}$. Also

$$\bar{\theta}(\alpha^t) = \sum_{r\in R_1} \alpha^{rt} = \sum_{r\in R_1} \alpha^r = 0 \text{ for } t \in R_1,$$

and $\bar{\theta}(\alpha^s) = 1$ for $s \in R_0$. Thus $\bar{\theta}(x)$ is the idempotent of $\mathfrak{N}$. Finally, $\vartheta(\alpha^s) = 0$ for $s \in R_0$ and $\vartheta(1) = 0$, so $\vartheta(x)$ is the idempotent of $\bar{\mathfrak{Q}}$. Similarly for $\bar{\mathfrak{N}}$. □

With the help of the idempotent, we are going to show that the extended QR code $\hat{\mathfrak{Q}}$ is fixed under the large permutation group PSL(2,n). First we need to determine the dual of the QR codes.

Theorem 6.5. *We have the following;*

$$\mathfrak{Q}^\perp = \bar{\mathfrak{Q}}, \quad \mathfrak{N}^\perp = \bar{\mathfrak{N}}, \ \text{if } n = 4k-1, \tag{6.2}$$

$$\mathfrak{Q}^\perp = \bar{\mathfrak{N}}, \quad \mathfrak{N}^\perp = \bar{\mathfrak{Q}}, \ \text{if } n = 4k+1. \tag{6.3}$$

In both cases,

$$\mathfrak{Q} \text{ is generated by } \bar{\mathfrak{Q}} \text{ and } \mathbf{1}, \tag{6.4}$$

$$\mathfrak{N} \text{ is generated by } \bar{\mathfrak{N}} \text{ and } \mathbf{1}. \tag{6.5}$$

Proof. Suppose $n = 4k-1$. The zeros of $\mathfrak{Q}$ are α^r for $r \in R_0$. Hence by Theorem 3.30, the zeros of $\mathfrak{Q}^\perp$ are 1 and α^{-r} for $r \in R_1$. But $-r \in R_0$, so $\mathfrak{Q}^\perp = \bar{\mathfrak{Q}}$. From Theorem 6.4,

$$\theta(x) = \vartheta(x) + \frac{1}{n}\sum_{i=0}^{n-1} x^i,$$

which implies 6.4. Similarly for the case $n = 4k+1$. □

Let

$$\vartheta(x) = \sum_{i=0}^{n-1} f_i x^i,$$

be the idempotent of $\bar{\mathfrak{Q}}$, given by Theorem 6.4. Then a generator matrix for $\bar{\mathfrak{Q}}$ is the $n \times n$ *circulant* matrix

$$\bar{G} = \begin{pmatrix} f_0 & f_1 & \cdots & f_{n-1} \\ f_{n-1} & f_0 & \cdots & f_{n-2} \\ \cdot & \cdot & \cdot & \cdot \\ f_1 & f_2 & \cdots & f_0 \end{pmatrix} \tag{6.6}$$

$$= (g_{ij}), 0 \le i,j \le n-1, \text{ with } g_{ij} = f_{j-i}, \tag{6.7}$$

and with subscripts taken modulo n. A generator matrix for $\mathfrak{Q}$ is

$$G = \begin{pmatrix} & & \bar{G} & & \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \tag{6.8}$$

and similarly for $\mathfrak{N}$ and $\bar{\mathfrak{N}}$. Note that $\bar{G}$ has rank $\frac{1}{2}(n-1)$.

QR codes can be extended by adding overall parity check in such a way that,

$$\begin{array}{ll} (\hat{\mathfrak{Q}})^{\perp} = \hat{\mathfrak{Q}}, (\hat{\mathfrak{N}})^{\perp} = \hat{\mathfrak{N}}, & \text{if n = 4k - 1;} \\ (\hat{\mathfrak{Q}})^{\perp} = \hat{\mathfrak{N}}, & \text{if n = 4k + 1.} \end{array} \tag{6.9}$$

If $\mathbf{a} = (a_0, a_1, \dots, a_{n-1})$ is a codeword of $\mathfrak{Q}$ (or $\mathfrak{N}$) and $n = 4k - 1$, then the extended code is formed by appending

$$a_{\infty} = -y \sum_{i=0}^{n-1} a_i,$$

where $\mathbf{1} + y^2 n = 0$. Since $(yn)^2 = -n = \omega^2$, it follows that $y = \frac{\epsilon\omega}{n}$, where $\epsilon = \pm 1$ and ω as defined in Remark 6.9. Note that y is chosen such that the codeword $(1, 1, \dots, 1, -yn)$ of $\hat{\mathfrak{Q}}$ (or $\hat{\mathfrak{N}}$) is orthogonal to itself. Thus the generator matrix of $\hat{\mathfrak{Q}}$ is obtained by adding column to 6.8, and is given by

$$\hat{G} = \begin{pmatrix} & & \bar{G} & & \mathbf{c}^t \\ 1 & 1 & 1 & \dots & -yn \end{pmatrix},$$

with $\mathbf{c} = (0, 0, \dots, 0)$.

It follows from Theorem 6.4 that the rows of $\hat{G}$ generate the extended binary QR code, $\hat{\mathfrak{Q}}$, of length $n+1$. Let us number the coordinate places of codewords in $\hat{\mathfrak{Q}}$ with points of the projective line of order n, i.e. ∞, 0, 1, 2, ..., $n-1$. The overall parity check is in front and it has number ∞.

Definition 6.6. Let n be a prime of the form $8m \pm 1$. The set of all permutations of $\{0, 1, 2, \dots, n-1, \infty\}$ of the form

$$x \to \frac{ax+b}{cx+d},$$

with $a, b, c, d \in \mathbb{F}_n$ and $ad - bc = 1$, forms a group called the *projective special linear* group PSL(2,n).

Theorem 6.7. *PSL(2,n) has the following properties;*

1. PSL*(2,n) is generated by three permutations*

$$\begin{aligned} S &: x \to x+1 \\ T &: x \to -\frac{1}{x} \\ V &: x \to \rho^2 x \end{aligned}$$

with ρ is a primitive element of $\mathbb{F}_n$.

2. PSL(2,n) *consists of* $\frac{1}{2}n(n^2-1)$ *permutations*

$$\begin{array}{rcl} V^iS^j & : & x \to \rho^{2i}x + j \\ V^iS^jTS^k & : & x \to k - (\rho^{2i}x + j)^{-1} \end{array}$$

with $0 \le i < \frac{1}{2}(n-1)$, $0 \le j, k < n$.

3. *If* $n \equiv -1 \bmod 8$, *the generators* S, V, T *satisfy*

$$S^n = V^{\frac{1}{2}(n-1)} = T^2 = (VT)^2 = (ST)^3 = 1,$$

and

$$V^{-1}SV = S^{\rho^2}.$$

4. PSL(2,n) *is* doubly transitive.

Clearly S is a cyclic shift on the positions different from ∞ and it leaves ∞ invariant. By the definition of a QR code, S leaves the extended code invariant. It remains to show that $\hat{\mathfrak{D}}$ is also fixed by T. Only the case $n = 8m - 1$ is treated. To show that T is fixed the $\hat{\mathfrak{D}}$, we need to show that each row of $\hat{G}$ is transformed by T into another codeword of $\hat{\mathfrak{D}}$ and T maps a row of $\hat{G}$ into a linear combination of at most three rows of $\hat{G}$.

1. Consider the first row of $\hat{G}$ as follows,

$$\text{row}(0) = |1 + \sum_{r \in R_1} x^r|0|.$$

Then

$$T(\text{row}(0)) = |\sum_{r \in R_0} x^r|1| = \text{row}(0) + \mathbf{1},$$

which is in $\hat{\mathfrak{D}}$.

2. Suppose $s \in R_0$, and the $s+1$-th row of $\hat{G}$ is

$$\text{row}(s) = |x^s + \sum_{r \in R_1} x^{r+s}|0|.$$

We shall show that $T(\text{row}(s)) = \text{row}(-\frac{1}{s}) + \text{row}(0) + \mathbf{1}$ is in $\hat{\mathfrak{D}}$. $T(\text{row}(s))$ has 1's in coordinate places $-\frac{1}{s}$ and $-\frac{1}{r+s}$ for $r \in R_1$, which comprise ∞ if $r = -s$, $2m-1$ residues and $2m$ non-residues. Also

$$\text{row}(-\frac{1}{s}) = |x^{-\frac{1}{s}} + \sum_{r \in R_1} x^{n-\frac{1}{s}}|0|$$

has 1's in places $-\frac{1}{s}$ and $r - \frac{1}{s}$ for $r \in R_1$, which comprise $2m$ residues and $2m$ non-residues. Therefore the sum $T(\text{row}(s)) + \text{row}(-\frac{1}{s})$ has a 1 in place ∞ and a 0 in place $-\frac{1}{s}$. If $-\frac{1}{r+s} \in R_1$, then $-\frac{1}{r+s} = r' - \frac{1}{s}$ for some $r' \in R_1$, and the 1's in the sum cancel. Thus the non-residue coordinate places in the sum always contain 0. On the other hand, if

$-\frac{1}{r+s} \in R_0$, then $-\frac{1}{r+s} \neq r' - \frac{1}{s}$ for all $r' \in R_1$, and so the sum contains 1 in coordinate places which are residues. So,

$$T(\text{row}(s)) + \text{row}(-\frac{1}{s}) = |\sum_{r \in R_0} x^r|1| = row(0) + \mathbf{1}.$$

Similarly if $t \in R_1$,

$$T(\text{row}(t)) = \text{row}(-\frac{1}{t}) + \text{row}(0).$$

These show that T maps a row of $\hat{G}$ into a linear combination of at most three rows of $\hat{G}$. Therefore S and T leave the extended QR code invariant, which proving the following theorem.

Theorem 6.8 (Gleason and Prange). *The automorphism group of the extended binary QR code of length $n+1$ contains PSL(2,n).*

Remark 6.9. The modified definition of extended code ensures that Theorem 6.8 is also true for the non-binary case. For the non-binary case, the idempotent will be define as the *Gaussian sum*

$$\omega = \sum_{i=1}^{n-1} \chi(i)\alpha^i,$$

where the Legendre symbol $\chi(i)$ is defined by,

$$\chi(i) = \begin{cases} 0, & \text{if } i \text{ is a multiple of } n \text{ ;} \\ 1, & \text{if } i \text{ is a quadratic residue mod } n \text{ ;} \\ -1, & \text{if } i \text{ is a non-residue mod } n. \end{cases}$$

Also $\chi(i)\chi(j) = \chi(ij)$. Note also, $\omega^q = \omega$, $\omega \in \mathbb{F}_q$. If $n = 4k-1$, then $\omega^2 = -n$. For additional information, see [11].

Remark 6.10. A group $\mathcal{G}$ of of permutations of the symbols $\{1, 2, \ldots, n\}$ is *transitive* if for any symbols i, j there is a permutation $\phi \in \mathcal{G}$ such that $i\phi = j$. More generally, $\mathcal{G}$ is *t-fold transitive* if given t distinct symbols $i_1, i_2, \ldots, i_t$, and t distinct symbols $j_1, j_2, \ldots, j_t$, there is a $\phi \in \mathcal{G}$ such that $i_1\phi = j_1, \ldots, i_t\phi = j_t$.

Corollary 6.11. *If $\bar{C}$ is fixed by a transitive permutation group, then*

1. *deleting any coordinate place gives an equivalent code C,*
2. *and if all weight in $\bar{C}$ are even, then C has odd minimum weight.*

Corollary 6.12. *A word of minimum weight in a binary QR code satisfies the conditions 1, 2, and 3 of Theorem 6.3.*

Proof. Use the fact that PSL(2,n) is transitive. And as immediate consequence of Remark 6.10 and Corollary 6.11. An equivalent code $\mathfrak{D}$ is obtained no matter which coordinate place of $\hat{\mathfrak{D}}$ is deleted. □

Definition 6.13. Let d_{QR} be the minimum positive integer d that satisfies Theorem 6.3 point 1, 2, and 3.

Theorem 6.14 (The Square Root (SQRT) bound). *Let C be the QR code of length n. Then the minimum distance of C is at least d_{QR}.*

Proof. For the binary case, this theorem is an immediate consequences of Corollary 6.12 and Definition 6.13. For the non-binary case, we should consider Remark 6.9. □

6.3. Minimum distance of Quadratic Residue codes

6.3.1 Examples

In this section, we will analyze the minimum distance of the Quadratic Residue (QR) codes based on Theorem 6.3 and compare the result with others bounds, i.e. the BCH bound, the HT bound, the HT-Roos bound, Roos bound and especially with the Shift bound.

Example 6.15. Let C be the binary cyclic code of length $n = 7$ with defining set $\{1\}$. This code satisfy $2^{\frac{7-1}{2}} \equiv 1 \bmod 7$. Thus q is a quadratic residue mod 7. The complete defining set

$$Z(C) = \{1, 2, 4\} = \{i^2 \bmod 7 \,|\, i \in \mathbb{Z}_7, i \neq 0\}$$

is the quadratic residue in $\mathbb{Z}_7$, and the non-zeros set

$$N(C) = \mathbb{Z}_7^* \setminus Z(C) = \{3, 5, 6\}$$

is the set of non-squares in $\mathbb{Z}_7$.

Let α be the primitive 7-th root of unity. We find that

$$x^7 - 1 = (x-1)g_0(x)g_1(x),$$

where $g_0(x) = \prod_{r \in Z(C)}(x - \alpha^r)$ and $g_1(x) = \prod_{r \in N(C)}(x - \alpha^r)$.

Since $7 \equiv 3 \bmod 4$ and $7 \equiv 7 \bmod 8$, hence by Theorem 6.3, the minimum weight of a codeword must satisfy $d_{QR}^2 - d_{QR} + 1 \geq 7$ and $d_{QR} \equiv 3 \bmod 4$. This yield, $d_{QR} \geq 3$.

There is one consecutive set in the complete defining set, namely $\{1, 2\}$. Hence, by the BCH bound, the minimum distance of C is $d_BCH \geq 3$. The minimum distance of C based on the HT bound is $d_{HT} \geq 3$, with $b = 1$, $a_1 = 1$, $a_2 = 2$, $s = 0$, and $\delta = 3$.

Take $A = \{1\}$ and let $a_2 = 3$. Take $B = \{a2 \cdot j \,|\, j = 0, 1\}$. Hence, by the Roos bound, the minimum distance of C is $d_{Roos} \geq 3$.

Next we need to construct a sequence of independent sets, in order to determine the lower bound on the minimum distance of C using the Shift bound. For each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(C)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,0\}} \{0\} \xrightarrow{\{1,0\}} \{1,0\} \xrightarrow{\{1,0\}} \{2,1,0\}.$$

So, we get $n(Z(C)) = 3$, where $n(Z(C))$ is the maximum size of independent sets with respect to $Z(C)$.

Let C_0 be the subcode of C by adding 0 into $Z(C)$ or $Z(C_0) = Z(C) \cup \{0\}$. Again, we construct the sequence of independent sets as follows; for each $i > 0$, $a_i + A_i \subseteq Z(C_0)$, $b_i \notin Z(C_0)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,3\}} \{3\} \xrightarrow{\{1,3\}} \{4,3\} \xrightarrow{\{4,3\}} \{1,0,3\} \xrightarrow{\{1,3\}} \{2,1,4,3\}.$$

So, we get $n(Z(C_0)) = 4$. By Definition 5.5,

$$d_{\text{shift}}(Z(C)) = \min\{n(Z(C)), n(Z(C_0))\} = \min\{3, 4\} = 3.$$

and by Theorem 5.6, the minimum distance is $d_{\text{shift}} \geq 3$. In fact, this is the perfect binary Hamming code with parameters [7, 4, 3].

Example 6.16. Let C be the binary cyclic code of length $n = 23$ with defining set $\{1\}$. Since $2^{\frac{23-1}{2}} \equiv 1 \bmod 23$, hence q is quadratic residue mod 23. The complete defining set of C

$$Z(C) = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\} = \{i^2 \bmod 23 \,|\, i \in \mathbb{Z}_{23}, i \neq 0\},$$

is the quadratic residues in $\mathbb{Z}_{23}$, and the non-zeros set

$$N(C) = \mathbb{Z}_{23}^* \setminus Z(C) = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\},$$

is the set of non-squares in $\mathbb{Z}_{23}$.

Let α be the primitive 23-th root of unity. We find that

$$x^{23} - 1 = (x-1)g_0(x)g_1(x),$$

where $g_0(x) = \prod_{r \in Z(C)}(x - \alpha^r)$ and $g_1(x) = \prod_{r \in N(C)}(x - \alpha^r)$. By Theorem 6.3, the corresponding QR code C has minimum distance $d_{QR} \geq 7$. By the BCH bound, it has minimum distance $d_{BCH} \geq 5$. From the Example 5.7, the minimum distance of this code based on the Shift bound is $d_{shift} \geq 6$.

Since $\sum_{i=0}^{3} \binom{23}{i} = 2^{11}$ and $|C| = 2^{12}$, it follows that d is equal to 7. In fact, the corresponding QR code is a perfect cyclic code called *the binary Golay code.*

Example 6.17. Let C be the binary cyclic code of length $n = 31$ with defining set $\{1, 5, 7\}$. Since $2^{\frac{31-1}{2}} \equiv 1 \bmod 31$, hence q is quadratic residue mod 31. The complete defining set of C

$$Z(C) = \{1, 2, 4, 5, 7, 8, 9, 10, 14, 16, 18, 19, 20, 25, 28\} = \{i^2 \bmod 31 \,|\, i \in \mathbb{Z}_{31}, i \neq 0\},$$

is the quadratic residues in $\mathbb{Z}_{31}$, and the non-zeros set

$$N(C) = \mathbb{Z}_{31}^* \setminus Z(C) = \{3, 6, 12, 17, 24, 11, 13, 21, 22, 26, 15, 23, 27, 29, 30\},$$

is the set of non-squares in $\mathbb{Z}_{31}$.

Let α be the primitive 31-th root of unity. We find that

$$x^{31} - 1 = (x-1)g_0(x)g_1(x),$$

where $g_0(x) = \prod_{r \in Z(C)}(x - \alpha^r)$ and $g_1(x) = \prod_{r \in N(C)}(x - \alpha^r)$. And the codes generated by $g_0(x)$ resp. $g_1(x)$ are equivalent.

Since $31 \equiv 3 \bmod 4$ and $31 \equiv 7 \bmod 8$, hence by Theorem 6.3, the minimum weight of a codeword must satisfy $d_{QR}^2 - d_{QR} + 1 \geq 7$ and $d_{QR} \equiv 3 \bmod 4$. This yield, $d_{QR} \geq 7$.

There is one consecutive set in $Z(C)$, namely $\{7, 8, 9, 10\}$. By the BCH bound, the minimum distance of C is $d_{BCH} \geq 5$. The minimum distance of C based on the HT bound is $d_{HT} \geq 5$ with $b = 18$, $a_1 = 1$, $a_2 = 21$, $s = 1$, and $\delta = 4$. Take $A = \{8, 9, 10\}$, and $B = \{10 \cdot j \,|\, j = 0, 1, 3\}$, hence by the Roos bound, the minimum distance of C is $d_{Roos} \geq 7$.

Next, we need to construct a sequence of independent sets, in order to determine the lower bound on the minimum distance of C using the Shift bound. For each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(C)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{4,0\}} A_3 \xrightarrow{\{14,11\}} A_4 \xrightarrow{\{14,0\}} A_5 \xrightarrow{\{7,0\}} A_6 \xrightarrow{\{1,0\}} A_7.$$

So, $n(Z(C)) = 7$.

Let C_0 be the subcode of C by adding 0 into $Z(C)$ or $Z(C_0) = Z(C) \cup \{0\}$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_0)$, $b_i \notin Z(C_0)$, and $A_{i+1} = (a_i + A_i) \cup \{\, b_i \}$:

$$\emptyset \xrightarrow{\{0,3\}} A_1 \xrightarrow{\{1,3\}} A_2 \xrightarrow{\{4,3\}} A_3 \xrightarrow{\{11,11\}} A_4 \xrightarrow{\{17,3\}} A_5 \xrightarrow{\{28,11\}} A_6 \xrightarrow{\{7,3\}} A_7 \xrightarrow{\{1,3\}} A_8.$$

So, $n(Z(C_0)) = 8$.

Let C_3 be the subcode of C by adding cyclotomic coset $\mathcal{C}_3$ into $Z(C)$ or $Z(C_3) = Z(C) \cup \mathcal{C}_3$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_3)$, $b_i \notin Z(C_3)$, and $A_{i+1} = (a_i + A_i) \cup \{\, b_i \}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{1,0\}} A_3 \xrightarrow{\{1,0\}} A_4 \xrightarrow{\{1,0\}} A_5 \xrightarrow{\{1,0\}} A_6 \xrightarrow{\{1,0\}} A_7 \xrightarrow{\{1,0\}} A_8 \xrightarrow{\{1,0\}} A_9 \xrightarrow{\{1,0\}} A_{10} \xrightarrow{\{1,0\}} A_{11}.$$

So, $n(Z(C_3)) = 11$.

Let C_{11} be the subcode of C by adding cyclotomic coset $\mathcal{C}_{11}$ into $Z(C)$ or $Z(C_{11}) = Z(C) \cup \mathcal{C}_{11}$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_{11})$, $b_i \notin Z(C_{11})$, and $A_{i+1} = (a_i + A_i) \cup \{\, b_i \}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{7,0\}} A_3 \xrightarrow{\{13,0\}} A_4 \xrightarrow{\{1,0\}} A_5 \xrightarrow{\{14,0\}} A_6 \xrightarrow{\{14,0\}} A_7 \xrightarrow{\{22,3\}} A_8 \xrightarrow{\{16,0\}} A_9 \xrightarrow{\{7,0\}} A_{10} \xrightarrow{\{7,0\}} A_{11}.$$

So, $n(Z(C_{11})) = 11$.

Let C_{15} be the subcode of C by adding cyclotomic coset $\mathcal{C}_{15}$ into $Z(C)$ or $Z(C_{15}) = Z(C) \cup \mathcal{C}_{15}$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_{15})$, $b_i \notin Z(C_{15})$, and $A_{i+1} = (a_i + A_i) \cup \{\, b_i \}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{4,0\}} A_3 \xrightarrow{\{5,0\}} A_4 \xrightarrow{\{23,12\}} A_5 \xrightarrow{\{13,3\}} A_6 \xrightarrow{\{5,0\}} A_7 \xrightarrow{\{10,0\}} A_8 \xrightarrow{\{10,0\}} A_9 \xrightarrow{\{10,0\}} A_{10} \xrightarrow{\{9,0\}} A_{11}.$$

So, $n(Z(C_{15})) = 11$.

Let $C_{0,3}$ be the subcode of C_0 by adding the cyclotomic coset $\mathcal{C}_3$ into $Z(C_0)$, i.e. $Z(C_{0,3}) = Z(C_0) \cup \mathcal{C}_3$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_{0,3})$, $b_i \notin Z(C_{0,3})$, and $A_{i+1} = (a_i + A_i) \cup \{\, b_i \}$:

$$\emptyset \xrightarrow{\{0,11\}} A_1 \xrightarrow{\{1,11\}} A_2 \xrightarrow{\{5,11\}} A_3 \xrightarrow{\{8,23\}} A_4 \xrightarrow{\{24,15\}} A_5 \xrightarrow{\{23,21\}} A_6 \xrightarrow{\{10,21\}} A_7 \xrightarrow{\{19,11\}} A_8 \xrightarrow{\{1,11\}} A_9 \xrightarrow{\{28,11\}} A_{10} \xrightarrow{\{1,11\}} A_{11} \xrightarrow{\{27,11\}} A_{12}.$$

So, $n(Z(C_{0,3})) = 12$.

Let $C_{0,11}$ be the subcode of C_0 by adding the cyclotomic coset $\mathcal{C}_{11}$ into $Z(C_0)$, i.e. $Z(C_{0,11}) = Z(C_0) \cup \mathcal{C}_{11}$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_{0,11})$, $b_i \notin Z(C_{0,11})$, and $A_{i+1} = (a_i + A_i) \cup \{\, b_i \}$:

$$\emptyset \xrightarrow{\{0,3\}} A_1 \xrightarrow{\{1,3\}} A_2 \xrightarrow{\{5,3\}} A_3 \xrightarrow{\{2,17\}} A_4 \xrightarrow{\{11,3\}} A_5 \xrightarrow{\{28,6\}} A_6 \xrightarrow{\{7,3\}} A_7 \xrightarrow{\{15,3\}} A_8 \xrightarrow{\{4,3\}} A_9 \xrightarrow{\{18,3\}} A_{10} \xrightarrow{\{19,3\}} A_{11} \xrightarrow{\{19,3\}} A_{12}.$$

So, $n(Z(C_{0,11})) = 12$.

Let $C_{0,15}$ be the subcode of C_0 by adding the cyclotomic coset $\mathcal{C}_{15}$ into $Z(C_0)$, i.e. $Z(C_{0,15}) = Z(C_0) \cup \mathcal{C}_{15}$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_{0,15})$, $b_i \notin Z(C_{0,15})$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,3\}} A_1 \xrightarrow{\{1,3\}} A_2 \xrightarrow{\{1,3\}} A_3 \xrightarrow{\{28,11\}} A_4 \xrightarrow{\{27,17\}} A_5 \xrightarrow{\{3,21\}} A_6 \xrightarrow{\{15,21\}} A_7$$
$$\xrightarrow{\{14,17\}} A_8 \xrightarrow{\{10,3\}} A_9 \xrightarrow{\{1,6\}} A_{10} \xrightarrow{\{10,3\}} A_{11} \xrightarrow{\{11,3\}} A_{12}.$$

So $n(Z(C_{0,15})) = 12$.

Let $C_{3,11}$ be the subcode of C_3 by adding the cyclotomic coset $\mathcal{C}_{11}$ into $Z(C_3)$, i.e. $Z(C_{3,11}) = Z(C_3) \cup \mathcal{C}_{11}$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_{3,11})$, $b_i \notin Z(C_{3,11})$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{1,0\}} A_3 \xrightarrow{\{1,0\}} A_4 \xrightarrow{\{1,0\}} A_5 \xrightarrow{\{1,0\}} A_6 \xrightarrow{\{1,0\}} A_7$$
$$\xrightarrow{\{1,0\}} A_8 \xrightarrow{\{1,0\}} A_9 \xrightarrow{\{1,0\}} A_{10} \xrightarrow{\{1,0\}} A_{11} \xrightarrow{\{1,0\}} A_{12} \xrightarrow{\{1,0\}} A_{13} \xrightarrow{\{1,0\}} A_{14}$$
$$\xrightarrow{\{1,0\}} A_{15}.$$

So, $n(Z(C_{3,11})) = 15$.

Let $C_{3,15}$ be the subcode of C_3 by adding the cyclotomic coset $\mathcal{C}_{15}$ into $Z(C_3)$, i.e. $Z(C_{3,15}) = Z(C_3) \cup \mathcal{C}_{15}$. We construct a sequence of independent sets as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_{3,15})$, $b_i \notin Z(C_{3,15})$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{1,0\}} A_3 \xrightarrow{\{1,0\}} A_4 \xrightarrow{\{2,0\}} A_5 \xrightarrow{\{5,0\}} A_6 \xrightarrow{\{10,0\}} A_7$$
$$\xrightarrow{\{10,0\}} A_8 \xrightarrow{\{5,0\}} A_9 \xrightarrow{\{3,0\}} A_{10} \xrightarrow{\{2,0\}} A_{11} \xrightarrow{\{10,0\}} A_{12} \xrightarrow{\{18,0\}} A_{13} \xrightarrow{\{2,0\}} A_{14}.$$

So, $n(Z(C_{3,15})) = 14$. We stop here, because the other subcodes of C give larger on size of maximal sequence of independent sets.

By Definition 5.5,

$$\begin{aligned} d_{\text{shift}}(Z(C)) &= \min\{n(R) \,|\, Z(C) \subseteq R \subseteq \mathbb{Z}_{31} \text{ and } R^* = R \neq \mathbb{Z}_{31}\} \\ &= \min\{n(Z(C)), n(Z(C_0)), n(Z(C_3)), n(Z(C_{11})), n(Z(C_{15})), n(Z(C_{0,3})), n(Z(C_{0,11})), \\ &\quad n(Z(C_{0,15})), n(Z(C_{3,11})), n(Z(C_{3,15}))\} \\ &= \min\{7, 7, 11, 11, 11, 12, 12, 12, 15, 14\} \\ &= 7 \end{aligned}$$

and by Theorem 5.6, the minimum distance of C based on the Shift bound is $d_{\text{shift}} \geq 7$.

Example 6.18. Let C be a 9-ary cyclic codes of length 17 with defining set $\{1\}$. The complete defining set of C can be written as,

$$Z(C_1) = \{1, 2, 4, 8, 9, 13, 15, 16\},$$

which is quadratic residue over $\mathbb{F}_9$.

Since $17 \equiv 1 \bmod 4$ and $17 \equiv 1 \bmod 8$, hence by the Square root bound, the minimum distance of C is $d_{QR} \geq 5$. By the BCH bound, the minimum distance of C is $d_{BCH} \geq 3$. Observe that $Z(C)$ contains $\{1, 2\} + \{0, 7, 14\}$; therefore, the minimum distance of C by the HT bound is $d_{HT} \geq 5$.

We would like to apply Definition 5.5 and Theorem 5.6. We need to compute the maximal size independent set of C_1. And as a result of our program, the sequence of independent sets is as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_1)$, $b_i \notin Z(C_1)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{1,11\}} A_3 \xrightarrow{\{7,14\}} A_4 \xrightarrow{\{7,14\}} A_5.$$

So $n(Z(C)) = 5$. We would like to compute its lower bound on the minimum distance using the Shift bound by applying Definition 5.5 and Theorem 5.6. Therefore we need to compute the largest independent sets of all subcodes of C.

Let C_0 be the subcode of C by adding 0 into defining set of C. Hence the complete defining set of subcode C_0 is $Z(C_0) = Z(C) \cup \{0\}$. Observe that

$$Z(C_0) = \{0, 1, 2, 4, 8, 9, 13, 15, 16\}.$$

By the BCH bound, the minimum distance of C_0 is $d_{BCH} \geq 6$. Note that, $Z(C_0)$ contains $\{0, 1\} + \{0, 1, 7, 14, 15\}$. Thus, by the HT bound, the minimum distance of C_0 is $d_{HT} \geq 7$.

Next, we will compute the maximal size independent set of C_0. The sequence of independent sets is as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_0)$, $b_i \notin Z(C_0)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,3\}} A_1 \xrightarrow{\{1,3\}} A_2 \xrightarrow{\{5,6\}} A_3 \xrightarrow{\{10,3\}} A_4 \xrightarrow{\{14,6\}} A_5 \xrightarrow{\{2,14\}} A_6.$$

So $n(Z(C_0)) = 6$.

Let C_3 be the subcode of C by adding the cyclotomic coset C_3 into $Z(C)$. And for C_3, we get a maximal independent set of size $n(Z(C_3)) = 17$. By Definition 5.5,

$$d_{\text{shift}}(Z(C_1)) = \min\{n(Z(C_1)), n(Z(C_{1,0})), n(Z(C_{1,3}))\} = \min\{5, 6, 17\} = 5.$$

Thus, by Theorem 5.6, the minimum distance of C with defining set $\{1\}$ is $d_{shift} \geq 5$.

Example 6.19. Let C be a 9-ary cyclic codes of length 19 with defining set $\{1\}$. The complete defining set can be written as,

$$Z(C) = \{1, 4, 5, 6, 7, 9, 11, 16, 17\},$$

which is quadratic residue over $\mathbb{F}_9$.

Since $19 \equiv 3 \bmod 4$ and $19 \equiv 3 \bmod 8$, hence by the Square root bound, the minimum distance of C is $d_{QR}^2 - d_{QR} + 1 \geq 19$. This yield $d_{QR} = 5$. By the BCH bound, the minimum distance of C is $d_{BCH} \geq 5$.

We would like to apply Definition 5.5 and Theorem 5.6. Therefore, we need to compute a maximal size of independent set of C_1. The sequence of independent sets is as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C)$, $b_i \notin Z(C)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,0\}} A_1 \xrightarrow{\{1,0\}} A_2 \xrightarrow{\{6,18\}} A_3 \xrightarrow{\{18,12\}} A_4 \xrightarrow{\{11,0\}} A_5 \xrightarrow{\{7,18\}} A_6.$$

So $n(Z(C)) = 6$.

We would like to compute a lower bound of C using the Shift bound by applying Definition 5.5 and Theorem 5.6. Therefore we need to compute the largest independent sets of all subcodes of C. Let C_0 be the subcode of C by adding 0 into defining set of C. The complete defining set of subcode C_0 is $Z(C_0) = Z(C) \cup \{0\}$. The sequence of independent sets is as follows : for each $i > 0$, $a_i + A_i \subseteq Z(C_0)$, $b_i \notin Z(C_0)$, and $A_{i+1} = (a_i + A_i) \cup \{b_i\}$:

$$\emptyset \xrightarrow{\{0,2\}} A_1 \xrightarrow{\{2,2\}} A_2 \xrightarrow{\{2,3\}} A_3 \xrightarrow{\{3,18\}} A_4 \xrightarrow{\{10,2\}} A_5 \xrightarrow{\{7,18\}} A_6.$$

So $n(Z(C_0)) = 6$. Let C_2 be subcode of C by adding the cyclotomic coset $\mathcal{C}_2$ into $Z(C)$. And for C_2, we get $n(Z(C_2)) = 19$. By Definition 5.5,

$$d_{\text{shift}}(Z(C)) = \min\{n(Z(C)), n(Z(C_0)), n(Z(C_2))\} = \min\{6, 6, 19\} = 6.$$

Thus, by Theorem 5.6, the minimum distance of C with defining set $\{1\}$ is $d_{shift} \geq 6$.

6.3.2 Tables

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
7	$\{1\}$	3	3	3	3	3	3
17	$\{1\}$	5	4	5	5	5	5
23	$\{1\}$	7	5	5	5	6	7
31	$\{1,5,7\}$	7	5	5	6	7	7
41	$\{3\}$	9	6	7	7	8	7
47	$\{1\}$	11	5	6	6	8	8
71	$\{1\}$	11	7	7	7	10	9
73	$\{1,3,9,25\}$	13	5	7	7	10	9
79	$\{1\}$	15	7	7	7	11	10
89	$\{1,5,9,11\}$	17	5	6	7	11	9
97	$\{1\}$	15	7	8	8	12	10
103	$\{1\}$	19	8	8	8	12	11
113	$\{1,9\}$	15	6	7	7	13	11

Table 6.1: *Table of the minimum distance for 2-ary quadratic residue codes.*

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
11	$\{1\}$	5	4	4	4	4	4
13	$\{1,4\}$	5	3	3	4	5	4
23	$\{1\}$	8	5	5	5	6	6
37	$\{1\}$	10	5	6	6	8	7
47	$\{5\}$	14	5	6	6	8	7
59	$\{1\}$	17	6	6	6	9	9
61	$\{1,4,5\}$	11	6	7	7	10	8
71	$\{1\}$	17	7	7	7	10	9
73	$\{1,2,4\}$	17	5	7	7	10	9

Table 6.2: *Table of the minimum distance for ternary quadratic residue codes.*

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
5	$\{1\}$	3	2	2	3	3	3
11	$\{1\}$	5	4	4	4	4	4
13	$\{1\}$	5	3	3	4	5	4
19	$\{1\}$	7	5	5	5	6	5
29	$\{1\}$	11	5	6	6	7	6
37	$\{1\}$	11	5	6	6	8	7
41	$\{3,6\}$	9	6	7	7	8	7
43	$\{1,6,9\}$	13	6	6	6	8	7
53	$\{1\}$	13	4	5	6	9	8
59	$\{1\}$	13	6	6	6	9	8

Table 6.3: *Table of the minimum distance for quaternary quadratic residue codes.*

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
11	$\{1\}$	5	4	4	4	4	4
19	$\{1\}$	7	5	5	5	6	5
29	$\{1\}$	11	5	6	6	7	6
31	$\{1,2,4,8,16\}$	9	5	5	6	7	6

Table 6.4: *Table of the minimum distance for quinary quadratic residue codes.*

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
19	$\{1\}$	8	5	5	5	6	5
29	$\{1,4\}$	11	5	6	6	7	6
31	$\{1,4,5\}$	12	5	5	6	7	6

Table 6.5: *Table of the minimum distance for 7-ary quadratic residue codes.*

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
7	$\{1\}$	4	3	3	3	3	3
17	$\{1\}$	7	3	4	5	5	5
19	$\{1\}$	9	5	5	5	6	5
29	$\{1\}$	11	5	6	6	7	6
31	$\{1\}$	11	5	5	6	7	6

Table 6.6: *Table of the minimum distance for 9-ary quadratic residue codes.*

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
7	$\{1\}$	4	3	3	3	3	3
13	$\{1,3,4\}$	7	3	3	4	5	4
17	$\{1\}$	9	3	4	5	5	5
23	$\{1\}$	11	5	5	5	6	6

Table 6.7: *Table of the minimum distance for 25-ary quadratic residue codes.*

n	roots	d	d_{BCH}	d_{HT}	d_{Roos}	d_{shift}	d_{QR}
5	$\{1\}$	3	2	2	3	3	3
11	$\{1\}$	6	4	4	4	4	4
13	$\{1\}$	6	5	5	5	5	4
17	$\{1\}$	9	4	5	5	5	5
23	$\{1\}$	11	5	5	5	6	6

Table 6.8: *Table of the minimum distance for 49-ary quadratic residue codes.*

Remark 6.20. Notice that $d_{shift} \geq d_{QR}$ for all cases considered. Except for the binary cyclic code of length 23.

7
Computational results

In this chapter, we give a result based on the computation by our minimum distance program.

k, the dimension of the code equals to $n - Z(C)$, where $Z(C)$ is the complete defining set of C.

d_{BCH}, lower bound on the minimum distance based on the BCH bound.

d_{HT}, lower bound on the minimum distance based on the HT bound.

d_{HTR}, lower bound on the minimum distance based on the HTR bound.

d_{Roos}, lower bound on the minimum distance based on the Roos bound.

$n(Z)$, the maximum size of the sequence of independent sets with $Z \subseteq \mathbb{Z}_n$.

d_{shift}, lower bound on the minimum distance based on the Shift bound.

$d_{brouwer}$, bound on the minimum distance of linear codes from Brouwer's table [1].

7.1. Binary cyclic codes of length 45

```
Coset representatives are :
 ( 0 1 3 5 7 9 15 21 )
The cyclotomic cosets are :
 ( 0 )
 ( 1 2 4 8 16 17 19 23 31 32 34 38 )
 ( 3 6 12 24 )
 ( 5 10 20 25 35 40 )
 ( 7 11 13 14 22 26 28 29 37 41 43 44 )
 ( 9 18 27 36 )
 ( 15 30 )
 ( 21 33 39 42 )


Table for the lower bounds on the minimum distance of
2-ary Cyclic codes of length 45

  #   k  d_BCH  d_HT  d_HTR  d_Roos   n(Z)   root(s)
  1   0     46    46     46      46     46   { 0 1 3 5 7 9 15 21 }
  2   1     45    45     45      45     45   { 1 3 5 7 9 15 21 }
```

3	2	30	30	30	30	30	{ 0 1 3 5 7 9 21 }
4	3	15	15	15	15	15	{ 1 3 5 7 9 21 }
5	4	24	24	24	24	24	{ 0 1 3 5 7 9 15 }
6	4	18	18	18	18	18	{ 0 1 3 5 7 15 21 }
7	4	24	24	24	24	24	{ 0 1 5 7 9 15 21 }
8	5	21	21	21	21	21	{ 1 3 5 7 9 15 }
9	5	9	9	9	9	9	{ 1 3 5 7 15 21 }
10	5	21	21	21	21	21	{ 1 5 7 9 15 21 }
11	6	18	18	18	18	18	{ 0 1 3 5 7 9 }
12	6	18	18	18	18	18	{ 0 1 3 5 7 21 }
13	6	18	18	18	18	18	{ 0 1 5 7 9 21 }
14	6	10	10	10	10	10	{ 0 1 3 7 9 15 21 }
15	7	15	15	15	15	15	{ 1 3 5 7 9 }
16	7	9	9	9	9	18	{ 1 3 5 7 21 }
17	7	15	15	15	15	15	{ 1 5 7 9 21 }
18	7	10	10	10	10	15	{ 1 3 7 9 15 21 }
19	8	12	12	12	12	12	{ 0 1 3 5 7 15 }
20	8	10	10	10	10	10	{ 0 1 3 7 9 21 }
21	8	9	10	11	12	12	{ 0 1 5 7 9 15 }
22	8	12	12	12	12	12	{ 0 1 5 7 15 21 }
23	9	9	9	9	9	15	{ 1 3 5 7 15 }
24	9	5	5	5	5	5	{ 1 3 7 9 21 }
25	9	9	10	11	12	15	{ 1 5 7 9 15 }
26	9	9	9	9	9	15	{ 1 5 7 15 21 }
27	10	12	12	12	12	12	{ 0 1 3 5 7 }
28	10	6	6	6	6	6	{ 0 1 5 7 9 }
29	10	12	12	12	12	12	{ 0 1 5 7 21 }
30	10	10	10	10	10	18	{ 0 1 3 7 9 15 }
31	10	10	10	10	10	18	{ 0 1 3 7 15 21 }
32	10	10	10	10	10	18	{ 0 1 7 9 15 21 }
33	11	9	9	9	9	9	{ 1 3 5 7 }
34	11	6	6	6	6	15	{ 1 5 7 9 }
35	11	9	9	9	9	9	{ 1 5 7 21 }
36	11	10	10	10	10	16	{ 1 3 7 9 15 }
37	11	8	9	9	9	15	{ 1 3 7 15 21 }
38	11	10	10	10	10	16	{ 1 7 9 15 21 }
39	12	8	8	8	10	16	{ 0 1 3 7 9 }
40	12	10	10	10	10	12	{ 0 1 3 7 21 }
41	12	6	6	6	6	6	{ 0 1 5 7 15 }
42	12	8	8	8	9	15	{ 0 1 7 9 21 }
43	12	8	8	8	8	8	{ 0 1 3 5 9 15 21 }
44	12	8	8	8	8	8	{ 0 3 5 7 9 15 21 }
45	13	5	5	5	5	15	{ 1 3 7 9 }
46	13	5	5	5	5	12	{ 1 3 7 21 }
47	13	6	6	6	6	9	{ 1 5 7 15 }
48	13	5	5	5	5	15	{ 1 7 9 21 }
49	13	8	8	8	8	21	{ 1 3 5 9 15 21 }
50	13	8	8	8	8	19	{ 3 5 7 9 15 21 }
51	14	6	6	6	6	6	{ 0 1 5 7 }
52	14	8	9	9	10	12	{ 0 1 3 7 15 }
53	14	8	9	9	10	11	{ 0 1 7 9 15 }
54	14	8	9	9	10	12	{ 0 1 7 15 21 }
55	14	8	8	8	8	13	{ 0 1 3 5 9 21 }
56	14	8	8	8	8	14	{ 0 3 5 7 9 21 }
57	15	3	3	3	3	3	{ 1 5 7 }
58	15	8	8	8	8	12	{ 1 3 7 15 }
59	15	8	9	9	10	11	{ 1 7 9 15 }
60	15	8	8	8	8	12	{ 1 7 15 21 }
61	15	7	7	7	7	7	{ 1 3 5 9 21 }
62	15	7	7	7	7	7	{ 3 5 7 9 21 }
63	16	8	8	8	8	10	{ 0 1 3 7 }
64	16	6	6	6	6	9	{ 0 1 7 9 }
65	16	8	8	8	8	10	{ 0 1 7 21 }
66	16	8	8	8	8	16	{ 0 1 3 5 9 15 }

67	16	8	8	8	8	12	{ 0 1 3 5 15 21 }
68	16	8	8	8	8	16	{ 0 1 5 9 15 21 }
69	16	8	8	8	8	16	{ 0 3 5 7 9 15 }
70	16	8	8	8	8	12	{ 0 3 5 7 15 21 }
71	16	8	8	8	8	16	{ 0 5 7 9 15 21 }
72	17	5	5	5	5	10	{ 1 3 7 }
73	17	5	5	5	5	9	{ 1 7 9 }
74	17	5	5	5	5	10	{ 1 7 21 }
75	17	7	8	8	8	12	{ 1 3 5 9 15 }
76	17	7	8	8	8	11	{ 1 3 5 15 21 }
77	17	8	8	8	8	12	{ 1 5 9 15 21 }
78	17	8	8	8	8	12	{ 3 5 7 9 15 }
79	17	7	8	8	8	12	{ 3 5 7 15 21 }
80	17	7	8	8	8	12	{ 5 7 9 15 21 }
81	18	6	6	6	6	6	{ 0 1 7 15 }
82	18	8	8	8	8	13	{ 0 1 3 5 9 }
83	18	8	8	8	8	10	{ 0 1 3 5 21 }
84	18	7	7	7	7	13	{ 0 1 5 9 21 }
85	18	7	7	7	7	13	{ 0 3 5 7 9 }
86	18	8	8	8	8	10	{ 0 3 5 7 21 }
87	18	8	8	8	8	13	{ 0 5 7 9 21 }
88	18	6	6	6	6	6	{ 0 1 3 9 15 21 }
89	18	6	6	6	6	6	{ 0 3 7 9 15 21 }
90	19	6	6	6	6	10	{ 1 7 15 }
91	19	7	7	7	7	13	{ 1 3 5 9 }
92	19	7	7	7	7	10	{ 1 3 5 21 }
93	19	7	7	7	7	13	{ 1 5 9 21 }
94	19	7	7	7	7	13	{ 3 5 7 9 }
95	19	7	7	7	7	10	{ 3 5 7 21 }
96	19	7	7	7	7	13	{ 5 7 9 21 }
97	19	6	6	6	6	15	{ 1 3 9 15 21 }
98	19	6	6	6	6	15	{ 3 7 9 15 21 }
99	20	6	6	6	6	8	{ 0 1 7 }
100	20	8	8	8	8	10	{ 0 1 3 5 15 }
101	20	6	6	6	6	10	{ 0 1 3 9 21 }
102	20	7	7	7	7	8	{ 0 1 5 9 15 }
103	20	7	7	7	7	10	{ 0 1 5 15 21 }
104	20	7	7	7	7	10	{ 0 3 5 7 15 }
105	20	6	6	6	6	10	{ 0 3 7 9 21 }
106	20	7	7	7	7	8	{ 0 5 7 9 15 }
107	20	8	8	8	8	10	{ 0 5 7 15 21 }
108	21	3	3	3	3	6	{ 1 7 }
109	21	7	7	7	7	11	{ 1 3 5 15 }
110	21	5	5	5	5	5	{ 1 3 9 21 }
111	21	7	7	7	7	11	{ 1 5 9 15 }
112	21	7	7	7	7	11	{ 1 5 15 21 }
113	21	7	7	7	7	11	{ 3 5 7 15 }
114	21	5	5	5	5	5	{ 3 7 9 21 }
115	21	7	7	7	7	11	{ 5 7 9 15 }
116	21	7	7	7	7	11	{ 5 7 15 21 }
117	22	8	8	8	8	8	{ 0 1 3 5 }
118	22	6	6	6	6	8	{ 0 1 5 9 }
119	22	6	7	7	7	8	{ 0 1 5 21 }
120	22	6	7	7	7	8	{ 0 3 5 7 }
121	22	6	6	6	6	8	{ 0 5 7 9 }
122	22	8	8	8	8	8	{ 0 5 7 21 }
123	22	6	6	6	6	12	{ 0 1 3 9 15 }
124	22	6	6	6	6	12	{ 0 1 3 15 21 }
125	22	6	6	6	6	12	{ 0 1 9 15 21 }
126	22	6	6	6	6	12	{ 0 3 7 9 15 }
127	22	6	6	6	6	12	{ 0 3 7 15 21 }
128	22	6	6	6	6	12	{ 0 7 9 15 21 }
129	23	7	7	7	7	7	{ 1 3 5 }
130	23	6	6	6	6	7	{ 1 5 9 }

131	23	6	6	6	6	8	{ 1 5 21 }
132	23	6	6	6	7	7	{ 3 5 7 }
133	23	6	6	6	6	7	{ 5 7 9 }
134	23	7	7	7	7	8	{ 5 7 21 }
135	23	6	6	6	6	9	{ 1 3 9 15 }
136	23	6	6	6	6	9	{ 1 3 15 21 }
137	23	6	6	6	6	9	{ 1 9 15 21 }
138	23	6	6	6	6	9	{ 3 7 9 15 }
139	23	6	6	6	6	9	{ 3 7 15 21 }
140	23	6	6	6	6	9	{ 7 9 15 21 }
141	24	6	6	6	6	10	{ 0 1 3 9 }
142	24	6	6	6	6	10	{ 0 1 3 21 }
143	24	4	4	4	4	4	{ 0 1 5 15 }
144	24	5	6	6	6	10	{ 0 1 9 21 }
145	24	5	6	6	6	10	{ 0 3 7 9 }
146	24	6	6	6	6	10	{ 0 3 7 21 }
147	24	4	4	4	4	4	{ 0 5 7 15 }
148	24	6	6	6	6	10	{ 0 7 9 21 }
149	24	3	4	4	4	4	{ 0 3 5 9 15 21 }
150	25	5	5	5	5	10	{ 1 3 9 }
151	25	5	5	5	5	10	{ 1 3 21 }
152	25	4	4	4	4	11	{ 1 5 15 }
153	25	5	5	5	5	10	{ 1 9 21 }
154	25	5	5	5	5	10	{ 3 7 9 }
155	25	5	5	5	5	10	{ 3 7 21 }
156	25	4	4	4	4	11	{ 5 7 15 }
157	25	5	5	5	5	10	{ 7 9 21 }
158	25	3	4	4	4	15	{ 3 5 9 15 21 }
159	26	4	4	4	4	8	{ 0 1 5 }
160	26	4	4	4	4	8	{ 0 5 7 }
161	26	6	6	6	6	8	{ 0 1 3 15 }
162	26	6	6	6	6	8	{ 0 1 9 15 }
163	26	6	6	6	6	8	{ 0 1 15 21 }
164	26	6	6	6	6	8	{ 0 3 7 15 }
165	26	6	6	6	6	8	{ 0 7 9 15 }
166	26	6	6	6	6	8	{ 0 7 15 21 }
167	26	3	4	4	4	10	{ 0 3 5 9 21 }
168	27	3	3	3	3	5	{ 1 5 }
169	27	3	3	3	3	5	{ 5 7 }
170	27	5	5	5	6	9	{ 1 3 15 }
171	27	6	6	6	6	9	{ 1 9 15 }
172	27	6	6	6	6	9	{ 1 15 21 }
173	27	6	6	6	6	9	{ 3 7 15 }
174	27	6	6	6	6	9	{ 7 9 15 }
175	27	5	5	5	6	9	{ 7 15 21 }
176	27	3	4	4	4	5	{ 3 5 9 21 }
177	28	6	6	6	6	6	{ 0 1 3 }
178	28	5	5	5	5	6	{ 0 1 9 }
179	28	5	5	5	5	6	{ 0 1 21 }
180	28	5	5	5	5	6	{ 0 3 7 }
181	28	5	5	5	5	6	{ 0 7 9 }
182	28	6	6	6	6	6	{ 0 7 21 }
183	28	3	4	4	4	6	{ 0 3 5 9 15 }
184	28	3	4	4	4	6	{ 0 3 5 15 21 }
185	28	3	4	4	4	6	{ 0 5 9 15 21 }
186	29	5	5	5	5	7	{ 1 3 }
187	29	5	5	5	5	7	{ 1 9 }
188	29	5	5	5	5	7	{ 1 21 }
189	29	5	5	5	5	7	{ 3 7 }
190	29	5	5	5	5	7	{ 7 9 }
191	29	5	5	5	5	7	{ 7 21 }
192	29	3	4	4	4	9	{ 3 5 9 15 }
193	29	3	4	4	4	9	{ 3 5 15 21 }
194	29	3	4	4	4	9	{ 5 9 15 21 }

195	30	4	4	4	4	4	{ 0 1 15 }
196	30	4	4	4	4	4	{ 0 7 15 }
197	30	3	4	4	4	8	{ 0 3 5 9 }
198	30	3	4	4	4	8	{ 0 3 5 21 }
199	30	3	4	4	4	8	{ 0 5 9 21 }
200	30	2	2	2	2	2	{ 0 3 9 15 21 }
201	31	4	4	4	4	9	{ 1 15 }
202	31	4	4	4	4	9	{ 7 15 }
203	31	3	4	4	4	7	{ 3 5 9 }
204	31	3	4	4	4	7	{ 3 5 21 }
205	31	3	4	4	4	7	{ 5 9 21 }
206	31	2	2	2	2	15	{ 3 9 15 21 }
207	32	4	4	4	4	6	{ 0 1 }
208	32	4	4	4	4	6	{ 0 7 }
209	32	3	4	4	4	4	{ 0 3 5 15 }
210	32	2	2	2	2	10	{ 0 3 9 21 }
211	32	3	4	4	4	4	{ 0 5 9 15 }
212	32	3	4	4	4	4	{ 0 5 15 21 }
213	33	3	3	3	3	3	{ 1 }
214	33	3	3	3	3	3	{ 7 }
215	33	3	4	4	4	9	{ 3 5 15 }
216	33	2	2	2	2	5	{ 3 9 21 }
217	33	3	4	4	4	9	{ 5 9 15 }
218	33	3	4	4	4	9	{ 5 15 21 }
219	34	3	4	4	4	6	{ 0 3 5 }
220	34	3	4	4	4	6	{ 0 5 9 }
221	34	3	4	4	4	6	{ 0 5 21 }
222	34	2	2	2	2	8	{ 0 3 9 15 }
223	34	2	2	2	2	6	{ 0 3 15 21 }
224	34	2	2	2	2	8	{ 0 9 15 21 }
225	35	3	4	4	4	5	{ 3 5 }
226	35	3	4	4	4	5	{ 5 9 }
227	35	3	4	4	4	5	{ 5 21 }
228	35	2	2	2	2	7	{ 3 9 15 }
229	35	2	2	2	2	3	{ 3 15 21 }
230	35	2	2	2	2	7	{ 9 15 21 }
231	36	2	2	2	2	6	{ 0 3 9 }
232	36	2	2	2	2	6	{ 0 3 21 }
233	36	2	2	2	2	2	{ 0 5 15 }
234	36	2	2	2	2	6	{ 0 9 21 }
235	37	2	2	2	2	5	{ 3 9 }
236	37	2	2	2	2	6	{ 3 21 }
237	37	2	2	2	2	9	{ 5 15 }
238	37	2	2	2	2	5	{ 9 21 }
239	38	2	2	2	2	6	{ 0 5 }
240	38	2	2	2	2	4	{ 0 3 15 }
241	38	2	2	2	2	4	{ 0 9 15 }
242	38	2	2	2	2	4	{ 0 15 21 }
243	39	2	2	2	2	3	{ 5 }
244	39	2	2	2	2	5	{ 3 15 }
245	39	2	2	2	2	5	{ 9 15 }
246	39	2	2	2	2	5	{ 15 21 }
247	40	2	2	2	2	4	{ 0 3 }
248	40	2	2	2	2	2	{ 0 9 }
249	40	2	2	2	2	4	{ 0 21 }
250	41	2	2	2	2	3	{ 3 }
251	41	2	2	2	2	5	{ 9 }
252	41	2	2	2	2	3	{ 21 }
253	41	2	2	2	2	3	{ 21 }
254	42	2	2	2	2	2	{ 0 15 }
255	43	2	2	2	2	3	{ 15 }
256	44	2	2	2	2	2	{ 0 }

#	k	d_shift	d_brouwer
1	0	46	
2	1	45	45
3	2	30	30
4	3	15	25
5	4	24	23
6	4	18	
7	4	24	
8	5	21	22
9	5	9	
10	5	21	
11	6	18	22
12	6	18	
13	6	18	
14	6	10	
15	7	15	20
16	7	9	
17	7	15	
18	7	10	
19	8	12	20
20	8	10	
21	8	12	
22	8	12	
23	9	9	18-19
24	9	5	
25	9	12	
26	9	9	
27	10	12	18
28	10	6	
29	10	12	
30	10	10	
31	10	10	
32	10	10	
33	11	9	16-18
34	11	6	
35	11	9	
36	11	10	
37	11	9	
38	11	10	
39	12	10	16
40	12	10	
41	12	6	
42	12	10	
43	12	8	
44	12	8	
45	13	5	16
46	13	5	
47	13	6	
48	13	5	
49	13	8	
50	13	8	

51	14	6	16
52	14	10	
53	14	10	
54	14	10	
55	14	8	
56	14	8	
57	15	3	14-15
58	15	9	
59	15	10	
60	15	9	
61	15	7	
62	15	7	
63	16	10	13-14
64	16	6	
65	16	10	
66	16	8	
67	16	8	
68	16	8	
69	16	8	
70	16	8	
71	16	8	
72	17	5	12-14
73	17	5	
74	17	5	
75	17	8	
76	17	8	
77	17	8	
78	17	8	
79	17	8	
80	17	8	
81	18	6	12-13
82	18	8	
83	18	8	
84	18	8	
85	18	8	
86	18	8	
87	18	8	
88	18	6	
89	18	6	
90	19	6	12
91	19	7	
92	19	7	
93	19	7	
94	19	7	
95	19	7	
96	19	7	
97	19	6	
98	19	6	
99	20	6	12
100	20	8	
101	20	6	
102	20	8	
103	20	8	
104	20	8	
105	20	6	
106	20	8	
107	20	8	

108	21	3	12
109	21	8	
110	21	5	
111	21	8	
112	21	8	
113	21	8	
114	21	5	
115	21	8	
116	21	8	
117	22	8	11
118	22	6	
119	22	8	
120	22	8	
121	22	6	
122	22	8	
123	22	6	
124	22	6	
125	22	6	
126	22	6	
127	22	6	
128	22	6	
129	23	7	10
130	23	6	
131	23	7	
132	23	7	
133	23	6	
134	23	7	
135	23	6	
136	23	6	
137	23	6	
138	23	6	
139	23	6	
140	23	6	
141	24	6	9-10
142	24	6	
143	24	4	
144	24	6	
145	24	6	
146	24	6	
147	24	4	
148	24	6	
149	24	4	
150	25	5	8-10
151	25	5	
152	25	4	
153	25	5	
154	25	5	
155	25	5	
156	25	4	
157	25	5	
158	25	4	
159	26	4	8-9
160	26	4	
161	26	6	
162	26	6	
163	26	6	
164	26	6	
165	26	6	

166	26	6	
167	26	4	
168	27	3	8
169	27	3	
170	27	6	
171	27	6	
172	27	6	
173	27	6	
174	27	6	
175	27	6	
176	27	4	
177	28	6	8
178	28	6	
179	28	6	
180	28	6	
181	28	6	
182	28	6	
183	28	4	
184	28	4	
185	28	4	
186	29	5	7-8
187	29	5	
188	29	5	
189	29	5	
190	29	5	
191	29	5	
192	29	4	
193	29	4	
194	29	4	
195	30	4	6-7
196	30	4	
197	30	4	
198	30	4	
199	30	4	
200	30	2	
201	31	4	6
202	31	4	
203	31	4	
204	31	4	
205	31	4	
206	31	2	
207	32	4	6
208	32	4	
209	32	4	
210	32	2	
211	32	4	
212	32	4	
213	33	3	6
214	33	3	
215	33	4	
216	33	2	
217	33	4	
218	33	4	
219	34	4	5
220	34	4	
221	34	4	

```
222  34         2
223  34         2
224  34         2
-----------------------------
225  35         4           4
226  35         4
227  35         4
228  35         2
229  35         2
230  35         2
-----------------------------
231  36         2           4
232  36         2
233  36         2
234  36         2
-----------------------------
235  37         2           4
236  37         2
237  37         2
238  37         2
-----------------------------
239  38         2           4
240  38         2
241  38         2
242  38         2
-----------------------------
243  39         2           3
244  39         2
245  39         2
246  39         2
-----------------------------
247  40         2           2
248  40         2
249  40         2
-----------------------------
250  41         2           2
251  41         2
252  41         2
253  41         2
-----------------------------
254  42         2           2
-----------------------------
255  43         2           2
-----------------------------
256  44         2           2
```

Remark 7.1. For $n = 45$, $d_{BCH} \leq d_{HT} \leq d_{HTR} \leq d_{Roos} \leq d_{shift} = d$.

7.2. Binary cyclic codes of length 73

```
Coset representatives are :
 ( 0 1 3 5 9 11 13 17 25 )
The cyclotomic cosets are :
 ( 0 )
 ( 1 2 4 8 16 32 37 55 64 )
 ( 3 6 12 19 23 24 38 46 48 )
 ( 5 7 10 14 20 28 39 40 56 )
 ( 9 18 36 41 57 65 69 71 72 )
 ( 11 15 21 22 30 42 44 47 60 )
 ( 13 26 29 31 43 51 52 58 62 )
 ( 17 33 34 45 53 59 63 66 68 )
 ( 25 27 35 49 50 54 61 67 70 )
```

Table for the lower bounds on the minimum distance of
2-ary Cyclic codes of length 73

#	k	d_BCH	d_HT	d_HTR	d_Roos	n(Z)	root(s)
1	0	74	74	74	74	74	{ 0 1 3 5 9 11 13 17 25 }
2	1	73	73	73	73	73	{ 1 3 5 9 11 13 17 25 }
3	9	28	28	28	28	28	{ 0 1 3 5 9 11 13 17 }
4	9	28	28	28	28	28	{ 0 1 3 5 9 11 13 25 }
5	9	28	28	28	28	28	{ 0 1 3 5 9 11 17 25 }
6	9	28	28	28	28	28	{ 0 1 3 5 9 13 17 25 }
7	9	28	28	28	28	28	{ 0 1 3 5 11 13 17 25 }
8	9	28	28	28	28	28	{ 0 1 3 9 11 13 17 25 }
9	9	28	28	28	28	28	{ 0 1 5 9 11 13 17 25 }
10	9	28	28	28	28	28	{ 0 3 5 9 11 13 17 25 }
11	10	25	25	25	25	31	{ 1 3 5 9 11 13 17 }
12	10	25	25	25	27	30	{ 1 3 5 9 11 13 25 }
13	10	25	25	25	25	30	{ 1 3 5 9 11 17 25 }
14	10	25	25	25	25	31	{ 1 3 5 9 13 17 25 }
15	10	25	25	25	26	30	{ 1 3 5 11 13 17 25 }
16	10	25	25	25	25	30	{ 1 3 9 11 13 17 25 }
17	10	25	25	25	25	30	{ 1 5 9 11 13 17 25 }
18	10	25	25	25	25	31	{ 3 5 9 11 13 17 25 }
19	18	20	20	20	20	24	{ 0 1 3 5 9 11 13 }
20	18	16	16	16	16	22	{ 0 1 3 5 9 11 17 }
21	18	18	18	18	18	21	{ 0 1 3 5 9 11 25 }
22	18	20	20	20	20	23	{ 0 1 3 5 9 13 17 }
23	18	18	18	18	18	22	{ 0 1 3 5 9 13 25 }
24	18	22	22	22	22	21	{ 0 1 3 5 9 17 25 }
25	18	18	18	18	18	22	{ 0 1 3 5 11 13 17 }
26	18	16	16	16	16	22	{ 0 1 3 5 11 13 25 }
27	18	16	16	16	17	21	{ 0 1 3 5 11 17 25 }
28	18	20	20	20	20	24	{ 0 1 3 5 13 17 25 }
29	18	16	16	16	16	20	{ 0 1 3 9 11 13 17 }
30	18	22	22	22	22	22	{ 0 1 3 9 11 13 25 }
31	18	18	18	18	18	22	{ 0 1 3 9 11 17 25 }
32	18	18	18	18	18	22	{ 0 1 3 9 13 17 25 }
33	18	20	20	20	20	22	{ 0 1 3 11 13 17 25 }
34	18	22	22	22	22	22	{ 0 1 5 9 11 13 17 }
35	18	16	16	16	16	21	{ 0 1 5 9 11 13 25 }
36	18	20	20	20	20	24	{ 0 1 5 9 11 17 25 }
37	18	16	16	16	16	22	{ 0 1 5 9 13 17 25 }
38	18	18	18	18	18	22	{ 0 1 5 11 13 17 25 }
39	18	20	20	20	20	24	{ 0 1 9 11 13 17 25 }
40	18	18	18	18	18	22	{ 0 3 5 9 11 13 17 }
41	18	20	20	20	20	24	{ 0 3 5 9 11 13 25 }
42	18	20	20	20	20	24	{ 0 3 5 9 11 17 25 }
43	18	16	16	16	16	22	{ 0 3 5 9 13 17 25 }
44	18	22	22	22	22	22	{ 0 3 5 11 13 17 25 }
45	18	16	16	16	16	22	{ 0 3 9 11 13 17 25 }
46	18	18	18	18	18	21	{ 0 5 9 11 13 17 25 }
47	19	17	17	17	17	21	{ 1 3 5 9 11 13 }
48	19	15	15	15	15	20	{ 1 3 5 9 11 17 }
49	19	13	13	13	16	21	{ 1 3 5 9 11 25 }
50	19	17	17	17	17	21	{ 1 3 5 9 13 17 }
51	19	13	13	13	13	21	{ 1 3 5 9 13 25 }
52	19	14	15	15	18	19	{ 1 3 5 9 17 25 }
53	19	13	13	13	16	21	{ 1 3 5 11 13 17 }
54	19	15	15	15	16	20	{ 1 3 5 11 13 25 }
55	19	15	15	15	15	21	{ 1 3 5 11 17 25 }
56	19	17	17	17	17	21	{ 1 3 5 13 17 25 }
57	19	15	15	15	15	21	{ 1 3 9 11 13 17 }
58	19	14	15	15	16	19	{ 1 3 9 11 13 25 }
59	19	13	13	13	14	21	{ 1 3 9 11 17 25 }
60	19	13	13	13	15	21	{ 1 3 9 13 17 25 }

61	19	17	17	17	17	21	{ 1 3 11 13 17 25 }
62	19	14	15	15	19	19	{ 1 5 9 11 13 17 }
63	19	15	15	15	15	21	{ 1 5 9 11 13 25 }
64	19	17	17	17	17	21	{ 1 5 9 11 17 25 }
65	19	15	15	15	15	20	{ 1 5 9 13 17 25 }
66	19	13	13	13	16	21	{ 1 5 11 13 17 25 }
67	19	17	17	17	17	21	{ 1 9 11 13 17 25 }
68	19	13	13	13	16	21	{ 3 5 9 11 13 17 }
69	19	17	17	17	17	21	{ 3 5 9 11 13 25 }
70	19	17	17	17	17	21	{ 3 5 9 11 17 25 }
71	19	15	15	15	15	21	{ 3 5 9 13 17 25 }
72	19	14	15	15	19	19	{ 3 5 11 13 17 25 }
73	19	15	15	15	16	21	{ 3 9 11 13 17 25 }
74	19	13	13	13	16	21	{ 5 9 11 13 17 25 }
75	27	16	16	16	16	18	{ 0 1 3 5 9 11 }
76	27	14	14	14	14	16	{ 0 1 3 5 9 13 }
77	27	14	14	14	14	16	{ 0 1 3 5 9 17 }
78	27	16	16	16	16	16	{ 0 1 3 5 9 25 }
79	27	10	10	10	10	16	{ 0 1 3 5 11 13 }
80	27	10	10	10	11	16	{ 0 1 3 5 11 17 }
81	27	10	10	10	11	16	{ 0 1 3 5 11 25 }
82	27	14	14	14	14	15	{ 0 1 3 5 13 17 }
83	27	16	16	16	16	18	{ 0 1 3 5 13 25 }
84	27	14	14	14	14	14	{ 0 1 3 5 17 25 }
85	27	14	14	14	14	15	{ 0 1 3 9 11 13 }
86	27	10	10	10	11	15	{ 0 1 3 9 11 17 }
87	27	16	16	16	16	16	{ 0 1 3 9 11 25 }
88	27	10	10	10	10	16	{ 0 1 3 9 13 17 }
89	27	16	16	16	16	16	{ 0 1 3 9 13 25 }
90	27	16	16	16	16	16	{ 0 1 3 9 17 25 }
91	27	16	16	16	16	18	{ 0 1 3 11 13 17 }
92	27	14	14	14	14	16	{ 0 1 3 11 13 25 }
93	27	10	10	10	10	16	{ 0 1 3 11 17 25 }
94	27	14	14	14	14	15	{ 0 1 3 13 17 25 }
95	27	14	14	14	14	15	{ 0 1 5 9 11 13 }
96	27	14	14	14	14	15	{ 0 1 5 9 11 17 }
97	27	10	10	10	10	16	{ 0 1 5 9 11 25 }
98	27	14	14	14	14	15	{ 0 1 5 9 13 17 }
99	27	10	10	10	11	16	{ 0 1 5 9 13 25 }
100	27	14	14	14	14	15	{ 0 1 5 9 17 25 }
101	27	16	16	16	16	16	{ 0 1 5 11 13 17 }
102	27	10	10	10	11	16	{ 0 1 5 11 13 25 }
103	27	16	16	16	16	18	{ 0 1 5 11 17 25 }
104	27	10	10	10	10	16	{ 0 1 5 13 17 25 }
105	27	14	14	14	14	16	{ 0 1 9 11 13 17 }
106	27	14	14	14	14	14	{ 0 1 9 11 13 25 }
107	27	14	14	14	14	15	{ 0 1 9 11 17 25 }
108	27	16	16	16	16	18	{ 0 1 9 13 17 25 }
109	27	14	14	14	14	16	{ 0 1 11 13 17 25 }
110	27	14	14	14	14	16	{ 0 3 5 9 11 13 }
111	27	10	10	10	10	16	{ 0 3 5 9 11 17 }
112	27	14	14	14	14	16	{ 0 3 5 9 11 25 }
113	27	16	16	16	16	18	{ 0 3 5 9 13 17 }
114	27	10	10	10	10	16	{ 0 3 5 9 13 25 }
115	27	14	14	14	14	14	{ 0 3 5 9 17 25 }
116	27	16	16	16	16	16	{ 0 3 5 11 13 17 }
117	27	14	14	14	14	14	{ 0 3 5 11 13 25 }
118	27	14	14	14	14	16	{ 0 3 5 11 17 25 }
119	27	14	14	14	14	15	{ 0 3 5 13 17 25 }
120	27	10	10	10	11	15	{ 0 3 9 11 13 17 }
121	27	14	14	14	14	16	{ 0 3 9 11 13 25 }
122	27	16	16	16	16	18	{ 0 3 9 11 17 25 }
123	27	10	10	10	11	15	{ 0 3 9 13 17 25 }
124	27	14	14	14	14	14	{ 0 3 11 13 17 25 }

125	27	16	16	16	16	16	{ 0 5 9 11 13 17 }
126	27	16	16	16	16	18	{ 0 5 9 11 13 25 }
127	27	14	14	14	14	16	{ 0 5 9 11 17 25 }
128	27	10	10	10	11	15	{ 0 5 9 13 17 25 }
129	27	16	16	16	16	16	{ 0 5 11 13 17 25 }
130	27	10	10	10	10	16	{ 0 9 11 13 17 25 }
131	28	13	13	13	13	15	{ 1 3 5 9 11 }
132	28	11	11	11	11	15	{ 1 3 5 9 13 }
133	28	11	11	11	11	13	{ 1 3 5 9 17 }
134	28	11	11	11	11	15	{ 1 3 5 9 25 }
135	28	9	9	9	9	9	{ 1 3 5 11 13 }
136	28	9	9	9	10	14	{ 1 3 5 11 17 }
137	28	9	9	9	11	15	{ 1 3 5 11 25 }
138	28	11	11	11	11	15	{ 1 3 5 13 17 }
139	28	13	13	13	13	15	{ 1 3 5 13 25 }
140	28	14	14	14	14	15	{ 1 3 5 17 25 }
141	28	14	14	14	14	15	{ 1 3 9 11 13 }
142	28	9	9	9	11	14	{ 1 3 9 11 17 }
143	28	11	11	11	11	14	{ 1 3 9 11 25 }
144	28	9	9	9	9	9	{ 1 3 9 13 17 }
145	28	11	11	11	11	15	{ 1 3 9 13 25 }
146	28	11	11	11	11	15	{ 1 3 9 17 25 }
147	28	13	13	13	13	15	{ 1 3 11 13 17 }
148	28	11	11	11	11	13	{ 1 3 11 13 25 }
149	28	9	9	9	9	9	{ 1 3 11 17 25 }
150	28	11	11	11	11	15	{ 1 3 13 17 25 }
151	28	11	11	11	11	13	{ 1 5 9 11 13 }
152	28	14	14	14	14	15	{ 1 5 9 11 17 }
153	28	9	9	9	9	9	{ 1 5 9 11 25 }
154	28	14	14	14	14	15	{ 1 5 9 13 17 }
155	28	9	9	9	11	14	{ 1 5 9 13 25 }
156	28	11	11	11	11	13	{ 1 5 9 17 25 }
157	28	11	11	11	11	15	{ 1 5 11 13 17 }
158	28	9	9	9	11	15	{ 1 5 11 13 25 }
159	28	13	13	13	13	15	{ 1 5 11 17 25 }
160	28	9	9	9	9	9	{ 1 5 13 17 25 }
161	28	11	11	11	11	13	{ 1 9 11 13 17 }
162	28	14	14	14	14	15	{ 1 9 11 13 25 }
163	28	11	11	11	11	16	{ 1 9 11 17 25 }
164	28	13	13	13	13	15	{ 1 9 13 17 25 }
165	28	11	11	11	11	16	{ 1 11 13 17 25 }
166	28	11	11	11	11	16	{ 3 5 9 11 13 }
167	28	9	9	9	9	9	{ 3 5 9 11 17 }
168	28	11	11	11	11	14	{ 3 5 9 11 25 }
169	28	13	13	13	13	15	{ 3 5 9 13 17 }
170	28	9	9	9	9	9	{ 3 5 9 13 25 }
171	28	14	14	14	14	15	{ 3 5 9 17 25 }
172	28	11	11	11	11	14	{ 3 5 11 13 17 }
173	28	14	14	14	14	15	{ 3 5 11 13 25 }
174	28	11	11	11	11	13	{ 3 5 11 17 25 }
175	28	11	11	11	11	13	{ 3 5 13 17 25 }
176	28	9	9	9	11	14	{ 3 9 11 13 17 }
177	28	11	11	11	11	13	{ 3 9 11 13 25 }
178	28	13	13	13	13	15	{ 3 9 11 17 25 }
179	28	9	9	9	11	15	{ 3 9 13 17 25 }
180	28	14	14	14	14	15	{ 3 11 13 17 25 }
181	28	11	11	11	11	15	{ 5 9 11 13 17 }
182	28	13	13	13	13	15	{ 5 9 11 13 25 }
183	28	11	11	11	11	16	{ 5 9 11 17 25 }
184	28	9	9	9	10	14	{ 5 9 13 17 25 }
185	28	11	11	11	11	15	{ 5 11 13 17 25 }
186	28	9	9	9	9	9	{ 9 11 13 17 25 }
187	36	14	14	14	14	14	{ 0 1 3 5 9 }
188	36	10	10	10	10	14	{ 0 1 3 5 11 }

189	36	10	10	10	10	12	{ 0 1 3 5 13 }
190	36	10	10	10	10	10	{ 0 1 3 5 17 }
191	36	10	10	10	10	10	{ 0 1 3 5 25 }
192	36	10	10	10	10	10	{ 0 1 3 9 11 }
193	36	8	8	8	8	14	{ 0 1 3 9 13 }
194	36	8	8	8	9	12	{ 0 1 3 9 17 }
195	36	10	10	10	10	10	{ 0 1 3 9 25 }
196	36	8	8	8	9	12	{ 0 1 3 11 13 }
197	36	10	10	10	10	14	{ 0 1 3 11 17 }
198	36	8	8	8	9	12	{ 0 1 3 11 25 }
199	36	10	10	10	10	12	{ 0 1 3 13 17 }
200	36	14	14	14	14	14	{ 0 1 3 13 25 }
201	36	8	8	8	8	14	{ 0 1 3 17 25 }
202	36	8	8	8	9	12	{ 0 1 5 9 11 }
203	36	10	10	10	10	10	{ 0 1 5 9 13 }
204	36	8	9	9	11	12	{ 0 1 5 9 17 }
205	36	8	8	8	9	12	{ 0 1 5 9 25 }
206	36	8	8	8	9	12	{ 0 1 5 11 13 }
207	36	10	10	10	10	10	{ 0 1 5 11 17 }
208	36	10	10	10	10	14	{ 0 1 5 11 25 }
209	36	8	8	8	8	14	{ 0 1 5 13 17 }
210	36	10	10	10	10	13	{ 0 1 5 13 25 }
211	36	8	8	8	9	12	{ 0 1 5 17 25 }
212	36	8	9	9	10	12	{ 0 1 9 11 13 }
213	36	10	10	10	10	10	{ 0 1 9 11 17 }
214	36	8	8	8	8	14	{ 0 1 9 11 25 }
215	36	8	8	8	9	12	{ 0 1 9 13 17 }
216	36	10	10	10	10	10	{ 0 1 9 13 25 }
217	36	14	14	14	14	14	{ 0 1 9 17 25 }
218	36	14	14	14	14	14	{ 0 1 11 13 17 }
219	36	10	10	10	10	10	{ 0 1 11 13 25 }
220	36	10	10	10	10	12	{ 0 1 11 17 25 }
221	36	10	10	10	10	12	{ 0 1 13 17 25 }
222	36	10	10	10	10	11	{ 0 3 5 9 11 }
223	36	10	10	10	10	12	{ 0 3 5 9 13 }
224	36	8	8	8	9	12	{ 0 3 5 9 17 }
225	36	8	8	8	8	14	{ 0 3 5 9 25 }
226	36	8	8	8	8	14	{ 0 3 5 11 13 }
227	36	8	8	8	9	12	{ 0 3 5 11 17 }
228	36	10	10	10	10	10	{ 0 3 5 11 25 }
229	36	14	14	14	14	14	{ 0 3 5 13 17 }
230	36	8	8	8	9	12	{ 0 3 5 13 25 }
231	36	8	9	9	11	12	{ 0 3 5 17 25 }
232	36	10	10	10	10	10	{ 0 3 9 11 13 }
233	36	10	10	10	10	14	{ 0 3 9 11 17 }
234	36	14	14	14	14	14	{ 0 3 9 11 25 }
235	36	10	10	10	10	14	{ 0 3 9 13 17 }
236	36	8	8	8	9	12	{ 0 3 9 13 25 }
237	36	10	10	10	10	10	{ 0 3 9 17 25 }
238	36	10	10	10	10	10	{ 0 3 11 13 17 }
239	36	8	9	9	11	12	{ 0 3 11 13 25 }
240	36	8	8	8	9	12	{ 0 3 11 17 25 }
241	36	10	10	10	10	10	{ 0 3 13 17 25 }
242	36	14	14	14	14	14	{ 0 5 9 11 13 }
243	36	8	8	8	8	14	{ 0 5 9 11 17 }
244	36	10	10	10	10	12	{ 0 5 9 11 25 }
245	36	10	10	10	10	10	{ 0 5 9 13 17 }
246	36	10	10	10	10	14	{ 0 5 9 13 25 }
247	36	10	10	10	10	10	{ 0 5 9 17 25 }
248	36	10	10	10	10	10	{ 0 5 11 13 17 }
249	36	10	10	10	10	10	{ 0 5 11 13 25 }
250	36	14	14	14	14	14	{ 0 5 11 17 25 }
251	36	8	8	8	9	12	{ 0 5 13 17 25 }
252	36	8	8	8	9	12	{ 0 9 11 13 17 }

253	36	8	8	8	9	12	{ 0 9 11 13 25 }
254	36	10	10	10	10	12	{ 0 9 11 17 25 }
255	36	10	10	10	10	13	{ 0 9 13 17 25 }
256	36	8	8	8	8	14	{ 0 11 13 17 25 }
257	37	11	11	11	11	11	{ 1 3 5 9 }
258	37	9	9	9	9	14	{ 1 3 5 11 }
259	37	9	9	9	9	11	{ 1 3 5 13 }
260	37	9	9	9	9	10	{ 1 3 5 17 }
261	37	9	9	9	9	9	{ 1 3 5 25 }
262	37	9	9	9	9	9	{ 1 3 9 11 }
263	37	7	8	8	8	13	{ 1 3 9 13 }
264	37	7	7	7	9	10	{ 1 3 9 17 }
265	37	5	7	7	7	10	{ 1 3 9 25 }
266	37	8	8	8	9	10	{ 1 3 11 13 }
267	37	9	9	9	9	13	{ 1 3 11 17 }
268	37	7	7	7	9	10	{ 1 3 11 25 }
269	37	9	9	9	9	11	{ 1 3 13 17 }
270	37	11	11	11	11	11	{ 1 3 13 25 }
271	37	7	8	8	8	13	{ 1 3 17 25 }
272	37	8	8	8	9	10	{ 1 5 9 11 }
273	37	9	9	9	9	11	{ 1 5 9 13 }
274	37	8	9	9	11	11	{ 1 5 9 17 }
275	37	7	7	7	9	10	{ 1 5 9 25 }
276	37	7	7	7	9	10	{ 1 5 11 13 }
277	37	9	9	9	9	9	{ 1 5 11 17 }
278	37	9	9	9	9	13	{ 1 5 11 25 }
279	37	7	8	8	8	13	{ 1 5 13 17 }
280	37	9	9	9	9	13	{ 1 5 13 25 }
281	37	8	8	8	9	10	{ 1 5 17 25 }
282	37	8	9	9	10	11	{ 1 9 11 13 }
283	37	9	9	9	9	10	{ 1 9 11 17 }
284	37	7	8	8	8	13	{ 1 9 11 25 }
285	37	8	8	8	9	10	{ 1 9 13 17 }
286	37	9	9	9	9	9	{ 1 9 13 25 }
287	37	11	11	11	11	11	{ 1 9 17 25 }
288	37	11	11	11	11	11	{ 1 11 13 17 }
289	37	9	9	9	9	10	{ 1 11 13 25 }
290	37	9	9	9	9	11	{ 1 11 17 25 }
291	37	9	9	9	9	11	{ 1 13 17 25 }
292	37	9	9	9	9	11	{ 3 5 9 11 }
293	37	9	9	9	9	11	{ 3 5 9 13 }
294	37	8	8	8	9	10	{ 3 5 9 17 }
295	37	7	8	8	8	13	{ 3 5 9 25 }
296	37	7	8	8	8	13	{ 3 5 11 13 }
297	37	7	7	7	9	11	{ 3 5 11 17 }
298	37	9	9	9	9	10	{ 3 5 11 25 }
299	37	11	11	11	11	11	{ 3 5 13 17 }
300	37	8	8	8	9	11	{ 3 5 13 25 }
301	37	8	9	9	11	11	{ 3 5 17 25 }
302	37	9	9	9	9	10	{ 3 9 11 13 }
303	37	9	9	9	9	14	{ 3 9 11 17 }
304	37	11	11	11	11	11	{ 3 9 11 25 }
305	37	9	9	9	9	14	{ 3 9 13 17 }
306	37	7	7	7	9	10	{ 3 9 13 25 }
307	37	9	9	9	9	9	{ 3 9 17 25 }
308	37	9	9	9	9	10	{ 3 11 13 17 }
309	37	8	9	9	11	11	{ 3 11 13 25 }
310	37	8	8	8	9	10	{ 3 11 17 25 }
311	37	9	9	9	9	10	{ 3 13 17 25 }
312	37	11	11	11	11	11	{ 5 9 11 13 }
313	37	7	8	8	8	13	{ 5 9 11 17 }
314	37	9	9	9	9	11	{ 5 9 11 25 }
315	37	9	9	9	9	9	{ 5 9 13 17 }
316	37	9	9	9	9	13	{ 5 9 13 25 }

317	37	9	9	9	9	10	{ 5 9 17 25 }
318	37	5	7	7	7	10	{ 5 11 13 17 }
319	37	9	9	9	9	9	{ 5 11 13 25 }
320	37	11	11	11	11	11	{ 5 11 17 25 }
321	37	7	7	7	9	10	{ 5 13 17 25 }
322	37	7	7	7	9	10	{ 9 11 13 17 }
323	37	8	8	8	9	10	{ 9 11 13 25 }
324	37	9	9	9	9	11	{ 9 11 17 25 }
325	37	9	9	9	9	14	{ 9 13 17 25 }
326	37	7	8	8	8	13	{ 11 13 17 25 }
327	45	10	10	10	10	10	{ 0 1 3 5 }
328	45	8	8	8	8	8	{ 0 1 3 9 }
329	45	6	6	6	7	9	{ 0 1 3 11 }
330	45	6	6	6	7	8	{ 0 1 3 13 }
331	45	7	7	7	7	10	{ 0 1 3 17 }
332	45	8	8	8	8	8	{ 0 1 3 25 }
333	45	6	7	7	8	10	{ 0 1 5 9 }
334	45	6	6	6	7	9	{ 0 1 5 11 }
335	45	7	7	7	7	10	{ 0 1 5 13 }
336	45	6	6	6	6	8	{ 0 1 5 17 }
337	45	6	6	6	7	9	{ 0 1 5 25 }
338	45	6	6	6	6	8	{ 0 1 9 11 }
339	45	6	6	6	6	8	{ 0 1 9 13 }
340	45	6	7	7	8	10	{ 0 1 9 17 }
341	45	8	8	8	8	8	{ 0 1 9 25 }
342	45	6	7	7	8	10	{ 0 1 11 13 }
343	45	10	10	10	10	9	{ 0 1 11 17 }
344	45	7	7	7	7	10	{ 0 1 11 25 }
345	45	6	6	6	7	8	{ 0 1 13 17 }
346	45	10	10	10	10	9	{ 0 1 13 25 }
347	45	6	6	6	7	8	{ 0 1 17 25 }
348	45	6	6	6	7	8	{ 0 3 5 9 }
349	45	7	7	7	7	10	{ 0 3 5 11 }
350	45	6	6	6	7	8	{ 0 3 5 13 }
351	45	6	7	7	8	10	{ 0 3 5 17 }
352	45	6	6	6	6	8	{ 0 3 5 25 }
353	45	10	10	10	10	10	{ 0 3 9 11 }
354	45	7	7	7	7	10	{ 0 3 9 13 }
355	45	6	6	6	7	9	{ 0 3 9 17 }
356	45	8	8	8	8	8	{ 0 3 9 25 }
357	45	6	6	6	6	8	{ 0 3 11 13 }
358	45	6	6	6	7	9	{ 0 3 11 17 }
359	45	6	7	7	8	10	{ 0 3 11 25 }
360	45	10	10	10	10	10	{ 0 3 13 17 }
361	45	6	7	7	8	10	{ 0 3 13 25 }
362	45	6	6	6	6	8	{ 0 3 17 25 }
363	45	6	6	6	7	8	{ 0 5 9 11 }
364	45	10	10	10	10	10	{ 0 5 9 13 }
365	45	6	6	6	6	8	{ 0 5 9 17 }
366	45	7	7	7	7	10	{ 0 5 9 25 }
367	45	8	8	8	8	8	{ 0 5 11 13 }
368	45	8	8	8	8	8	{ 0 5 11 17 }
369	45	10	10	10	10	9	{ 0 5 11 25 }
370	45	8	8	8	8	8	{ 0 5 13 17 }
371	45	6	6	6	7	9	{ 0 5 13 25 }
372	45	6	7	7	8	10	{ 0 5 17 25 }
373	45	6	7	7	8	10	{ 0 9 11 13 }
374	45	7	7	7	7	10	{ 0 9 11 17 }
375	45	6	6	6	7	8	{ 0 9 11 25 }
376	45	6	6	6	7	9	{ 0 9 13 17 }
377	45	6	6	6	7	9	{ 0 9 13 25 }
378	45	10	10	10	10	10	{ 0 9 17 25 }
379	45	8	8	8	8	8	{ 0 11 13 17 }
380	45	6	6	6	6	8	{ 0 11 13 25 }

381	45	6	6	6	7	8	{ 0 11 17 25 }
382	45	7	7	7	7	10	{ 0 13 17 25 }
383	46	9	9	9	9	9	{ 1 3 5 }
384	46	5	5	5	7	8	{ 1 3 9 }
385	46	6	6	6	7	8	{ 1 3 11 }
386	46	5	5	5	6	9	{ 1 3 13 }
387	46	7	7	7	7	9	{ 1 3 17 }
388	46	5	5	5	7	8	{ 1 3 25 }
389	46	6	7	7	8	9	{ 1 5 9 }
390	46	6	6	6	7	8	{ 1 5 11 }
391	46	7	7	7	7	9	{ 1 5 13 }
392	46	5	5	5	6	8	{ 1 5 17 }
393	46	6	6	6	7	8	{ 1 5 25 }
394	46	5	5	5	6	8	{ 1 9 11 }
395	46	5	5	5	6	8	{ 1 9 13 }
396	46	6	7	7	8	9	{ 1 9 17 }
397	46	5	5	5	7	8	{ 1 9 25 }
398	46	6	7	7	8	9	{ 1 11 13 }
399	46	9	9	9	9	9	{ 1 11 17 }
400	46	7	7	7	7	9	{ 1 11 25 }
401	46	5	5	5	6	9	{ 1 13 17 }
402	46	9	9	9	9	9	{ 1 13 25 }
403	46	5	5	5	6	9	{ 1 17 25 }
404	46	5	5	5	6	9	{ 3 5 9 }
405	46	7	7	7	7	9	{ 3 5 11 }
406	46	5	5	5	6	9	{ 3 5 13 }
407	46	6	7	7	8	9	{ 3 5 17 }
408	46	5	5	5	6	8	{ 3 5 25 }
409	46	9	9	9	9	9	{ 3 9 11 }
410	46	7	7	7	7	9	{ 3 9 13 }
411	46	6	6	6	7	8	{ 3 9 17 }
412	46	5	5	5	7	8	{ 3 9 25 }
413	46	5	5	5	6	8	{ 3 11 13 }
414	46	6	6	6	7	8	{ 3 11 17 }
415	46	6	7	7	8	9	{ 3 11 25 }
416	46	9	9	9	9	9	{ 3 13 17 }
417	46	6	7	7	8	9	{ 3 13 25 }
418	46	5	5	5	6	8	{ 3 17 25 }
419	46	5	5	5	6	9	{ 5 9 11 }
420	46	9	9	9	9	9	{ 5 9 13 }
421	46	5	5	5	6	8	{ 5 9 17 }
422	46	7	7	7	7	9	{ 5 9 25 }
423	46	5	5	5	7	8	{ 5 11 13 }
424	46	5	5	5	7	8	{ 5 11 17 }
425	46	9	9	9	9	9	{ 5 11 25 }
426	46	5	5	5	7	8	{ 5 13 17 }
427	46	6	6	6	7	8	{ 5 13 25 }
428	46	6	7	7	8	9	{ 5 17 25 }
429	46	6	7	7	8	9	{ 9 11 13 }
430	46	7	7	7	7	9	{ 9 11 17 }
431	46	5	5	5	6	9	{ 9 11 25 }
432	46	6	6	6	7	8	{ 9 13 17 }
433	46	6	6	6	7	8	{ 9 13 25 }
434	46	9	9	9	9	9	{ 9 17 25 }
435	46	5	5	5	7	8	{ 11 13 17 }
436	46	5	5	5	6	8	{ 11 13 25 }
437	46	5	5	5	6	9	{ 11 17 25 }
438	46	7	7	7	7	9	{ 13 17 25 }
439	54	6	6	6	6	6	{ 0 1 3 }
440	54	5	5	5	6	6	{ 0 1 5 }
441	54	6	6	6	6	6	{ 0 1 9 }
442	54	5	5	5	6	6	{ 0 1 11 }
443	54	4	5	5	5	6	{ 0 1 13 }
444	54	4	5	5	5	6	{ 0 1 17 }

445	54	6	6	6	6	6	{ 0 1 25 }
446	54	4	5	5	5	6	{ 0 3 5 }
447	54	6	6	6	6	6	{ 0 3 9 }
448	54	5	5	5	6	6	{ 0 3 11 }
449	54	4	5	5	5	6	{ 0 3 13 }
450	54	5	5	5	6	6	{ 0 3 17 }
451	54	6	6	6	6	6	{ 0 3 25 }
452	54	4	5	5	5	6	{ 0 5 9 }
453	54	6	6	6	6	6	{ 0 5 11 }
454	54	6	6	6	6	6	{ 0 5 13 }
455	54	6	6	6	6	6	{ 0 5 17 }
456	54	5	5	5	6	6	{ 0 5 25 }
457	54	4	5	5	5	6	{ 0 9 11 }
458	54	5	5	5	6	6	{ 0 9 13 }
459	54	5	5	5	6	6	{ 0 9 17 }
460	54	6	6	6	6	6	{ 0 9 25 }
461	54	6	6	6	6	6	{ 0 11 13 }
462	54	6	6	6	6	6	{ 0 11 17 }
463	54	4	5	5	5	6	{ 0 11 25 }
464	54	6	6	6	6	6	{ 0 13 17 }
465	54	5	5	5	6	6	{ 0 13 25 }
466	54	4	5	5	5	6	{ 0 17 25 }
467	55	5	5	5	5	5	{ 1 3 }
468	55	5	5	5	6	6	{ 1 5 }
469	55	4	5	5	5	7	{ 1 9 }
470	55	5	5	5	6	6	{ 1 11 }
471	55	4	5	5	5	6	{ 1 13 }
472	55	4	5	5	5	6	{ 1 17 }
473	55	5	5	5	5	5	{ 1 25 }
474	55	4	5	5	5	6	{ 3 5 }
475	55	5	5	5	5	5	{ 3 9 }
476	55	5	5	5	6	6	{ 3 11 }
477	55	4	5	5	5	6	{ 3 13 }
478	55	5	5	5	6	6	{ 3 17 }
479	55	4	5	5	5	7	{ 3 25 }
480	55	4	5	5	5	6	{ 5 9 }
481	55	5	5	5	5	5	{ 5 11 }
482	55	5	5	5	5	5	{ 5 13 }
483	55	4	5	5	5	7	{ 5 17 }
484	55	5	5	5	6	6	{ 5 25 }
485	55	4	5	5	5	6	{ 9 11 }
486	55	5	5	5	6	6	{ 9 13 }
487	55	5	5	5	6	6	{ 9 17 }
488	55	5	5	5	5	5	{ 9 25 }
489	55	4	5	5	5	7	{ 11 13 }
490	55	5	5	5	5	5	{ 11 17 }
491	55	4	5	5	5	6	{ 11 25 }
492	55	5	5	5	5	5	{ 13 17 }
493	55	5	5	5	6	6	{ 13 25 }
494	55	4	5	5	5	6	{ 17 25 }
495	63	4	4	4	4	4	{ 0 1 }
496	63	4	4	4	4	4	{ 0 3 }
497	63	4	4	4	4	4	{ 0 5 }
498	63	4	4	4	4	4	{ 0 9 }
499	63	4	4	4	4	4	{ 0 11 }
500	63	4	4	4	4	4	{ 0 13 }
501	63	4	4	4	4	4	{ 0 17 }
502	63	4	4	4	4	4	{ 0 25 }
503	64	3	3	3	3	3	{ 1 }
504	64	3	3	3	3	3	{ 3 }
505	64	3	3	3	3	3	{ 5 }
506	64	3	3	3	3	3	{ 9 }
507	64	3	3	3	3	3	{ 11 }
508	64	3	3	3	3	3	{ 13 }

509	64	3	3	3	3	3	{ 17 }
510	64	3	3	3	3	3	{ 25 }
511	64	3	3	3	3	3	{ 25 }
512	72	2	2	2	2	2	{ 0 }

#	k	d_shift	d_brouwer
1	0	74	
2	1	73	73
3	9	28	32
4	9	28	
5	9	28	
6	9	28	
7	9	28	
8	9	28	
9	9	28	
10	9	28	
11	10	28	32
12	10	28	
13	10	28	
14	10	28	
15	10	28	
16	10	28	
17	10	28	
18	10	28	
19	18	24	24-27
20	18	22	
21	18	21	
22	18	23	
23	18	22	
24	18	21	
25	18	22	
26	18	22	
27	18	22	
28	18	24	
29	18	22	
30	18	22	
31	18	21	
32	18	22	
33	18	23	
34	18	22	
35	18	22	
36	18	24	
37	18	21	
38	18	21	
39	18	24	
40	18	22	
41	18	24	
42	18	24	
43	18	22	
44	18	22	
45	18	22	
46	18	21	
47	19	21	24-26
48	19	21	
49	19	21	
50	19	21	
51	19	21	
52	19	19	
53	19	21	

54	19	21	
55	19	21	
56	19	21	
57	19	20	
58	19	19	
59	19	21	
60	19	21	
61	19	21	
62	19	19	
63	19	20	
64	19	21	
65	19	20	
66	19	21	
67	19	21	
68	19	21	
69	19	21	
70	19	21	
71	19	21	
72	19	19	
73	19	19	
74	19	21	
75	27	18	20-22
76	27	16	
77	27	16	
78	27	16	
79	27	16	
80	27	15	
81	27	15	
82	27	16	
83	27	18	
84	27	15	
85	27	15	
86	27	15	
87	27	16	
88	27	16	
89	27	16	
90	27	16	
91	27	18	
92	27	16	
93	27	16	
94	27	15	
95	27	15	
96	27	14	
97	27	16	
98	27	15	
99	27	16	
100	27	15	
101	27	16	
102	27	16	
103	27	18	
104	27	16	
105	27	16	
106	27	15	
107	27	15	
108	27	18	
109	27	16	
110	27	16	
111	27	16	
112	27	16	
113	27	18	
114	27	16	
115	27	14	
116	27	16	

117	27	15	
118	27	16	
119	27	15	
120	27	15	
121	27	16	
122	27	18	
123	27	15	
124	27	14	
125	27	16	
126	27	18	
127	27	16	
128	27	16	
129	27	16	
130	27	16	
131	28	15	18-22
132	28	16	
133	28	13	
134	28	15	
135	28	9	
136	28	14	
137	28	15	
138	28	15	
139	28	15	
140	28	15	
141	28	15	
142	28	14	
143	28	15	
144	28	9	
145	28	14	
146	28	15	
147	28	15	
148	28	13	
149	28	9	
150	28	15	
151	28	13	
152	28	14	
153	28	9	
154	28	15	
155	28	14	
156	28	13	
157	28	14	
158	28	15	
159	28	15	
160	28	9	
161	28	13	
162	28	15	
163	28	15	
164	28	15	
165	28	15	
166	28	16	
167	28	9	
168	28	14	
169	28	15	
170	28	9	
171	28	14	
172	28	14	
173	28	15	
174	28	13	
175	28	13	
176	28	15	
177	28	13	
178	28	15	
179	28	15	

180	28	14	
181	28	15	
182	28	15	
183	28	16	
184	28	14	
185	28	15	
186	28	9	
187	36	14	16-18
188	36	14	
189	36	12	
190	36	10	
191	36	10	
192	36	10	
193	36	14	
194	36	12	
195	36	10	
196	36	12	
197	36	14	
198	36	12	
199	36	12	
200	36	14	
201	36	14	
202	36	12	
203	36	10	
204	36	12	
205	36	12	
206	36	12	
207	36	10	
208	36	14	
209	36	14	
210	36	13	
211	36	12	
212	36	12	
213	36	10	
214	36	14	
215	36	12	
216	36	10	
217	36	14	
218	36	14	
219	36	10	
220	36	12	
221	36	12	
222	36	12	
223	36	11	
224	36	12	
225	36	14	
226	36	14	
227	36	12	
228	36	10	
229	36	14	
230	36	12	
231	36	12	
232	36	10	
233	36	14	
234	36	14	
235	36	14	
236	36	12	
237	36	10	
238	36	10	
239	36	12	
240	36	12	
241	36	10	
242	36	14	

243	36	14	
244	36	12	
245	36	10	
246	36	14	
247	36	10	
248	36	10	
249	36	10	
250	36	14	
251	36	12	
252	36	12	
253	36	12	
254	36	12	
255	36	14	
256	36	14	
257	37	11	14-17
258	37	9	
259	37	9	
260	37	10	
261	37	9	
262	37	9	
263	37	9	
264	37	9	
265	37	10	
266	37	9	
267	37	9	
268	37	9	
269	37	9	
270	37	11	
271	37	9	
272	37	9	
273	37	10	
274	37	11	
275	37	9	
276	37	9	
277	37	9	
278	37	9	
279	37	9	
280	37	9	
281	37	9	
282	37	11	
283	37	10	
284	37	9	
285	37	9	
286	37	9	
287	37	11	
288	37	11	
289	37	10	
290	37	9	
291	37	9	
292	37	9	
293	37	9	
294	37	9	
295	37	9	
296	37	9	
297	37	9	
298	37	10	
299	37	11	
300	37	9	
301	37	11	
302	37	10	
303	37	9	
304	37	11	
305	37	9	

306	37	9	
307	37	9	
308	37	9	
309	37	11	
310	37	9	
311	37	10	
312	37	11	
313	37	9	
314	37	9	
315	37	9	
316	37	9	
317	37	10	
318	37	10	
319	37	9	
320	37	11	
321	37	9	
322	37	9	
323	37	9	
324	37	9	
325	37	9	
326	37	9	
327	45	9	10-13
328	45	8	
329	45	9	
330	45	8	
331	45	10	
332	45	8	
333	45	10	
334	45	9	
335	45	10	
336	45	8	
337	45	9	
338	45	8	
339	45	8	
340	45	10	
341	45	8	
342	45	10	
343	45	9	
344	45	10	
345	45	8	
346	45	9	
347	45	8	
348	45	8	
349	45	10	
350	45	8	
351	45	10	
352	45	8	
353	45	10	
354	45	10	
355	45	9	
356	45	8	
357	45	8	
358	45	9	
359	45	10	
360	45	10	
361	45	10	
362	45	8	
363	45	8	
364	45	10	
365	45	8	
366	45	10	
367	45	8	
368	45	8	

369	45	9	
370	45	8	
371	45	9	
372	45	10	
373	45	10	
374	45	10	
375	45	8	
376	45	9	
377	45	9	
378	45	10	
379	45	8	
380	45	8	
381	45	8	
382	45	10	

383	46	9	10-12
384	46	8	
385	46	8	
386	46	8	
387	46	9	
388	46	8	
389	46	9	
390	46	8	
391	46	9	
392	46	8	
393	46	8	
394	46	8	
395	46	8	
396	46	9	
397	46	8	
398	46	9	
399	46	9	
400	46	9	
401	46	8	
402	46	9	
403	46	8	
404	46	8	
405	46	9	
406	46	8	
407	46	9	
408	46	8	
409	46	9	
410	46	9	
411	46	8	
412	46	8	
413	46	8	
414	46	8	
415	46	9	
416	46	9	
417	46	9	
418	46	8	
419	46	8	
420	46	9	
421	46	8	
422	46	9	
423	46	8	
424	46	8	
425	46	9	
426	46	8	
427	46	8	
428	46	9	
429	46	9	
430	46	9	
431	46	8	

432	46	8	
433	46	8	
434	46	9	
435	46	8	
436	46	8	
437	46	8	
438	46	9	

439	54	6	7-8
440	54	6	
441	54	6	
442	54	6	
443	54	6	
444	54	6	
445	54	6	
446	54	6	
447	54	6	
448	54	6	
449	54	6	
450	54	6	
451	54	6	
452	54	6	
453	54	6	
454	54	5	
455	54	6	
456	54	6	
457	54	6	
458	54	6	
459	54	6	
460	54	5	
461	54	6	
462	54	6	
463	54	6	
464	54	5	
465	54	6	
466	54	6	

467	55	5	6-8
468	55	6	
469	55	6	
470	55	6	
471	55	6	
472	55	6	
473	55	5	
474	55	6	
475	55	5	
476	55	6	
477	55	6	
478	55	6	
479	55	6	
480	55	6	
481	55	5	
482	55	5	
483	55	6	
484	55	6	
485	55	6	
486	55	6	
487	55	6	
488	55	5	
489	55	6	
490	55	5	
491	55	6	
492	55	5	
493	55	6	

494	55	6	
495	63	4	4
496	63	4	
497	63	4	
498	63	4	
499	63	4	
500	63	4	
501	63	4	
502	63	4	
503	64	3	4
504	64	3	
505	64	3	
506	64	3	
507	64	3	
508	64	3	
509	64	3	
510	64	3	
511	64	3	
512	72	2	2

Remark 7.2. For $n = 73$, there is no improvements on the Brouwer's table. In fact, most of cases are equal to the lower bound on Brouwer's table.

Bibliography

[1] A.E. Brouwer. *Handbook of Coding Theory*, volume 1, chapter Bounds on the size of linear codes, pages 295–461. Elsevier Science B.V., Amsterdam, 1998. 79

[2] T. Budd. *Data Structures in C++ using the STL*, chapter 11. Addison-Wesley, Oregon State University, 1997. budd@cs.orst.edu.

[3] P.J. Cameron. *Combinatorics : Topics, Techniques, Algorithms*. Cambridge University Press, 2nd edition, 2001.

[4] R. Diestel. *Graph Theory*. Springer-Verlag Heidelberg, New York, 3rd edition, 2005. 51

[5] A. Drozdek. *Data structures and algorithms in C++*. Thomson:Course Technology, Boston, 5th edition, 2005.

[6] A.M.H. Gerards. Graphs and algorithms. Lecture notes on Graphs and Algorithms, 2006. 51

[7] E.M. Gurari. Course note : Introduction to data structures. http://www.cse.ohio-state.edu/ gurari/course/cis680/cis680.html, Autumn 1999. 50

[8] C.R.P. Hartmann and K.K. Tzeng. Generalizations of the bch bound. *Information and Control*, 20:489–498, 1972. Academic Press Inc. 26, 28

[9] T. Kasami. The weight enumerators for several classes of subcodes of the 2nd order binary reed-muller codes. *Information and Control*, 18:369–394, 1971. 41

[10] Donald E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*. Addison-Wesley, 3rd edition, 1997. 51

[11] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*, volume 16. North-Holland Publishing Company, Elvesier/North-Holland Inc., 52 Vanderbilt Avenue, New York, NY 10017, 3rd edition, 1996. 3, 8, 11, 13, 14, 22, 25, 69

[12] G.R. Pellikaan. *Arithmetic, Geometry and Coding Theory*, volume 4, chapter The shift bound for cyclic, Reed-Muller and geometric Goppa codes, pages 155–174. Luminy, Berlin, 1993,1996. 44

[13] G.R. Pellikaan. The construction and decoding of algebraic codes. Lecture Notes, March 2004. 3, 11, 12, 13, 14, 21, 25

[14] W.C. Huffman R.A. Brualdi and V.S. Pless. *Handbook of Coding Theory*, volume 1, chapter An introduction to algebraic codes, pages 3–139. Elsevier Science B.V., Amsterdam, 1998.

[15] C. Roos. A generalization of the bch bound for cyclic codes, including the hartmann-tzeng bound. *Journal of Combinatorial Theory*, Series A(33):229–232, 1982. Academic Press. 30, 37

[16] C. Roos. A new lower bound for the minimum distance of a cyclic code. *IEEE Transaction on Information Theory*, IT-29(3), May 1983. 30, 33

[17] R.L. Rivest T.H. Cormen, C.E. Leiserson and C. Stein. *Introduction to Algorithms*. MIT-Press and McGraw-Hill, 2nd edition, 2001. 51

[18] M. van Eupen and J.H. van Lint. On the minimum distance of ternary cyclic codes. *IEEE Transactions on Information Theory*, 39(2), March 1993. 44

[19] J.H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, Berlin-Heidelberg, 3rd revised and expanded ed. edition, 1982-1992-1999. 3, 6, 8, 11, 25, 34

[20] J.H. van Lint and R.M. Wilson. On the minimum distance of cyclic codes. *IEEE Transactions on Information Theory*, IT-32(1), January 1986. 33, 34, 35, 41, 43, 48

[21] H.C.A. van Tilborg. Coding theory : a first course. Lecture Notes on Error-Correcting Codes. 3, 11, 13, 14, 21, 25