Universitat Autònoma de Barcelona

Partial permutation decoding for Z4-linear Hadamard and Kerdock codes

Roland D. Barrolleta

Universitat Autònoma de Barcelona

Departament d'Enginyeria de la Informació i de les Comunicacions

Partial permutation decoding for $\mathbb{Z}_4$-linear Hadamard and Kerdock codes

Submitted to Universitat Autònoma de Barcelona in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

by Roland D. Barrolleta

Cerdanyola del Vallès, September 2016

Advisor: Dr. Mercè Villanueva

Professor at Universitat Autònoma de Barcelona

I certify that I have read this thesis entitled "Partial permutation decoding for $\mathbb{Z}_4$-linear Hadamard and Kerdock codes" and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Cerdanyola del Vallès, September 2016

Dr. Mercè Villanueva
(Advisor)

A todos aquellos que
confiaron en mí

Abstract

The hardest problem in the process of transmitting information is decoding. One of the major areas of research in coding theory is to find efficient decoding algorithms. Permutation decoding is a technique that strongly depends on the existence of a special subset, called a PD-set, of the permutation automorphism group of a code to assist in decoding received vectors. Recently, a new permutation decoding method suitable for $\mathbb{Z}_4$-linear codes was introduced. This dissertation aims to provide s-PD-sets, which enable the correction of up to s errors, for some families of $\mathbb{Z}_4$-linear codes to perform permutation decoding. We give upper bounds on s for which s-PD-sets of minimum size $s+1$ for systematic Hadamard and Kerdock codes can exist. Two different criteria to find s-PD-sets of size $s+1$ for $\mathbb{Z}_4$-linear codes are provided. Explicit and recursive constructions of s-PD-sets of size $s+1$ for binary linear and $\mathbb{Z}_4$-linear Hadamard codes fulfilling the first criterion are presented. Likewise, we show explicit constructions of s-PD-sets of size $s+1$ for $\mathbb{Z}_4$-linear Hadamard and Kerdock codes satisfying the second criterion. Finally, new MAGMA functions to deal with permutation decoding for linear codes over finite fields and $\mathbb{Z}_4$-linear codes, among other suitable decoding methods for $\mathbb{Z}_4$-linear codes, have been developed based on the results given in this dissertation.

Resum

El problema més difícil en el procés de transmetre informació és la descodificació. Una de les àrees més importants d'investigació dins de la teoria de la codificació és la recerca d'algoritmes eficients de descodificació. La descodificació per permutacions és una tècnica que depèn totalment de l'existència d'un subconjunt especial, anomenat PD-conjunt, del grup d'automorfismes d'un codi per contribuir a la descodificació dels vectors rebuts. Recentment, s'ha introduït un nou mètode de descodificació per permutacions idoni per a codis $\mathbb{Z}_4$-lineals. L'objectiu principal d'aquesta tesis és proporcionar s-PD-conjunts, els quals permeten la correcció de fins a s errors, per a algunes famílies de codis $\mathbb{Z}_4$-lineals per a l'aplicació de la descodificació per permutacions. Es donen fites superiors sobre els valors de s per als quals poden existir s-PD-conjunts de mida mínima $s+1$ per a codis de Hadamard sistemàtics i codis de Kerdock. Es proporcionen dos criteris diferents per trobar s-PD-conjunts de mida $s+1$ per a codis $\mathbb{Z}_4$-lineals. Es presenten construccions explícites i recursives de s-PD-conjunts de mida $s+1$ per a codis binaris lineals de Hadamard i per a codis $\mathbb{Z}_4$-lineals de Hadamard, que compleixen el primer criteri. Així mateix, es mostren construccions explícites de s-PD-conjunts de mida $s+1$ per a codis $\mathbb{Z}_4$-lineals de Hadamard i de Kerdock que satisfan el segon criteri. Finalment, s'han desenvolupat noves funcions en MAGMA, basades en els resultats obtinguts en aquesta tesis, per treballar amb la descodificació per permutacions per a codis lineals sobre cossos finits i codis $\mathbb{Z}_4$-lineals. De la mateixa forma, s'han desenvolupat noves funcions per treballar amb altres mètodes de descodificació adequats per a codis $\mathbb{Z}_4$-lineals.

Acknowledgements

En primer lugar me gustaría expresar mis más sinceros agradecimientos a mi directora, Mercè Villanueva, por todo su apoyo y consejos a lo largo de la realización de esta tesis doctoral.

I would like to express my gratitude to Prof. Leo Storme for his unconditional care and for all valuable talks and helpful advices he gave me during my stay at Ghent University.

Me gustaría dar las gracias de manera especial a Evelia y a Margarita, por haber compartido su pasión por el álgebra conmigo.

Durante estos tres años en el dEIC, he madurado no solo como matemático sino también como persona. Esta etapa no habría sido lo mismo sin el resto de doctorandos del departamento, ya amigos muchos de ellos, y todas las experiencias compartidas con ellos.

Finalmente, quiero agradecer a mis amigos y a mi familia por estar siempre ahí, por entender mis ausencias, por levantarme el ánimo en los malos momentos.

Contents

Chapter 1

Introduction

Coding theory is the study of methods for the efficient and accurate transfer of information from one place to another. The physical medium through which the information is transmitted is called a *channel*. A channel is said to be *noisy* if the received information can be different from the information that was sent. Coding theory deals with the problem of detecting and correcting transmission errors caused by the noise of the channel. Figure 1.1 provides a rough idea of a communication channel.

According to this scheme, a message, denoted by m, generated by the source is first encoded. This process results in a codeword, denoted by x, which is sent over the channel, where noise in the form of a vector, denoted by e, may distort the codeword producing a received vector y. The received vector y is then decoded, obtaining an estimate $\hat{m}$ of the message m. Hopefully, $\hat{m}$ and the original transmitted message m agree. The process of correcting errors and retrieving the message is called decoding. Since there is a one-to-one correspondence between codewords and messages, decoding for us is to obtain an estimate $\hat{y}$ of the received vector y for which $\hat{y} = x$ is expected. From our perspective, the most important part of this scheme is the noise, for without it there would be no need for the development of coding theory.

Traditionally, codewords are n-tuples over the ring $\mathbb{Z}_2$. Therefore, we can consider a code as a subset of $\mathbb{Z}_2^n$ that contains all codewords. Linear codes, that is, codes where the sum of any two codewords is a codeword, are the most commonly used and studied codes as they are easier to describe, encode and decode than nonlinear codes. For any linear code there is a generator matrix comprising a subset of codewords of the code that allows us to generate the whole code without needing to store each of the codewords. Nevertheless, it turns out that this approach is not enough to fully cover all "good" codes:

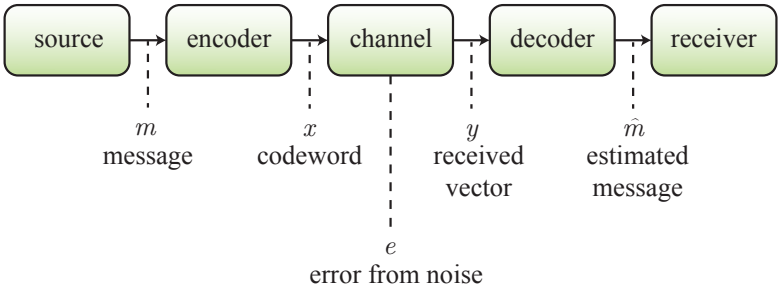

Figure 1.1: Communication channel

there exist some binary nonlinear codes with good properties. More specifically, there exist binary nonlinear codes that have more codewords than the best linear codes with the same length and minimum distance. This is the case, for example, of Preparata and Kerdock codes.

The influential paper [HKC+94] completely transformed the study of nonlinear codes since it was shown that several well-known families of nonlinear binary codes, including Preparata and Kerdock codes, can be simply constructed as binary images under the Gray map of linear codes over $\mathbb{Z}_4$. The term $\mathbb{Z}_4$-linear code is used to denote such a binary code with an algebraic structure over $\mathbb{Z}_4$, whereas the term quaternary linear code denotes a linear code over $\mathbb{Z}_4$.

The process of decoding, that is, finding which codeword x (and thus which message m) was sent when a vector y is received, is complex and usually becomes harder if the size of the code increases. One of the fundamental problems in coding theory is to find efficient decoding algorithms. Permutation decoding is a decoding method, developed by MacWilliams [Mac64] and Prange [Pra62], which uses a subset of the automorphism group of the code, called s-PD-set, to assist in decoding a received vector. Since the efficiency of the permutation decoding method depends on the size of the s-PD-set, there has been a lot of interest in finding s-PD-sets of small size for some families of linear codes.

This dissertation aims to provide s-PD-sets of minimum size $s+1$ for some families of $\mathbb{Z}_4$-linear codes to perform partial permutation decoding

for these codes. The s-PD-sets obtained in this work represent the first given for binary nonlinear codes. Moreover, we also analyse different classical decoding techniques for these $\mathbb{Z}_4$-linear codes to compare their outcomes and decide which decoding method gives better performance. The overview of the dissertation is the following:

- Chapter 2 provides an introduction to coding theory and helps this dissertation to be as self contained as possible. Firstly, we review basic definitions and results related to binary codes and $\mathbb{Z}_4$-linear codes, as well as the concept of equivalence between codes. We recall two widely known families of binary codes, called Hadamard and Kerdock codes, which are nonlinear in general. Nevertheless, some codes in these families are $\mathbb{Z}_4$-linear: they can be obtained as binary images under the Gray map of linear codes over $\mathbb{Z}_4$. Secondly, we present an in-depth examination of the recently proposed permutation decoding algorithm for $\mathbb{Z}_4$-linear codes [BBFV15], which generalizes the original decoding algorithm for linear codes developed in [Mac64, Pra62]. We introduce the known concept of s-PD-set, that is, a set of permutations that enables the correction of up to s errors without which it would not be possible to perform permutation decoding. Finally, we compute explicitly new upper bounds, denoted by f_m, on the maximum value of s for which s-PD-sets of minimum size $s+1$ may be found for systematic Hadamard and Kerdock codes of length 2^m.

- Chapter 3 is devoted to providing new s-PD-sets of minimum size $s+1$ to perform partial permutation decoding for the binary linear Hadamard code H_m of length $2^m, m \geq 4$. We present a first criterion that characterizes under which conditions a set of permutations forms an s-PD-set of size $s+1$ for H_m. An explicit construction fulfilling the criterion is also presented, together with two different recursive constructions of s-PD-sets for this family of codes. The upper bound f_m obtained at the end of Chapter 2 is achieved for all $m \geq 4$ by using the explicit construction.

- Chapter 4 establishes similar results for (nonlinear) $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ of length 2^m and type $2^\gamma 4^\delta$, where $m = \gamma + 2\delta - 1$, to those presented in Chapter 3 for binary linear Hadamard codes. Specifically, we start by giving another bound, denoted by $f_{\gamma,\delta}$ and generally smaller than f_m, for $H_{\gamma,\delta}$ that becomes crucial when s-PD-sets of size $s+1$ are searched for in a certain subgroup of $\mathrm{PAut}(H_{\gamma,\delta})$. This subgroup will

be further explained later in this chapter. For the $\mathbb{Z}_4$-linear Hadamard codes $H_{0,\delta}$ of length $2^{2\delta-1}$ with $\delta \geq 3$, s-PD-sets of size $s+1$ with s up to the upper bound $f_m = f_{0,\delta}$ are constructed. Additionally, for each $H_{\gamma,\delta}$ with $\gamma > 0$ and $\delta \geq 3$, s-PD-sets of size $s+1$ up to $f_{0,\delta}$ are given.

- Chapter 5 aims to acquire s-PD-sets of minimum size $s+1$ for $\mathbb{Z}_4$-linear codes in general. With this goal in mind we present a second criterion that can be applied to any $\mathbb{Z}_4$-linear code under certain conditions, unlike the method presented in Chapter 4, which can only be applied to $\mathbb{Z}_4$-linear Hadamard codes. As a specific application of this new criterion, explicit constructions of s-PD-sets of size $s+1$ for $\mathbb{Z}_4$-linear Hadamard and Kerdock codes are also presented. The s-PD-sets for $\mathbb{Z}_4$-linear Hadamard codes acquired in this chapter are different from the ones obtained in Chapter 4. Despite the fact that these new s-PD-sets do not achieve the upper bound introduced in Chapter 2 (and attained for some $\mathbb{Z}_4$-linear Hadamard codes in Chapter 4), they are generated by a single permutation.

- Chapter 6 presents the computational part of this dissertation. We give an introduction to other suitable decoding methods for $\mathbb{Z}_4$-linear codes: syndrome, lifted and coset decoding. We provide a comparison of the performance of these three methods and the permutation decoding method when they are applied to $\mathbb{Z}_4$-linear Hadamard and Kerdock codes. The s-PD-sets for these codes have been obtained by using the techniques introduced in the previous chapters of this dissertation. Finally, the manual of the implemented functions in MAGMA for quaternary linear codes and linear codes over finite fields is also included.

- Chapter 7 presents our conclusions and proposes future lines of research on this topic.

Finally, we must mention that part of the research included in this dissertation was presented at several conferences and published in their proceedings [BV14, BV15, BV16b, BPV16]:

[BV14] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for binary linear Hadamard codes," *Electron. Note Discr. Math.*, vol. 46, pp. 35–42, 2014. Proc. of the *IX Jornadas de Matemática Discreta y Algorítmica*, Tarragona, 7–9 July 2014.

[BV15] R. D. Barrolleta and M. Villanueva, "PD-sets for (nonlinear) Hadamard $\mathbb{Z}_4$-linear codes," in Proc. of the *21st Conference on Aplications of Computer Algebra (ACA 2015)*, Kalamata, Greece, pp. 135–139, 20–23 July 2015.

[BPV16] R. D. Barrolleta, J. Pujol, and M. Villanueva, "Comparing decoding methods for quaternary linear codes," to appear in *Electron. Note Discr. Math.*, 2016. Proc. of the *X Discrete Mathematics Days*, Barcelona, 6–8 July 2016.

[BV16b] R. D. Barrolleta and M. Villanueva, "PD-sets for $\mathbb{Z}_4$-linear codes: Hadamard and Kerdock codes," in Proc. of the *IEEE International Symposium on Information Theory (ISIT 2016)*, Barcelona, pp. 1317–1321, 10–15 July 2016.

Moreover, the results included at the end of Chapter 2, and in Chapters 3 and 4 have been already submitted to a journal [BV16b] (where it has been accepted after revisions) whereas those of Chapter 5 are planned to be submitted [BV16c]:

[BV16a] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for binary linear and $\mathbb{Z}_4$-linear Hadamard codes," submitted to *Designs, Codes and Cryptography*, 2016. arXiv:1512.01839.

[BV16c] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for some families of $\mathbb{Z}_4$-linear codes," 2016 (in preparation).

This work was partially supported by the Spanish MINECO under Grant TIN2013-40524-P, and by the Catalan AGAUR under Grant 2014SGR-691.

Ultimately, I visited the Department of Mathematics at Ghent University in Ghent, Belgium, from 1 September to 30 November 2014 with the objective of learning the main topics of the COST Action IC1104 project titled *Random Network Coding and Designs over* $\mathbb{F}_q$. The reader is referred to [Lam13] for an in-depth introduction to this topic. Recently, there has been interest in codes whose codewords are vector subspaces of a given vector space over $\mathbb{F}_q$ (unlike classical coding theory where codewords are vectors), where $\mathbb{F}_q$ is the finite field with q elements, due to their application in random network coding. After getting acquainted with these new codes, we focused on obtaining new geometric properties on constant dimension codes. A constant dimension code is a code fulfilling that each codeword has the same dimension k. Among all constant dimension codes, we studied those which all codewords intersect

pairwise in a subspace of dimension $k-t$. We found an upper bound for which we could find families of nontrivial constant dimension codes of this type, avoiding then the so-called *sunflowers*. After finding such a bound, we characterized the different families of the maximal nontrivial constant dimension codes. We also read [BEM99, Eis02, ER14] for constructing other nontrivial constant dimension codes intersecting in a $(k-t)$-dimensional subspace, for all $t \geq 3$. The work done during this stay has been presented at several international conferences [BBS+15, BBS+16] and has been already submitted to a journal [BSSV16], where it has also been accepted after minor revisions:

[BBS+15] R. D. Barrolleta, M. De Boeck, L. Storme, E. Suárez Canedo, and P. Vandendriessche, "A geometrical bound for the sunflower property," in Proc. of *Design and Application of Random Network Codes (DARNEC '15)*, Istanbul, Turkey, pp. 39, 4–6 November 2015.

[BBS+16] R. D. Barrolleta, M. De Boeck, L. Storme, E. Suárez Canedo, and P. Vandendriessche, "On constant distance random network codes," in Proc. of *Network Coding and Designs*, Dubrovnik, Croatia, pp. 48–49, 4–8 April 2016.

[BSSV16] R. D. Barrolleta, L. Storme, E. Suárez Canedo, and P. Vandendriessche, "On primitive constant dimension codes and a geometrical sunflower bound," submitted to *Adv. in Math. of Commun.*, 2016.

Chapter 2

Coding theory

This chapter provides an introduction to coding theory. Definitions and basic results of this theory used in the subsequent chapters are presented. Section 2.1 reviews basic concepts related to linear and nonlinear binary codes. Section 2.2 deals with $\mathbb{Z}_4$-linear codes, which are the Gray map image of quaternary linear codes. Section 2.3 introduces a well-known family of binary codes, called $\mathbb{Z}_4$-linear Hadamard codes, which will be the subject of study during most of the chapters in this dissertation. Section 2.4 looks at the special family of $\mathbb{Z}_4$-linear Kerdock codes. Section 2.5 presents a brief introduction to the equivalence of codes. Section 2.6 explains what permutation decoding is and how it works for binary linear and (nonlinear) $\mathbb{Z}_4$-linear codes. Throughout this section, the role of s-PD-sets in this decoding method is also explained. Finally, in Section 2.7, the minimum size of an s-PD-set for a given systematic binary code is determined. Moreover, this size for systematic binary Hadamard and Kerdock codes is explicitly computed.

The reader is referred to [MS77, HP03] for more information on the concepts given in Section 2.1; to [HKC$^+$94, Wan97, Kro01, PRV06] on $\mathbb{Z}_4$-linear codes in general, and $\mathbb{Z}_4$-linear Hadamard and Kerdock in particular; and to [MS77, BBFV15] on permutation decoding for linear and $\mathbb{Z}_4$-linear codes. The results given in Section 2.7 represent the first contributions of this dissertation. They have been included in the proceedings of some conferences [BV14, BV15, BV16b] and are planning to be published in [BV16a, BV16c].

2.1 Binary codes

Let $\mathbb{Z}_2$ be the ring of integers modulo 2 and let $\mathbb{Z}_2^n$ denote the set of all binary vectors of length n. Any nonempty subset C of $\mathbb{Z}_2^n$ is a *binary code* of length

n and a subgroup of $\mathbb{Z}_2^n$ is called a *binary linear code* of length n. From now on, the elements of a code will be called codewords. A binary linear code of length n can also be regarded as a linear subspace of $\mathbb{Z}_2^n$. In this case, the dimension k of a binary linear code C is defined as the dimension of the linear subspace C over $\mathbb{Z}_2$.

The *Hamming weight* of a vector $v \in \mathbb{Z}_2^n$, denoted by $\mathrm{wt}(v)$, is the number of nonzero coordinates in v. The *minimum Hamming weight* of a binary code C, denoted by $\mathrm{wt}(C)$, is the minimum value of $\mathrm{wt}(v)$ for all $v \in C \backslash \{\mathbf{0}\}$, where $\mathbf{0}$ represents the all-zero vector. The *Hamming distance* between two vectors $u, v \in \mathbb{Z}_2^n$, denoted by $d(u, v)$, is the number of coordinates in which u and v differ, that is, $d(u, v) = \mathrm{wt}(u + v)$. The *minimum Hamming distance* of a binary code C, denoted by $d(C)$, is the minimum value of $d(u, v)$, for all $u, v \in C$ satisfying $u \neq v$. The minimum Hamming distance of a binary code C will be denoted by d only if the code C we are referring to is clear from the context. It is well known that if C is a binary linear code, $d(C) = \mathrm{wt}(C)$.

The minimum Hamming distance d of a binary code C determines the number of errors that the code can correct. Let y be a received vector. If $s \leq \lfloor (d-1)/2 \rfloor$, then there is only one codeword $c \in C$ such that $d(c, y) \leq s$. The parameter

$$t = \lfloor (d-1)/2 \rfloor$$

is called the *error-correcting capability* of the code C and C is said to be a t-error-correcting code.

Let C be a binary code of length n and size $|C| = 2^k$. For a vector $v \in \mathbb{Z}_2^n$ and a set $I \subseteq \{1, \ldots, n\}$, $|I| = k$, we denote the restriction of v to the coordinates in I by $v_I \in \mathbb{Z}_2^k$ and the set $\{v_I : v \in C\}$ by C_I. A set $I \subseteq \{1, \ldots, n\}$ of k coordinate positions such that $|C_I| = 2^k$ is called an *information set* for C. If such a set I exists, then C is said to be a *systematic code*. For each information set I of size k, the set $\{1, \ldots, n\} \backslash I$ of the remaining $n - k$ coordinate positions is called a *check set* for C.

The most common ways to describe a linear code are with either a generator or a parity check matrix. A *generator matrix* for a linear code C of length n and dimension k is a $k \times n$ matrix G whose rows form a basis of C. In general, there are many generator matrices for a linear code. Any set of k coordinate positions corresponding to k independent columns of G forms an information set for C. A *parity check matrix* H for a linear code C is a $(n-k) \times n$ matrix of rank $n-k$ whose null space is the code C. A generator matrix G and a parity check matrix H for the linear code C satisfy $GH^T = \mathbf{0}$, where H^T denotes the transpose matrix of H. A generator matrix G is said

to be in *standard form* if its first k columns form the identity matrix of size k, denoted by Id_k. If $G = (\mathrm{Id}_k \,|A)$ is a generator matrix for the linear code C in standard form, then

$$H = (-A^T | \,\mathrm{Id}_{n-k}), \tag{2.1}$$

is a parity check matrix for C. A parity check matrix H as in (2.1) is said to be in standard form.

It is possible to create longer codes by adding a coordinate. There are many possible ways to extend a code, but the most common is to choose the extension so that the sum of all coordinates is 0. Let C be a binary linear code of length n, dimension k and minimum distance d. The *extended code* of C, denoted by $\widehat{C}$, is defined as

$$\widehat{C} = \{(x_1, \ldots, x_{n+1}) \in \mathbb{Z}_2^{n+1} \ : \ (x_2, \ldots, x_{n+1}) \in C \text{ with } \sum_{i=1}^{n+1} x_i = 0\}.$$

The extended code $\widehat{C}$ of a linear code C is also linear. Moreover, $\widehat{C}$ is a binary linear code of length $n+1$, dimension k and minimum distance $\widehat{d}$, where $\widehat{d} = d$ or $d+1$. Let G and H be generator and parity check matrices, respectively, for C. Then, a generator matrix $\widehat{G}$ for $\widehat{C}$ can be obtained from G by adding an extra column to G so that the sum of the coordinates of each row of $\widehat{G}$ is 0. A parity check matrix $\widehat{H}$ for $\widehat{C}$ is the matrix

$$\widehat{H} = \begin{pmatrix} 1 & \mathbf{1} \\ \mathbf{0} & H \end{pmatrix}, \tag{2.2}$$

where $\mathbf{0}$ and $\mathbf{1}$ represent the all-zero and all-one vector, respectively.

The *inner product* of two vectors $u, v \in \mathbb{Z}_2^n$ is defined as

$$\langle u, v \rangle = \sum_{i=1}^{n} u_i v_i \in \mathbb{Z}_2.$$

If $\langle u, v \rangle = 0$, then u and v are called *orthogonal*. Denote the set of vectors which are orthogonal to all codewords of a binary code C by $C^\perp$, that is,

$$C^\perp = \{x \in \mathbb{Z}_2^n : \langle x, u \rangle = 0, \text{ for all } u \in C\}.$$

When C is a linear code, then $C^\perp$ is called the *dual* of the code C. If G and H are generator and parity check matrices, respectively, for C, then H and G are generator and parity check matrices, respectively, for $C^\perp$.

Two binary codes C_1 and C_2 of length n are said to be *equivalent* (or *permutation equivalent*) if one can be obtained form the other by permuting the coordinates. The concept of equivalent codes will be more deeply discussed in Section 2.5.

Let $m \geq 2$. The $m \times (2^m - 1)$ matrix whose columns are the numbers $1, 2, \ldots, 2^m - 1$ written as binary numerals is the parity check matrix of a binary code of length $2^m - 1$, dimension $2^m - 1 - m$ and minimum Hamming distance 3. Alternatively, the columns of this parity check matrix can be seen as the $2^m - 1$ nonzero vectors in $\mathbb{Z}_2^m$. Any rearrangement of columns of this matrix gives an equivalent code, and hence any one of these equivalent codes will be called *binary Hamming code* of length $2^m - 1$. The *binary simplex code* of length $2^m - 1$, denoted by S_m, is the dual of the binary Hamming code of length $2^m - 1$.

Example 1. *The matrix*

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

has as columns all the $2^4 - 1 = 15$ *nonzero vectors in* $\mathbb{Z}_2^4$. *Then, H is a parity check matrix for the binary Hamming code of length* 15. *The matrix H is also a generator matrix in standard form for the binary simplex code* S_4.

2.2 $\mathbb{Z}_4$-linear codes

Let $\mathbb{Z}_4$ be the rings of integers modulo 4 and let $\mathbb{Z}_4^n$ be the set of all n-tuples over the ring $\mathbb{Z}_4$. Henceforth, the elements of $\mathbb{Z}_4^n$ will also be called vectors despite the fact that $\mathbb{Z}_4^n$ is not a vector space. Any nonempty subset $\mathcal{C}$ of $\mathbb{Z}_4^n$ is a *quaternary code* of length n and a subgroup of $\mathbb{Z}_4^n$ is called a *quaternary linear code* of length n.

Let e_i be the binary vector or tuple over $\mathbb{Z}_4$ with a 1 in the ith coordinate and zeros elsewhere. Let $\mathbf{0}, \mathbf{1}, \mathbf{2}$ and $\mathbf{3}$ be the binary vectors or n-tuples over $\mathbb{Z}_4$ having 0, 1, 2 and 3, respectively, repeated in each coordinate. It will be clear by the context whether we refer to binary vectors or n-tuples over $\mathbb{Z}_4$.

The elements of $\mathbb{Z}_4$ have the following Lee weights: $\mathrm{wt}_L(0) = 0$, $\mathrm{wt}_L(1) = \mathrm{wt}_L(3) = 1$ and $\mathrm{wt}_L(2) = 2$. Then, the *Lee weight* of a vector $u \in \mathbb{Z}_4^n$, denoted by $\mathrm{wt}_L(u)$, is the addition of the weights of its coordinates, whereas the *Lee*

distance between two vectors $u, v \in \mathbb{Z}_4^n$, denoted by $d_L(u,v)$, is $d_L(u,v) = \mathrm{wt}_L(u-v)$. The *minimum Lee distance* of a quaternary code $\mathcal{C}$, denoted by $d_L(\mathcal{C})$, is the minimum value of $d_L(u,v)$ for all $u, v \in \mathcal{C}$ satisfying $u \neq v$. The *minimum Lee weight* of a quaternary code $\mathcal{C}$, denoted by $\mathrm{wt}_L(C)$, is the minimum value of $\mathrm{wt}_L(v)$, for all $u \in \mathcal{C}\backslash\{\mathbf{0}\}$. Again, if $\mathcal{C}$ is a quaternary linear code, $d_L(\mathcal{C}) = \mathrm{wt}_L(C)$.

Quaternary codes can be viewed as binary codes under the usual Gray map $\Phi : \mathbb{Z}_4^n \to \mathbb{Z}_2^{2n}$ defined as

$$\Phi((y_1, \ldots, y_n)) = (\phi(y_1), \ldots, \phi(y_n)), \tag{2.3}$$

where $\phi(0) = (0,0)$, $\phi(1) = (0,1)$, $\phi(2) = (1,1)$, $\phi(3) = (1,0)$, for all $y = (y_1, \ldots, y_n) \in \mathbb{Z}_4^n$. The Gray map is an isometry which transforms Lee distances over $\mathbb{Z}_4^n$ into Hamming distances over $\mathbb{Z}_2^{2n}$. Therefore, the minimum Lee distance of a quaternary code $\mathcal{C}$ coincides with the minimum Hamming distance of $C = \Phi(\mathcal{C})$, that is, $d_L(\mathcal{C}) = d(\Phi(\mathcal{C}))$.

Two quaternary codes $\mathcal{C}_1$ and $\mathcal{C}_2$ of length n are said to be *permutation equivalent* if one can be obtained form the other by permuting the coordinates. The concept of equivalent codes will be more deeply discussed in Section 2.5.

If $\mathcal{C}$ is a quaternary linear code, then the binary code $C = \Phi(\mathcal{C})$ is said to be a $\mathbb{Z}_4$-*linear code*. Moreover, since $\mathcal{C}$ is a subgroup of $\mathbb{Z}_4^n$, it is isomorphic to an abelian group $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$ and we say that $\mathcal{C}$ (or equivalently, the corresponding $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$) is of type $2^\gamma 4^\delta$ as a group. The code $\mathcal{C}$ has $|\mathcal{C}| = 2^{\gamma+2\delta}$ codewords, where $2^{\gamma+\delta}$ of them have order two.

A quaternary linear code $\mathcal{C}$ of length n and type $2^\gamma 4^\delta$ can also be regarded as a $\mathbb{Z}_4$-submodule of $\mathbb{Z}_4^n$. As a $\mathbb{Z}_4$-module, $\mathcal{C}$ may or may not be free. Recall that a $\mathbb{Z}_4$-module M is free if there exists a subset $E \subseteq M$ such that every element in M is uniquely expressible as a linear combination over $\mathbb{Z}_4$ of the elements in E [HP03]. Then, the quaternary linear code $\mathcal{C}$ is free if $\gamma = 0$. Although $\mathcal{C}$ is not a free module in general, every codeword is uniquely expressible in the form

$$\sum_{i=1}^{\gamma} \lambda_i u_i + \sum_{j=1}^{\delta} \mu_j v_j,$$

where $\lambda_i \in \mathbb{Z}_2$ for all $1 \leq i \leq \gamma$, $\mu_j \in \mathbb{Z}_4$ for all $1 \leq j \leq \delta$ and u_i, v_j are codewords of $\mathcal{C}$ of order two and four, respectively. The matrix $\mathcal{G}$ that has as rows the codewords u_i and v_j is a generator matrix for $\mathcal{C}$. As for linear codes, there is a standard form for the generator matrix of $\mathcal{C}$. In [HKC+94], it was

shown that any quaternary linear code of type $2^\gamma 4^\delta$ is permutation equivalent to a quaternary linear code $\mathcal{C}_S$ with a generator matrix of the following form

$$\mathcal{G}_S = \left(\begin{array}{ccc} 2T & 2\mathrm{Id}_\gamma & \mathbf{0} \\ \hline S & R & \mathrm{Id}_\delta \end{array}\right), \tag{2.4}$$

where R, T are matrices over $\mathbb{Z}_4$ with entries in $\{0,1\} \subseteq \mathbb{Z}_4$ of size $\delta \times \gamma$ and $\gamma \times (\beta - \gamma - \delta)$, respectively; and S is a matrix over $\mathbb{Z}_4$ of size $\delta \times (\beta - \gamma - \delta)$.

In general, a $\mathbb{Z}_4$-linear code is nonlinear if it is regarded as a code over $\mathbb{Z}_2$. The following lemmas are useful when dealing with the linearity of $\mathbb{Z}_4$-linear codes. Let $u * v$ denote the component-wise product of two vectors $u, v \in \mathbb{Z}_4^n$.

Lemma 2 ([HKC+94, Wan97]). *For all $u, v \in \mathbb{Z}_4^n$, we have*

$$\Phi(u+v) = \Phi(u) + \Phi(v) + \Phi(2u * v).$$

Lemma 3 ([HKC+94, Wan97]). *Let $\mathcal{C}$ be a quaternary linear code. The $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$ is a binary linear code if and only if $2u * v \in \mathcal{C}$, for all $u, v \in \mathcal{C}$.*

One can strengthen Lemma 3 via the generators of order four of the quaternary linear code. Specifically, if $\mathcal{G}$ is a generator matrix of a quaternary linear code $\mathcal{C}$ of type $2^\gamma 4^\delta$ and $u_1, \ldots, u_\gamma$ and $v_1, \ldots, v_\delta$ are the rows of order two and order four in $\mathcal{G}$, respectively, then the $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$ is a binary linear code if and only if $2v_i * v_j \in \mathcal{C}$, for all $1 \le i < j \le \delta$. It is clear that $2u_i * v = \mathbf{0} \in \mathcal{C}$ for all $1 \le i \le \gamma$ and $v \in \mathcal{C}$; and $2v_j * v_j = 2v_j \in \mathcal{C}$ for all $1 \le j \le \delta$.

Example 4. *Let $\mathcal{C}$ be the quaternary linear code of length 16 with generator matrix*

$$\mathcal{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix}. \tag{2.5}$$

Denote the ith row of matrix (2.5) by v_i. It is straightforward to check that

$$2v_2 * v_3 = (0000020200000202) \notin \mathcal{C}.$$

Thus, by Lemma 3, the $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$ is a binary nonlinear code. The quaternary linear code $\mathcal{C}$ is permutation equivalent, by using the permutation $(1, 14, 11, 8, 5, 16, 13, 10, 7, 4, 2, 15, 12, 9, 6, 3) \in \mathrm{Sym}(16)$, to a quaternary linear code $\mathcal{C}_S$ with generator matrix $\mathcal{G}_S$ in standard form (2.4), where

$$\mathcal{G}_S = \left(\begin{array}{cccccccccccccccc} 3 & 2 & 3 & 2 & 1 & 3 & 2 & 1 & 0 & 2 & 1 & 0 & 3 & 1 & 0 & 0 \\ 2 & 3 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 0 & 0 & 1 \end{array}\right). \tag{2.6}$$

The code $\mathcal{C}$ is of type $2^0 4^3$, so it has $4^3 = 64$ codewords.

Example 5. *Let $\mathcal{C}$ be the quaternary linear code of length 16 with generator matrix*

$$\mathcal{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ u_1 \\ u_2 \end{pmatrix} \tag{2.7}$$

It is easy to see that $2v_1 * v_2 = 2v_2 \in \mathcal{C}$, so $C = \Phi(\mathcal{C})$ is a binary linear code by Lemma 3. The code $\mathcal{C}$ is permutation equivalent via the permutation $(1, 15, 11, 7, 4, 2, 16, 12, 8, 5, 13, 9, 14, 10, 6, 3) \in \mathrm{Sym}(16)$ to a quaternary linear code $\mathcal{C}_S$ with generator matrix $\mathcal{G}_S$ in standard form (2.4), where

$$\mathcal{G}_S = \left(\begin{array}{cccccccccccccccc} 0 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 2 & 0 & 0 \\ \hline 3 & 2 & 0 & 3 & 2 & 0 & 3 & 2 & 1 & 0 & 3 & 2 & 1 & 1 & 1 & 0 \\ 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 1 \end{array}\right). \tag{2.8}$$

The code $\mathcal{C}$ is of type $2^2 4^2$, so it has $2^2 4^2 = 64$ codewords.

The *inner product* of two vectors $u, v \in \mathbb{Z}_4^n$ is defined as

$$\langle u, v \rangle = \sum_{i=1}^{n} u_i v_i \in \mathbb{Z}_4.$$

Given a quaternary linear code $\mathcal{C}$ of length n and type $2^\gamma 4^\delta$, the *quaternary dual code* of $\mathcal{C}$, denoted by $\mathcal{C}^\perp$, is defined as

$$\mathcal{C}^\perp = \{x \in \mathbb{Z}_4^n : \langle x, u \rangle = 0, \text{ for all } u \in \mathcal{C}\}.$$

The quaternary dual code is of length n and type $2^\gamma 4^{n-\gamma-\delta}$ [HKC+94]. The weight enumerator polynomial of $\mathcal{C}^\perp$ is related to the weight enumerator polynomial of $\mathcal{C}$ by the MacWilliams identity [MS77, Chapter 5]. The corresponding binary code $\Phi(\mathcal{C}^\perp)$ is denoted by $C_\perp$ and called the $\mathbb{Z}_4$-*dual code* of C. The codes C and $\mathcal{C}_\perp$ are not necessarily linear, so they are not dual in the binary linear sense, but the weight enumerator of $C_\perp$ is the MacWilliams transform of the weight enumerator of C.

Since 1994, quaternary linear codes became significant because, in some cases, after applying the Gray map, we obtain binary nonlinear codes better than any known binary linear code with the same parameters: length, number of codewords and minimum distance. This is the case, for example, of Kerdock and Preparata codes. This discovery is due to the influential paper [HKC+94] where, among other things, it is shown that the Kerdock

codes and some Preparata-like codes are $\mathbb{Z}_4$-linear codes and, moreover, the $\mathbb{Z}_4$-dual code of the Kerdock code is a Preparata-like code. Later, several other $\mathbb{Z}_4$-linear codes with the same parameters as some well known families of binary linear codes (for example, extended Hamming, Hadamard, and Reed-Muller codes) have been studied and classified [BPR03, Kro01, PRV06, PRS09, PPV11].

Since then, a lot of research has been done on quaternary linear codes and linear codes over more general finite rings. Nevertheless, the examples of better-than-linear codes found since then are comparatively sparse. In [KZ13], the *extended dualized Kerdock codes* $\hat{\mathcal{K}}_{k+1}^*$ ($k \geq 3$ odd), which are quaternary linear codes with high minimum Lee distance, are constructed. In [KWZ16], it is shown that the codes $\hat{\mathcal{K}}_4^*$ and $\hat{\mathcal{K}}_6^*$ satisfy that the minimum Hamming distance of their Gray map images is higher than the minimum Hamming distance of any comparable binary linear code. A table with the current better-than-linear codes can be found in [KWZ16]. For moderate lengths, in order to determine whether a nonlinear code is better-than-linear or not, the online tables [Gra09, BCFS16] containing the best known linear codes can be used. Tables with the best known $\mathbb{Z}_4$-linear codes and binary nonlinear codes are also available at [AA09] and [LRS99], respectively.

Despite this dissertation is focused on obtaining s-PD-sets for $\mathbb{Z}_4$-linear codes, the results presented in this work may be generalized for other nonlinear binary codes with subjacent algebraic structures different from $\mathbb{Z}_4$. One possible generalization of $\mathbb{Z}_4$-linear codes are $\mathbb{Z}_2\mathbb{Z}_4$-linear codes. A code $\mathcal{C}$ is said to be $\mathbb{Z}_2\mathbb{Z}_4$-additive if the set of coordinates can be partitioned into two subsets X and Y such that the punctured code of $\mathcal{C}$ by deleting the coordinates outside X (respectively, Y) is a binary linear code (respectively, a quaternary linear code). Their corresponding binary images, via a generalized Gray map, are called $\mathbb{Z}_2\mathbb{Z}_4$-linear codes. The fundamental parameters as well as the standard forms for generator and parity check matrices and the duality concepts for these codes are studied in [BFP$^+$10, BFP$^+$14]. Other possible generalizations of $\mathbb{Z}_4$-linear codes are $\mathbb{Z}_{2^k}$-linear codes, which are defined as the binary image of $\mathbb{Z}_{2^k}$-ary codes by generalized Gray maps in [Car91, Kro07, DF11]. Finally, it is also worth mentioning that in [AS13, AS14] $\mathbb{Z}_2\mathbb{Z}_{2^s}$-additive and $\mathbb{Z}_{p^r}\mathbb{Z}_{p^s}$-additive codes are introduced, generalizing naturally both $\mathbb{Z}_2\mathbb{Z}_4$-additive codes and $\mathbb{Z}_{2^k}$-ary codes.

2.3 $\mathbb{Z}_4$-linear Hadamard codes

A *Hadamard matrix* H of order n is a $n \times n$ matrix of $+1's$ and $-1's$ such that $HH^T = n\,\mathrm{Id}$. It is well known that if a Hadamard matrix H or order n exists, then n is $1, 2$ or a multiple of 4 [MS77, AK92]. Two Hadamard matrices are *equivalent* if one matrix can be obtained from the other by permuting rows and (or) columns and multiplying rows and (or) columns by -1. We can change the first row and column of H intro $+1's$ and we obtain and equivalent Hadamard matrix H', which is called normalized. If $+1's$ are replaced by $0's$ and $-1's$ by $1's$, H' is changed into a *binary Hadamard matrix* $c(H')$. The binary code consisting of the rows of $c(H')$ and their complements is called a *binary Hadamard code*.

Hadamard codes have been used in real word applications. They were used in early satellite transmissions, for example, in the 1972 Mariner mission to Mars. Modern CDMA cellphones use Hadamard matrices (Walsh covers) to modulate transmission on the uplink and minimise interference with other transmissions to the base station. The Walsh-Hadamard Transform is in common use as a fast discrete transform. New applications are pattern recognition, neuroscience and optical communication, among others. In addition, they are also used in cryptography and stenography. Despite this, there is still no uniform technique for constructing all the known Hadamard matrices. In fact, the Hadamard conjecture, that states that for every natural number n there exists a Hadamard matrix of order $4n$, remains one of the great unsolved problems of mathematics.

A binary Hadamard code of length n is a binary code with $2n$ codewords and minimum distance $n/2$. In a binary Hadamard code, all codewords, except the all-one and all-zero codewords, have Hamming weight $n/2$. It is well known that there is a unique binary linear Hadamard code H_m of length $n = 2^m$, for any $m \geq 2$, which is the dual of the extended Hamming code of length 2^m [MS77, Chapter 2]. A generator matrix G_m for H_m can be constructed as follows:

$$G_m = \begin{pmatrix} 1 & \mathbf{1} \\ \mathbf{0} & G' \end{pmatrix}, \tag{2.9}$$

where G' is any matrix having as column vectors the $2^m - 1$ nonzero vectors from $\mathbb{Z}_2^m$, with the vectors e_i, $i \in \{1, \ldots, m\}$, in the first m positions. Note that G' can be seen as a generator matrix of the binary simplex code S_m of length $2^m - 1$, as noticed in Section 2.1.

Example 6. *Let H_4 be the binary linear Hadamard code of length 16 with*

generator matrix

$$G_4 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{2.10}$$

constructed as in (2.9). Note that in this construction the generator matrix for the simplex code S_4 of length 15 *given in Example 1 is involved.*

In general, binary Hadamard codes are nonlinear. In this case, it is desirable to have a subjacent algebraic structure, like a group or a ring. From the coding theory perspective, it is also desired that the algebraic structure preserves the Hamming distance. This is the case, for example, of $\mathbb{Z}_4$-linear codes. The quaternary linear codes that, under the Gray map, give a binary Hadamard code are called *quaternary linear Hadamard codes* and the corresponding $\mathbb{Z}_4$-linear codes are called $\mathbb{Z}_4$-*linear Hadamard codes*.

For any $m \geq 3$ and each $\delta \in \{1, \ldots, \lfloor \frac{m+1}{2} \rfloor\}$, there is a unique (up to equivalence) $\mathbb{Z}_4$-linear Hadamard code of length 2^m which is the Gray map image of a quaternary linear code of length $\beta = 2^{m-1}$ and type $2^\gamma 4^\delta$, where $m = \gamma + 2\delta - 1$. Moreover, for a fixed m, all these codes are pairwise nonequivalent, except for $\delta = 1$ and $\delta = 2$, since these ones are equivalent to the binary linear Hadamard code H_m of length 2^m [Kro01]. Therefore, the number of nonequivalent $\mathbb{Z}_4$-linear Hadamard codes of length 2^m is $\lfloor \frac{m-1}{2} \rfloor$ for all $m \geq 3$. Note that when $\delta \geq 3$, the $\mathbb{Z}_4$-linear Hadamard codes are nonlinear.

Let $\mathcal{H}_{\gamma,\delta}$ be the quaternary linear Hadamard code of length $\beta = 2^{m-1}$ and type $2^\gamma 4^\delta$, where $m = \gamma + 2\delta - 1$, and let $H_{\gamma,\delta} = \Phi(\mathcal{H}_{\gamma,\delta})$ be the corresponding $\mathbb{Z}_4$-linear code of length $2\beta = 2^m$. A generator matrix $\mathcal{G}_{\gamma,\delta}$ for $\mathcal{H}_{\gamma,\delta}$ can be constructed by using the following recursive constructions:

$$\mathcal{G}_{\gamma+1,\delta} = \begin{pmatrix} \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} \\ \mathbf{0} & \mathbf{2} \end{pmatrix}, \tag{2.11}$$

$$\mathcal{G}_{\gamma,\delta+1} = \begin{pmatrix} \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} \\ \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \end{pmatrix}, \tag{2.12}$$

starting from $\mathcal{G}_{0,1} = (1)$. First, the matrix $\mathcal{G}_{0,\delta}$ is obtained from $\mathcal{G}_{0,1}$ by using recursively $\delta - 1$ times (2.12), and then $\mathcal{G}_{\gamma,\delta}$ is constructed from $\mathcal{G}_{0,\delta}$ by using γ times (2.11). Note that the rows of order four remain in the upper part of $\mathcal{G}_{\gamma,\delta}$ while those of order two stay in the lower part.

Example 7. *The code $\mathcal{C}$ introduced in Example 4 is the quaternary linear Hadamard code $\mathcal{H}_{0,3}$ of length $\beta = 16$ and type $2^0 4^3$. The $\mathbb{Z}_4$-linear Hadamard code $H_{0,3} = \Phi(\mathcal{H}_{0,3})$ is a binary Hadamard code of length 32 with 64 codewords and minimum Hamming distance 16. The code $H_{0,3}$ is the smallest $\mathbb{Z}_4$-linear Hadamard code which is nonlinear.*

On the other hand, the code $\mathcal{C}$ introduced in Example 5 is the quaternary linear Hadamard code $\mathcal{H}_{2,2}$ of length $\beta = 16$ and type $2^2 4^2$. The $\mathbb{Z}_4$-linear Hadamard code $H_{2,2} = \Phi(\mathcal{H}_{2,2})$ is the binary linear Hadamard code of length 32 with 64 codewords and minimum Hamming distance 16. The binary linear Hadamard code of this length can also be obtained as the Gray map image of $\mathcal{H}_{4,1}$. Therefore, both codes $H_{2,2}$ and $H_{4,1}$ are equivalent to the code H_4 given in Example 6. Finally, see that there are exactly

$$\left\lfloor \frac{m-1}{2} \right\rfloor = \left\lfloor \frac{5-1}{2} \right\rfloor = 2$$

nonequivalent $\mathbb{Z}_4$-linear Hadamard codes of length $2^5 = 32$, which are either the codes $H_{0,3}$ and $H_{2,2}$ or the codes $H_{0,3}$ and $H_{4,1}$.

The $\mathbb{Z}_4$-linear Hadamard codes have been studied and classified in [Kro01, PRV06]. Hadamard matrices with different subjacent algebraic structures have been extensively studied, as well as the links with other topics in algebraic combinatorics [Hor07]. This is the case, for example, of $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes and Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes. The $\mathbb{Z}_2\mathbb{Z}_4$-additive codes such that, under the generalized Gray map, give a binary Hadamard code are called $\mathbb{Z}_2\mathbb{Z}_4$-additive Hadamard codes and the corresponding $\mathbb{Z}_2\mathbb{Z}_4$-linear codes are called $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes. These codes have been studied in [PRV06, RSV09, KV15] and generalize the $\mathbb{Z}_4$-linear Hadamard codes presented in this section. On the other hand, Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes, which are binary Hadamard codes after a suitable Gray map from a subgroup of direct products of $\mathbb{Z}_2$, $\mathbb{Z}_4$, and Q_8 (where Q_8 is the quaternionic group of order eight) have been studied in [RR13, MR15].

2.4 $\mathbb{Z}_4$-linear Kerdock codes

We now define the $\mathbb{Z}_4$-linear Kerdock code of length 2^m as the Gray map image of the extended code of a cyclic quaternary linear code of length $n = 2^{m-1} - 1$.

Let $\mathbb{Z}_4[x]$ and $\mathbb{Z}_2[x]$ be the polynomial ring over $\mathbb{Z}_4$ and $\mathbb{Z}_2$, respectively. Let $\mu : \mathbb{Z}_4[x] \to \mathbb{Z}_2[x]$ be the map that performs a modulo 2 reduction of the

coefficients of $h(x) \in \mathbb{Z}_4[x]$. A monic polynomial $h(x) \in \mathbb{Z}_4[x]$ is said to be a *primitive basic irreducible* polynomial if $\mu(h(x))$ is primitive over $\mathbb{Z}_2[x]$.

Let $h(x)$ be a primitive basic irreducible polynomial of degree $m-1$ over $\mathbb{Z}_4$ such that $h(x)$ divides $x^n - 1$, and let $g(x)$ be the reciprocal polynomial to the polynomial $(x^n - 1)/((x-1)h(x))$. Note that $h(x)$ is the Hensel lift of the binary primitive polynomial $\mu(h(x))$ of degree $m-1$. Let $\mathcal{K}_m^-$ be the quaternary linear cyclic code of length $n = 2^{m-1} - 1$ with generator polynomial $g(x)$. The *quaternary Kerdock code* $\mathcal{K}_m$ is the code obtained from $\mathcal{K}_m^-$ by adding a zero-sum check symbol at the end of each codeword of $\mathcal{K}_m^-$. The $\mathbb{Z}_4$*-linear Kerdock code* K_m is defined to be the Gray map image $\Phi(\mathcal{K}_m)$ of the quaternary Kerdock code $\mathcal{K}_m$, so K_m is a binary code of length 2^m and size 4^m. Moreover, it is known that the minimum Hamming distance of K_m is $2^{m-1} - 2^{\lfloor (m-2)/2 \rfloor}$ [HKC+94, Wan97, HP03].

Suppose that $g(x) = g_0 + g_1 x + \cdots + g_r x^r \in \mathbb{Z}_4[x]$ is a generator polynomial of $\mathcal{K}_m^-$, where $r = 2^{m-1} - m - 1$. Let $g_\infty = -(g_0 + \cdots + g_r)$. It is clear that the $m \times 2^{m-1}$ matrix

$$\begin{pmatrix} g_0 & g_1 & \cdots & g_m & & & g_\infty \\ & g_0 & g_1 & \cdots & g_m & & g_\infty \\ & & \ddots & & & \ddots & \vdots \\ & & & g_0 & g_1 & \cdots & g_m & g_\infty \end{pmatrix} \tag{2.13}$$

is a generator matrix of $\mathcal{K}_m$.

Example 8. *The polynomial* $h(x) = x^3 + 2x^2 + x - 1 \in \mathbb{Z}_4[x]$ *is a primitive basic irreducible polynomial of degree* 3 *since* $\mu(h(x)) = x^3 + x + 1 \in \mathbb{Z}_2[x]$ *is primitive. The polynomial* $g(x)$ *is equal to* $h(x)$ *since* $x^3 + 2x^2 + x - 1$ *is precisely the reciprocal polynomial of* $(x^7-1)/((x-1)h(x)) = x^3 - x^2 + 2x - 1$. *A generator matrix of the quaternary Kerdock code* $\mathcal{K}_4$ *constructed as in (2.13) is*

$$\begin{pmatrix} 3 & 1 & 2 & 1 & 0 & 0 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 & 0 & 0 & 1 \\ 0 & 0 & 3 & 1 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 3 & 1 & 2 & 1 & 1 \end{pmatrix}.$$

The $\mathbb{Z}_4$*-linear Kerdock code* $K_4 = \Phi(\mathcal{K}_4)$ *is a nonlinear binary code of length* $2^4 = 16$, *size* $4^4 = 256$ *and minimum Hamming distance* 6.

A comprehensive study of (binary) Kerdock codes can be found in [MS77, Chapter 15], where they are defined as the union of the first-order Reed-Muller code $RM(1,m)$ and $2^{m-1} - 1$ cosets of $RM(1,m)$ in $RM(2,m)$ corresponding to quadratic bent functions such that the sum of any two is again

a quadratic bent function. In [HKC+94], it is shown that for even $m \geq 4$ a simple rearrangement of the coordinates of K_m leads directly to the original definition of Kerdock code given in [Ker72]. In [Nec89], a connection between sequences over $\mathbb{Z}_4$ and Kerdock codes punctured in two coordinates was first discovered. Kerdock codes and Kerdock-like codes have also been studied in [Kan95, BPRZ03, Phe15].

Note that we use the term Kerdock code more broadly than other authors in the following sense: we call the $\mathbb{Z}_4$-linear Kerdock code, the code we obtain after applying the Gray map to the quaternary Kerdock code $\mathcal{K}_m$, regardless of whether m is even or not. The Kerdock codes of length 2^m were first defined only for even values of m, $m \geq 4$.

2.5 Equivalence of codes

In this section, we study when two codes are "essentially the same". This concept is termed "equivalence". Usually, we are interested in properties of codes, such as weight distribution, which remain unchanged when passing from one code to another that is essentially the same.

One way to view two binary linear codes as "essentially the same" is to regard them "the same" if they are isomorphic as vector spaces. Nevertheless, in that case the concept of weight is lost: codewords of one weight may be sent to codewords of a different weight by the isomorphism. Furthermore, we are interested in equivalence between codes which could be binary nonlinear or linear over other fields or rings. Clearly, any permutation of coordinates that sends one code to another preserves the weight of codewords, regardless the field or ring.

Let $\mathrm{Sym}(n)$ be the symmetric group of permutations on the set $\{1, \ldots, n\}$ and let $\mathrm{id} \in \mathrm{Sym}(n)$ be the identity permutation. Let R be a ring. Throughout this dissertation, R will usually be $\mathbb{Z}_2$, $\mathbb{Z}_4$ or the finite field with q elements $\mathbb{F}_q$. The group operation in $\mathrm{Sym}(n)$ is the function composition, $\sigma_1\sigma_2$, which maps any element x to $\sigma_1(\sigma_2(x))$, $\sigma_1, \sigma_2 \in \mathrm{Sym}(n)$. A $\sigma \in \mathrm{Sym}(n)$ acts linearly on words of R^n by permuting their coordinates as follows: $\sigma((v_1, \ldots, v_n)) = (v_{\sigma^{-1}(1)}, \ldots, v_{\sigma^{-1}(n)})$.

Two codes $C_1 \subseteq R^n$ and $C_2 \subseteq R^n$ are said to be *permutation equivalent* if there is a permutation of coordinates which sends C_1 into C_2. The *permutation automorphism group* of a code $C \subseteq R^n$, denoted by $\mathrm{PAut}(C)$, is the set of permutations that map C to itself. Indeed, the set of all permutations that preserve the set of codewords is a group. If C is a code of length n, then

$\mathrm{PAut}(C)$ is a subgroup of the symmetric group $\mathrm{Sym}(n)$.

When we are considering codes over rings other than $\mathbb{Z}_2$, equivalence could take a more general form. For example, over $\mathbb{Z}_4$, there are other maps that preserve the Lee weight of codewords. These maps include those which multiply the coordinates of the codewords by units of the ring.

A *monomial matrix* is a square matrix with exactly one nonzero divisor entry in each row and column. A monomial matrix M can be represented as a product of a diagonal matrix D and a permutation matrix P. Hence, $M = DP$. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two linear codes of the same length over the ring R, and let $\mathcal{G}_1$ be a generator matrix of $\mathcal{C}_1$. Then, $\mathcal{C}_1$ and $\mathcal{C}_2$ are said to be *monomially equivalent* if there is a monomial matrix M such that $M\mathcal{G}_1$ is a generator matrix of $\mathcal{C}_2$. Monomially equivalence and permutation equivalence are precisely the same for binary linear codes. The *monomial automorphism group* of a linear code $\mathcal{C}$ over a ring, denoted by $\mathrm{MAut}(\mathcal{C})$, is the set of monomial matrices that map $\mathcal{C}$ into itself. Since any permutation can be seen as a permutation matrix, $\mathrm{PAut}(\mathcal{C})$ is a subgroup of $\mathrm{MAut}(\mathcal{C})$.

In the literature, we can find results on the permutation automorphism groups of the binary codes considered in this dissertation. Specifically, it is well known that the permutation automorphism group of the binary linear Hadamard code of length 2^m is the general affine group $\mathrm{AGL}(m, 2)$, which has order $2^m(2^m-1)\cdots(2^m-2^{m-1})$ [MS77, Chapter 13]. In [PPV14], the order of the permutation automorphism group for the quaternary linear Hadamard codes $\mathcal{H}_{\gamma,\delta}$ is established. These groups are completely characterized by computing the orbits of the action of $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ on $\mathcal{H}_{\gamma,\delta}$ and by giving generators of the group. Since the dual of a quaternary linear Hadamard code is an extended 1-perfect code in the quaternary sense, the permutation automorphism group of these codes is also computed. In [KV15], the order of the monomial automorphism group for $\mathbb{Z}_2\mathbb{Z}_4$-additive Hadamard codes and the permutation automorphism group of the corresponding $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes are given. The permutation automorphism of the $\mathbb{Z}_2\mathbb{Z}_4$-linear extended 1-perfect codes is studied in [Kro16] and the permutation automorphism group of the span of these codes in [PR02]. For even $m \geq 6$, the permutation automorphism group of the $\mathbb{Z}_4$-linear Kerdock code K_m is determined in [Car91]. The order of this group is $2^m(2^{m-1}-1)(m-1)$. A comprehensive study of the monomial automorphism group of quaternary linear Kerdock, Preparata, Delsarte-Goethals and Goethals-Delsarte codes, introduced in [HKC+94], can be found in [Wan97]. Finally, the reader is referred to [HP03] for basic properties of the permutation and monomial automorphism groups of codes over $\mathbb{Z}_2$, $\mathbb{Z}_4$ and $\mathbb{F}_q$, and examples of equivalent

codes.

2.6 Permutation decoding

Permutation decoding is a technique introduced in [Mac64] by MacWilliams for linear codes that involves finding a subset of the permutation automorphism group of a linear code in order to assist in decoding. The idea behind this technique is to move all the errors in a received vector out of the information positions by using a permutation that preserves the code.

Let C be a linear t-error-correcting code with information set I. The permutation decoding technique is suitable for linear codes with a fairly large permutation automorphism group since it is strongly based on the existence of a special subset of $\mathrm{PAut}(C)$. Specifically, a subset $S \subseteq \mathrm{PAut}(C)$ is said to be an s-PD-*set* with respect to the information set I for the code C if every s-set of coordinate positions is moved out of I by at least one element of S, where $1 \leq s \leq t$. When $s = t$, S is said to be a PD-*set*.

When dealing with permutation decoding we must answer two questions. First, how can we guarantee that the information positions in a vector are correct? And second, how do we find an s-PD-set for the code C, that is, a subset of $\mathrm{PAut}(C)$ that moves all the nonzero entries in every possible error vector of weight s or less out of the information positions? The first question is completely solved for linear codes in the following theorem, while the second one is only solved for some particular families of linear codes.

Theorem 9 ([MS77, Chapter 16]). *Let C be a linear t-error-correcting code with information set I and parity check matrix H in standard form (2.1). Suppose $y = x + e$, where $x \in C$ and $\mathrm{wt}(e) \leq t$. Then the information coordinates of y are correct if and only if $\mathrm{wt}(Hy^T) \leq t$.*

With this result in mind, we can formulate the permutation decoding algorithm. Assume we have successfully found a PD-set S with respect to the information set I for the linear code C.

1. If $\mathrm{wt}(Hy^T) \leq t$, then there is no error in the information positions of y and we can retrieve x as $y_I G$.

2. Else we search $\sigma \in S$ satisfying that $\mathrm{wt}(H\sigma(y)^T) \leq t$. If there is no such σ, then more than t errors have occurred.

3. If we find σ, then the information positions of $\sigma(y)$ are error-free. We take x' as the unique codeword such that $x'_I = \sigma(y)_I$ and the decoded vector is $x = \sigma^{-1}(x')$.

Regarding the second question announced above, in [FKM12], it is shown how to find s-PD-sets of size $s+1$ that satisfy the Gordon-Schönheim bound for partial permutation decoding for the binary simplex code of length 2^m-1 for all $m \geq 4$ and $1 < s \leq \left\lfloor \frac{2^m-m-1}{m} \right\rfloor$. In [KS16], the ideas behind the establishment of these PD-sets for binary simplex codes are applied to the MacDonald codes [Mac60], obtaining similar results. In [Sen09], 2-PD-sets of size 5 and 4-PD-sets of size $\binom{m+1}{2} + 2$ are found for binary linear Hadamard codes H_m for all $m > 4$. In [KMM10], the method introduced in [Sen09] is extended to find $(m-1)$-PD-sets of size $\frac{1}{2}(m^2+m+4)$ for H_m for all $m \geq 5$, and $(m+1)$-PD-sets of size $\frac{1}{6}(m^3+5m+12)$ for H_m for all $m \geq 6$. They also obtain $(m-3)$-PD-sets of size $\frac{1}{6}(m^3+5m+12)$ for the second order Reed-Muller code $R(2,m)$, for all $m \geq 8$. Small PD-sets that satisfy the Gordon-Schönheim bound have also been found for binary Golay codes [Gor82, Wol83] and for the binary simplex code S_4 [KV08]. In [BS11], a method for constructing information sets valid for semisimple Abelian codes is presented. In [BS13], the geometrical properties of those information sets are applied to obtain sufficient conditions for a t-error-correcting Abelian code to have a s-PD-set for all $s \leq t$.

The aforementioned permutation decoding algorithm does not work for nonlinear codes in general, since the condition stated in Theorem 9 becomes useless for these codes. An alternative permutation decoding algorithm suitable for $\mathbb{Z}_4$-linear codes is presented in [BBFV15] where Theorem 12 replaces Theorem 9.

For each quaternary coordinate position $i \in \{1,\ldots,\beta\}$, we denote by $\varphi_1(i)$ and $\varphi_2(i)$ the corresponding pair of binary coordinate positions in $\{1,\ldots,2\beta\}$, that is, $\varphi_1(i) = 2i-1$ and $\varphi_2(i) = 2i$. Let I_1 and I_2 be the following sets of size γ and 2δ, respectively:

$$I_1 = \{\varphi_1(\beta-\gamma-\delta+1),\ldots,\varphi_1(\beta-\delta)\},$$
$$I_2 = \{\varphi_1(\beta-\delta+1),\varphi_2(\beta-\delta+1),\ldots,\varphi_1(\beta),\varphi_2(\beta)\}.$$

Theorem 10 ([BBFV15]). *If C is a $\mathbb{Z}_4$-linear code of length 2β and type $2^\gamma 4^\delta$, then C is a systematic code. Moreover, if the generator matrix of $\mathcal{C} = \Phi^{-1}(C)$ is in standard form (2.4), then $I = I_1 \cup I_2$ is an information set for C.*

Given a information vector $a = (a_1, \ldots, a_{\gamma+2\delta}) \in \mathbb{Z}_2^{\gamma+2\delta}$, we consider the representation $a = (c, d)$, where $c = (a_1, \ldots, a_\gamma)$ and $d = (a_{\gamma+1}, \ldots, a_{\gamma+2\delta})$. For a quaternary vector $x = (x_1, \ldots, x_\ell)$, define $\Phi_1(x) = (\phi_1(x_1), \ldots, \phi_1(x_\ell))$, where ϕ_1 is the first coordinate of the Gray map. Let $\sigma : \mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta \to \mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$ be the bijection defined as

$$\sigma(\Phi^{-1}(a)) = \sigma(c, \Phi^{-1}(d)) = (c + \Phi_1(\Phi^{-1}(d)R), \Phi^{-1}(d)),$$

where $\Phi : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_4^{n_2} \to \mathbb{Z}_2^{n_1+2n_2}$ is the generalized Gray map, that is,

$$\Phi(x, y) = (x, \phi(y_1), \ldots, \phi(y_{n_2})),$$

for all $x \in \mathbb{Z}_2^{n_1}$ and $y = (y_1, \ldots, y_{n_2}) \in \mathbb{Z}_4^{n_2}$. The parameters n_1 and n_2 of the extended Gray map Φ will be treated dependently on the context.

Theorem 11 ([BBFV15]). *Let C be a $\mathbb{Z}_4$-linear code of length 2β, type $2^\gamma 4^\delta$ and such that the generator matrix $\mathcal{G}_S$ of $\mathcal{C} = \Phi^{-1}(C)$ is in standard form (2.4). Then the function $f : \mathbb{Z}_2^{\gamma+2\delta} \to \mathbb{Z}_2^{2\beta}$, defined as*

$$f(a) = \Phi(\sigma(\Phi^{-1}(a))\mathcal{G}_S),$$

is a systematic encoding for C and the information set $I = I_1 \cup I_2$.

Theorem 12 ([BBFV15]). *Let C be a binary systematic t-error-correcting code of length n. Let I be an information set for C and let f be a systematic encoding for C and I. Suppose $y = x + e$, where $x \in C$ and $\mathrm{wt}(e) \leq t$. Then the information coordinates of y are correct (and thus, $x_I = y_I$) if and only if $\mathrm{wt}(y + f(y_I)) \leq t$.*

Assume now we have found a PD-set S with respect to the information set I for the $\mathbb{Z}_4$-linear code C. The alternative permutation decoding process to the algorithm presented for linear codes is

1. If $\mathrm{wt}(y + f(y_I)) \leq t$, then there is no error in the information positions of y and we can retrieve x as $f(y_I)$.

2. Else we search $\sigma \in S$ satisfying that $\mathrm{wt}(\sigma(y) + f(\sigma(y)_I)) \leq t$. If there is no such σ, then more than t errors have happened.

3. If we find σ, then the information positions of $\sigma(y)$ are error-free. Then, the decoded vector is $x = \sigma^{-1}(f(\sigma(y)_I))$.

Example 13. *Let $\mathcal{C}$ (respectively, $\mathcal{C}_S$) be the quaternary linear code with generator matrix $\mathcal{G}$ (respectively, $\mathcal{G}_S$) considered in Example 4. Let C (respectively, C_S) be the binary image under the Gray map of $\mathcal{C}$ (respectively, $\mathcal{C}_S$). Denote the* 4-PD*-set of size* 5 *for C introduced in Example 51 by S. Since $\mathcal{C}$ is permutation equivalent to $\mathcal{C}_S$, by using the permutation*

$$(1, 14, 11, 8, 5, 16, 13, 10, 7, 4, 2, 15, 12, 9, 6, 3) \in \mathrm{Sym}(16),$$

it is clear that $\psi(C) = C_S$, where

$$\begin{aligned}\psi \;&= (1, 27, 21, 15, 9, 31, 25, 19, 13, 7, 3, 29, 23, 17, 11, 5)\\ &\quad (2, 28, 22, 16, 10, 32, 26, 20, 14, 8, 4, 30, 24, 18, 12, 6) \in \mathrm{Sym}(32).\end{aligned}$$

Then, one may take $S_S = \{\psi\sigma\psi^{-1} : \sigma \in S\} = \{\mathrm{id}, \overline{\sigma}_1, \overline{\sigma}_2, \overline{\sigma}_3, \overline{\sigma}_4\}$, where

$$\begin{aligned}\overline{\sigma}_1 = \;&(1, 29, 23, 21, 15, 13, 7, 5)(2, 30, 24, 22, 16, 14, 8, 6)\\ &(3, 31, 25, 27, 17, 19, 9, 11)(4, 32, 26, 28, 18, 20, 10, 12)\\ \overline{\sigma}_2 = \;&(1, 23, 11, 31)(2, 24, 12, 32)(3, 5, 13, 25)(4, 6, 14, 26)\\ &(7, 27, 19, 15)(8, 28, 20, 16)(9, 17, 21, 29)(10, 18, 22, 30),\\ \overline{\sigma}_3 = \;&(1, 21, 7, 3)(2, 22, 8, 4)(5, 23, 17, 15)(6, 24, 18, 16)\\ &(9, 31, 13, 27)(10, 32, 14, 28)(11, 25, 19, 29)(12, 26, 20, 30),\\ \overline{\sigma}_4 = \;&(1, 11)(2, 12)(3, 13)(4, 14)(5, 25)(6, 26)(7, 19)(8, 20)\\ &(9, 21)(10, 22)(15, 27)(16, 28)(17, 29)(18, 30)(23, 31)(24, 32),\end{aligned}$$

as a 4-PD*-set for C_S. Denote the ith row of the matrix (2.6) by $\mathbf{v}_i$. Let*

$$\begin{aligned}x \;&= \Phi(\mathbf{v}_3) = \Phi(0011122223333001)\\ &= (00000101011111111110101010000001).\end{aligned}$$

Suppose now that the received vector is $y = x + e$, where

$$y = (00000101010101011110101010000011),$$

that is, the nonzero coordinates of the error vector are placed in positions $J = \{11, 13, 15, 31\}$. The information set given by Theorem 10 is $I = I_2 = \{\varphi_1(14), \varphi_2(14), \varphi_1(15), \varphi_2(15), \varphi_1(16), \varphi_2(16)\} = \{27, 28, 29, 30, 31, 32\}$. Then, the information coordinates of the vector y respect to this information set I are $y_I = (000011)$. Since $\gamma = 0$, there is no matrix R and $\sigma = \mathrm{id}$. Then

$$\begin{aligned}f(y_I) \;&= \Phi(\sigma(\Phi^{-1}(y_I))\mathcal{G}_S) = \Phi((002)\mathcal{G}_S) = \Phi(2\mathbf{v}_3)\\ &= (000011111100000000011111111000011),\end{aligned}$$

so $\mathrm{wt}(y+f(y_I)) = 12 > 4 = s$. *There are errors in the information positions of* y. *Nevertheless, considering the vector*

$$z = \overline{\sigma}_1(y) = (010101011001011000111000111000000),$$

we obtain that $z_I = (100000)$ *and*

$$\begin{aligned} f(z_I) &= \Phi(\sigma(\Phi^{-1}(z_I))\mathcal{G}_S) = \Phi((300)\mathcal{G}_S) = \Phi(3\mathbf{v}_1) \\ &= (011101111001111000111000011000000), \end{aligned}$$

so $\mathrm{wt}(z + f(z_I)) \leq 4 = s$. *Thus, there is no error in the information positions of* $\sigma(y)$. *The decoded vector is* $\sigma^{-1}(f(\sigma(y)_I)) = \sigma^{-1}(f(z_I))$, *that is equal to* x. *Note that* $\overline{\sigma}_1$ *is the unique permutation in the set* S_S *such that* $\overline{\sigma}_1(J) \cap I \neq \emptyset$, *since* $\mathrm{id}(J) = J$, $\overline{\sigma}_1(J) = \{3, 7, 13, 25\}$, $\overline{\sigma}_2(J) = \{1, 7, 25, 31\}$, $\overline{\sigma}_3(J) = \{5, 13, 25, 27\}$ *and* $\overline{\sigma}_4(J) = \{1, 3, 23, 27\}$. *Then, an error pattern in the positions* $J = \{11, 13, 15, 31\}$ *can only be corrected by using* $\overline{\sigma}_1$.

Example 14. *Let* $\mathcal{C}$ *(respectively,* $\mathcal{C}_S$*) be the quaternary linear code with generator matrix* $\mathcal{G}$ *(respectively,* $\mathcal{G}_S$*) considered in Example 5. Let* C *(respectively,* C_S*) be the binary image under the Gray map of* $\mathcal{C}$ *(respectively,* $\mathcal{C}_S$*). Computations in* MAGMA *[BCFS16] shows that* $S_S = \{\mathrm{id}, \sigma_1, \sigma_2\}$, *where*

$$\begin{aligned} \sigma_1 &= (1, 9, 29, 5)(2, 10, 30, 6)(3, 27)(4, 28)(7, 23)(8, 24) \\ &\quad (11, 17, 15, 21)(12, 18, 16, 22)(13, 31)(14, 32)(19, 25)(20, 26), \\ \sigma_2 &= (1, 23, 29, 19)(2, 24, 30, 20)(3, 21, 31, 17)(4, 22, 32, 18) \\ &\quad (5, 27, 9, 13)(6, 28, 10, 14)(7, 15, 25, 11)(8, 16, 26, 12), \end{aligned}$$

is a 2-PD*-set of size* 3 *for* C_S. *Denote the* i*th row of order two of the matrix (2.8) by* $\mathbf{u}_i$ *and the* i*th row of order four of the same matrix by* $\mathbf{v}_i$. *Let*

$$\begin{aligned} x &= \Phi(\mathbf{u}_2) = \Phi(0000022222220200) \\ &= (000000000011111111111111100110000). \end{aligned}$$

Suppose now that the received vector is $y = x + e$, *where*

$$y = (000000000011111111\mathbf{0}1111100111\mathbf{1}000),$$

that is, the nonzero coordinates of the error vector are placed in positions $J = \{19, 29\}$. *The information set given by Theorem 10 is* $I = I_1 \cup I_2 = \{\varphi_1(13), \varphi_1(14)\} \cup \{\varphi_1(15), \varphi_2(15), \varphi_1(16), \varphi_2(16)\} = \{25, 27, 29, 30, 31, 32\}$.

Then, the information coordinates of the vector y respect to this information set I are $y_I = (011000)$. In this example, $\sigma \neq \mathrm{id}$ because $\gamma \neq 0$. Since

$$\begin{aligned}\sigma(\Phi^{-1}(y_I)) &= \sigma(0130)\\ &= ((0,1) + \Phi_1\left((3,0)\begin{pmatrix}1 & 1\\ 0 & 0\end{pmatrix}\right), 3, 0)\\ &= (1030),\end{aligned}$$

we obtain that

$$\begin{aligned}f(y_I) &= \Phi(\sigma(\Phi^{-1}(y_I))\mathcal{G}_S) = \Phi((1030)\mathcal{G}_S) = \Phi(\mathbf{u}_1 + 3\mathbf{v}_1)\\ &= (011111100000011101111000011010 00),\end{aligned}$$

so $\mathrm{wt}(y + f(y_I)) = 16 > 2 = s$. *There are errors in the information positions of y. Nevertheless, considering the vector*

$$z = \sigma_1(y) = (011111111100000000100000011001111),$$

we obtain that $z_I = (101111)$ and

$$\begin{aligned}f(z_I) &= \Phi(\sigma(\Phi^{-1}(z_I))\mathcal{G}_S) = \Phi((0122)\mathcal{G}_S) = \Phi(\mathbf{u}_2 + 2\mathbf{v}_1 + 2\mathbf{v}_2)\\ &= (111111111100000000000000011001111),\end{aligned}$$

so $\mathrm{wt}(z + f(z_I)) \leq 2 = s$. *Thus, there is no error in the information positions of $\sigma_1(y)$. The decoded vector is $\sigma_1^{-1}(f(\sigma_1(y)_I)) = \sigma_1^{-1}(f(z_I))$, that is equal to x. Again σ_1 is the unique permutation in the set S_S such that $\sigma_1(J) \cap I \neq \emptyset$, since* $\mathrm{id}(J) = J, \sigma_1(J) = \{5, 25\}$ *and* $\sigma_2(J) = \{1, 19\}$.

2.7 Minimum size of PD-sets

Recall that the results presented in this section are novel, hence they represent the first contributions of this dissertation.

There is a well-known bound on the minimum size of PD-sets for linear codes (based on the length, dimension and minimum distance of such codes) that can be adapted to systematic codes (not necessarily linear) easily. This bound is given by the next result, which is quoted and proved for linear codes in [Huf98]. In general, for systematic codes, we can follow the same proof since the linearity of the code is only used to guarantee that the code is systematic. We include the proof here for the convenience of the reader.

Proposition 15. *Let C be a systematic t-error-correcting code of length n and size $|C| = 2^k$. Let $r = n - k$ be the redundancy of C. If S is a* PD-*set for C, then*

$$|S| \geq \left\lceil \frac{n}{r} \left\lceil \frac{n-1}{r-1} \left\lceil \cdots \left\lceil \frac{n-t+1}{r-t+1} \right\rceil \ \cdots \right\rceil \right\rceil \right\rceil. \tag{2.14}$$

Proof. Let $S = \{\sigma_1, \ldots, \sigma_\ell\}$ be a PD-set for the systematic t-error-correcting code C. Let R be a check set for C, that is, a set of $n-k$ redundancy positions. Define R_i as the image of R under σ_i^{-1} for $i \in \{1, \ldots, \ell\}$. Since S is a PD-set, we can guarantee that every subset of coordinates of size $e \leq t$ is contained in at least one set R_i. In particular, the sets $R_1, \ldots, R_\ell$ cover all t-subsets of the n-set Ω of coordinate positions. Let $N(t, r, n)$ be the minimum number of r-subsets of the n-set Ω which cover all t-subsets of Ω. It is clear that $\ell \geq N(t, r, n)$. Let $R_1, \ldots, R_{N(t,r,n)}$ be a collection of r-subsets of Ω covering all t-subsets. Define the set $\chi = \{(R_i, \omega) : 1 \leq i \leq N(t, r, n),\ \omega \in R_i\}$. Note that $|\chi| = rN(t, r, n)$ since $|R_i| = r$, for all $i \in \{1, \ldots, \ell\}$. For each $\omega \in \Omega$, we define χ_ω to be the set $\{(R_i, \omega) \in \chi\}$. Notice that $|\chi_\omega| \geq N(t-1, r-1, n-1)$ because the sets obtained from R_i, where $(R_i, \omega) \in \chi_\omega$, by deleting ω produce a collection of $(r-1)$-subsets which cover all $(t-1)$-subsets of the $(n-1)$-set $\Omega \backslash \{\omega\}$. Hence, there are at least $nN(t-1, r-1, n-1)$ pairs in χ and we obtain that

$$N(t, r, n) = \frac{n}{r} N(t-1, r-1, n-1).$$

By repeated applications of this inequality together with the observation that $N(1, r, n) = \lceil n/r \rceil$, the result follows. □

As we have seen in Section 2.6, in [BBFV15], it is shown that $\mathbb{Z}_4$-linear codes are systematic and a systematic encoding is given for these codes. Therefore, the result given by Proposition 15 can be applied to any $\mathbb{Z}_4$-linear code, not necessarily linear.

The above inequality (2.14) is often called the *Gordon-Schönheim bound*. This bound can be adapted to s-PD-sets for all s up to the error-correcting capability of the code.

Recall that any binary Hadamard code of length n has $2n$ codewords and minimum Hamming distance $n/2$ [MS77, Chapter 2] (see Section 2.3). Note that the error-correcting capability of any binary linear or $\mathbb{Z}_4$-linear Hadamard code of length $n = 2^m$ is

$$t_m = \lfloor (d-1)/2 \rfloor = \left\lfloor (2^{m-1} - 1)/2 \right\rfloor = 2^{m-2} - 1.$$

Moreover, all these codes are systematic and have size $2n = 2^{m+1}$. Therefore, the right-hand side of the bound given by (2.14), for binary linear and $\mathbb{Z}_4$-linear Hadamard codes of length 2^m and for all $1 \leq s \leq t_m$, becomes

$$g_m(s) = \left\lceil \frac{2^m}{2^m - m - 1} \left\lceil \frac{2^m - 1}{2^m - m - 2} \left\lceil \cdots \left\lceil \frac{2^m - s + 1}{2^m - m - s} \right\rceil \right\rceil \cdots \right\rceil \right\rceil.$$

We compute the minimum value of $g_m(s)$ in the following lemma.

Lemma 16. *Let m be an integer, $m \geq 4$. For $1 \leq s \leq t_m$, $g_m(s) \geq s + 1$, where $t_m = 2^{m-2} - 1$ is the error-correcting capability of any binary linear and $\mathbb{Z}_4$-linear Hadamard code of length 2^m.*

Proof. We need to prove that $g_m(s) \geq s + 1$. This fact is clear, since the central term $\lceil (2^m - s + 1)/(2^m - m - s) \rceil = 2$, for all $s \in \{1, \ldots, 2^{m-2} - 1\}$, and in each stage of the ceiling function working from inside, $g_m(s)$ increases its value by at least 1. □

The smaller the size of the PD-set is, the more efficient permutation decoding becomes. We start studying the simple case, when we have that $g_m(s) = s+1$. For each binary linear and $\mathbb{Z}_4$-linear Hadamard code of length 2^m, $m \geq 4$, we define the integer

$$f_m = \max\{s \ : \ 2 \leq s, \ g_m(s) = s + 1\},$$

which represents the greater s in which we can find s-PD-sets of size $s+1$. The following result characterize this parameter from the value of m. Note that for $m = 3$, since the error-correcting capability is $t_3 = 1$, the permutation decoding becomes unnecessary and we do not take it into account in the results.

Lemma 17. *Let m be an integer, $m \geq 4$. Then, $f_m = \lfloor \frac{2^m - m - 1}{1+m} \rfloor = \lfloor \frac{2^m}{1+m} \rfloor - 1$.*

Proof. The result can be proved easily by Lemma 16 and an argument similar to the proof of Lemma 2 in [FKM12]. However, we include the detailed proof for the convenience of the reader.

There are s stages in the computation of the above formula for $g_m(s)$ and we have already proved in Lemma 16 that the central term of this formula is 2, that is, $\lceil (2^m - s + 1)/(2^m - m - s) \rceil = 2$, for all $m \geq 4$ and $1 \leq s \leq 2^{m-2} - 1$. Since we require that $g_m(s) = s + 1$, we must ensure that at each stage, that is, in each ceiling function, the increase is exactly 1. For the second term, working from inside, we must guarantee that

$$\left\lceil 2 \left(\frac{2^m - (s-1) + 1}{2^m - m - (s-1)} \right) \right\rceil = \left\lceil 2 + \frac{2m + 2}{2^m - m - (s-1)} \right\rceil = 3,$$

or, equivalently,

$$\frac{2m+2}{2^m-m-(s-1)} \leq 1.$$

Thus, the condition in the second step of the computation implies that $s \leq 2^m - 3m - 1$. For the third term, we obtain similarly that $s \leq 2^m - 4m - 1$ and for the lth we get

$$\left\lceil l \left(\frac{2^m-(s-l+1)+1}{2^m-m-(s-l+1)} \right) \right\rceil = l+1$$

if and only if $s \leq 2^m - (l+1)m - 1$. Now, taking $s = l$, we have that $s \leq 2^m - (s+1)m - 1$, thus $s(1+m) \leq 2^m - m - 1$ and finally

$$s \leq \left\lfloor \frac{2^m-m-1}{1+m} \right\rfloor.$$

Since $s \leq 2^m - (l+1)m - 1 \leq 2^m - lm - 1 \leq \cdots \leq 2^m - 3m - 1$, for all $l, m \geq 4$, the condition $s \leq 2^m - (l+1)m - 1$ is enough to obtain the greater s such that $g_m(s) = s+1$. Thus, $f_m = \left\lfloor \frac{2^m-m-1}{1+m} \right\rfloor$. □

The bound f_m holds for any systematic binary Hadamard code of length 2^m regardless of whether it is $\mathbb{Z}_4$-linear or not. This is the case, for example, of $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes [PRV06, RSV09].

Recall that the quaternary Kerdock code $\mathcal{K}_m$ is a quaternary linear code of length 2^{m-1}, type 4^m and minimum Lee distance $2^{m-1} - 2^{\lfloor (m-1)/2 \rfloor}$ [HKC+94] (see Section 2.4). Then $K_m = \Phi(\mathcal{K}_m)$ is a binary code of length $n = 2^m$ and size $|K_m| = 2^{2m}$. The error-correcting capability of the binary Kerdock code K_m is

$$t_m = \lfloor (d-1)/2 \rfloor = 2^{m-2} - 2^{\lfloor (m-1)/2 \rfloor - 1} - 1.$$

Thus, the right-hand side of the bound given by (2.14) becomes for K_m

$$g_m(s) = \left\lceil \frac{2^{m+1}}{2^{m+1}-2m} \left\lceil \frac{2^{m+1}-1}{2^{m+1}-2m-1} \left\lceil \cdots \left\lceil \frac{2^{m+1}-s+1}{2^{m+1}-2m-s+1} \right\rceil \right\rceil \cdots \right\rceil \right\rceil.$$

Again, we can define the integer $f_m = \max\{s \ : \ 2 \leq s, \ g_m(s) = s+1\}$ for the binary Kerdock code of length $n = 2^m$, $m \geq 4$. From now on, the integers t_m and f_m will be treated depending on the context and they will represent, respectively, the error-correcting capability and the bound for the existence of s-PD-sets of size $s+1$ for binary Hadamard codes and Kerdock codes of length 2^m. The following lemmas state equivalent results for binary Kerdock codes to the ones obtained from Lemmas 16 and 17 for systematic binary Hadamard codes.

Lemma 18. *Let m be an integer, $m \geq 4$. For $1 \leq s \leq t_m$, $g_m(s) \geq s+1$, where $t_m = 2^{m-2} - 2^{\lfloor (m-1)/2 \rfloor - 1} - 1$ is the error-correcting capability of the binary Kerdock code K_m of length 2^m.*

Lemma 19. *Let m be an integer, $m \geq 4$. Then, $f_m = \lfloor \frac{2^{m-1}}{m} \rfloor - 1$.*

Tables 2.1 and 2.2 show the values of the bound f_m for the existence of s-PD-sets of size $s+1$ and the error-correcting capability t_m for systematic binary Hadamard codes and binary Kerdock codes of length 2^m with $4 \leq m \leq 13$, respectively.

An argument similar to the one used in the proofs of Lemmas 16 and 17 can be used to show that for any systematic binary code C of length n and size $|C| = 2^k$, $s \leq \lfloor \frac{n}{k} \rfloor - 1$ if S is an s-PD-set of size $s+1$ for C.

m	f_m	t_m	m	f_m	t_m
4	2	3	9	50	127
5	4	7	10	92	255
6	8	15	11	169	511
7	15	31	12	314	1023
8	27	63	13	584	2047

Table 2.1: Values of f_m and t_m for systematic binary Hadamard codes of length 2^m with $4 \leq m \leq 13$.

m	f_m	t_m	m	f_m	t_m
4	1	2	9	27	119
5	2	5	10	50	247
6	4	13	11	92	495
7	8	27	12	169	1007
8	15	59	13	314	2015

Table 2.2: Values of f_m and t_m for the binary Kerdock code of length 2^m with $4 \leq m \leq 13$.

Chapter 3

PD-sets for binary linear Hadamard codes

This chapter aims to provide s-PD-sets of minimum size to perform partial permutation decoding for the binary linear Hadamard code H_m of length $2^m, m \geq 4$. With this aim in mind, in Section 3.1, we regard the permutation automorphism group of H_m as a certain subgroup of the general linear group $\mathrm{GL}(m+1,2)$, and we provide a criterion on subsets of matrices of this subgroup to be an s-PD-set of minimum size $s+1$ for H_m. In Section 3.2, using this criterion we give explicit constructions of s-PD-sets of size $s+1$ for all $m \geq 4$ and $2 \leq s \leq f_m = \lfloor \frac{2^m-m-1}{1+m} \rfloor$ for these codes. Finally, in Section 3.3, we define two recursive constructions to obtain s-PD-sets of size $l \geq s+1$ for H_{m+1} from an s-PD-set of the same size for H_m.

As we have already mentioned in Section 2.6, in [Sen09], 2-PD-sets of size 5 and 4-PD-sets of size $\binom{m+1}{2} + 2$ are found for binary linear Hadamard codes H_m for all $m \geq 5$. In [KMM10], the method introduced in [Sen09] is extended to find $(m-1)$-PD-sets of size $\frac{1}{2}(m^2+m+4)$ for H_m for all $m \geq 5$, and $(m+1)$-PD-sets of size $\frac{1}{6}(m^3+5m+12)$ for H_m for all $m \geq 6$. Note that no s-PD-set at all is provided for H_4 and neither of the ones presented for H_m, $m \geq 5$, is of minimum size. In [FKM12], s-PD-sets of minimum size $s+1$ for the binary simplex code of length 2^m-1 for all $m \geq 4$ and $1 < s \leq \lfloor \frac{2^m-m-1}{m} \rfloor$ are given. In this chapter, we follow a similar argument to the one introduced for simplex codes in [FKM12] to obtain s-PD-sets of minimum size $s+1$ for H_m, $m \geq 4$.

The results given in Sections 3.1 and 3.3 are already published in our paper [BV14], although without proofs, and were presented at "IX Jornadas de Matemática Discreta y Algorítmica" in Tarragona, Spain, 2014. All results

presented in this chapter, including the proofs, are going to be published in [BV16a] together with the results given in Chapter 4 and some results from Section 2.7. Finally, also mention that Theorem 28 has also been proved independently [KMM16], as it was pointed out to us by one of the referees in the review process of [BV16a].

3.1 First criterion to find s-PD-sets of size $s+1$

For any $m \geq 2$, there is a unique binary linear Hadamard code H_m of length 2^m [MS77]. Recall that a generator matrix G_m for H_m can be constructed as follows:

$$G_m = \begin{pmatrix} 1 & \mathbf{1} \\ \mathbf{0} & G' \end{pmatrix},$$

where G' is any matrix having as column vectors the $2^m - 1$ nonzero vectors from $\mathbb{Z}_2^m$, with the vectors e_i, $i \in \{1, \ldots, m\}$, in the first m positions.

By construction, from (2.9), it is clear that $I_m = \{1, \ldots, m+1\}$ is an information set for H_m. Let w_i be the ith column vector of G_m, $i \in \{1, \ldots, 2^m\}$. By labelling the coordinate positions with the columns of G_m, we can take as an information set I_m for H_m the first $m+1$ column vectors of G_m considered as row vectors, that is, $I_m = \{w_1, \ldots, w_{m+1}\} = \{e_1, e_1 + e_2, \ldots, e_1 + e_{m+1}\}$. Then, depending on the context, I_m will be taken as a subset of $\{1, \ldots, 2^m\}$ or $\{1\} \times \mathbb{Z}_2^m$.

Example 20. *Let H_4 be the binary linear Hadamard code of length 16 with generator matrix (2.10) given in Example 6. The set $I_4 = \{1, 2, 3, 4, 5\}$, or equivalently the set of column vectors $I_4 = \{w_1, w_2, w_3, w_4, w_5\} = \{e_1, e_1 + e_2, e_1 + e_3, e_1 + e_4, e_1 + e_5\}$ of G_4, is an information set for H_4.*

It is known that the permutation automorphism group $\mathrm{PAut}(H_m)$ of H_m is isomorphic to the general affine group $\mathrm{AGL}(m, 2)$ [MS77, Chapter 13]. Let $\mathrm{GL}(m, 2)$ be the general linear group over $\mathbb{Z}_2$. Recall that $\mathrm{AGL}(m, 2)$ consists of all mappings $\alpha : \mathbb{Z}_2^m \to \mathbb{Z}_2^m$ of the form $\alpha(x) = Ax + b$ for $x \in \mathbb{Z}_2^m$, where $A \in \mathrm{GL}(m, 2)$ and $b \in \mathbb{Z}_2^m$, together with the function composition as the group operation. The monomorphism

$$\begin{array}{rccc} \varphi : & \mathrm{AGL}(m, 2) & \longrightarrow & \mathrm{GL}(m+1, 2) \\ & (b, A) & \longmapsto & \begin{pmatrix} 1 & b \\ \mathbf{0} & A \end{pmatrix} \end{array}$$

defines an isomorphism between $\mathrm{AGL}(m, 2)$ and the subgroup of $\mathrm{GL}(m+1, 2)$ consisting of all nonsingular matrices whose first column is e_1. Therefore,

from now on, we also regard $\mathrm{PAut}(H_m)$ as this subgroup. Note that any matrix $M \in \mathrm{PAut}(H_m)$ can be seen as a permutation of coordinate positions, that is, as an element of $\mathrm{Sym}(2^m)$. By multiplying each column vector w_i of G_m by M, we obtain another column vector $w_j = w_i M$, which means that the ith coordinate position moves to the jth coordinate position, $i, j \in \{1, \ldots, 2^m\}$.

Example 21. *Let M be the matrix*

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \in \mathrm{PAut}(H_2).$$

The generator matrix of the binary linear Hadamard code H_2 constructed as in (2.9) is

$$G_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Since the columns of G_2 are $w_1 = (1,0,0)$, $w_2 = (1,1,0)$, $w_3 = (1,0,1)$ and $w_4 = (1,1,1)$, we have that $w_1 M = w_1$, $w_2 M = w_3$, $w_3 M = w_2$ and $w_4 M = w_4$, so the permutation associated to $M \in \mathrm{PAut}(H_2) \subseteq \mathrm{GL}(3,2)$ is $\sigma = (2,3) \in \mathrm{PAut}(H_2) \subseteq \mathrm{Sym}(4)$.

Example 22. *Let M be the matrix*

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \in \mathrm{PAut}(H_4),$$

and let G_4 be the generator matrix of the binary linear Hadamard code H_4 introduced in Example 6. One can easily check that $w_i M = w_{\sigma(i)}$, where $\sigma = (1,14,11,9,6,10,13,3,15,5,16,2,12,8)(4,7) \in \mathrm{PAut}(H_4) \subseteq \mathrm{Sym}(16)$.

Let $M \in \mathrm{PAut}(H_m)$ and let m_i be the ith row of M, $i \in \{1, \ldots, m+1\}$. We define M^* as the matrix where the first row is m_1 and the ith row is $m_1 + m_i$, $i \in \{2, \ldots, m+1\}$.

An s-PD-set of size $s+1$ for H_m meets the Gordon-Schönheim bound if $2 \leq s \leq f_m$. The following theorem provides us a condition on sets of matrices of $\mathrm{PAut}(H_m)$ in order to be s-PD-sets of size $s+1$ for H_m.

Theorem 23. *Let H_m be the binary linear Hadamard code of length 2^m, with $m \geq 4$. Let $P_s = \{M_i \ : \ 0 \leq i \leq s\}$ be a set of $s+1$ matrices in* $\mathrm{PAut}(H_m)$. *Then, P_s is an s-*PD*-set of size $s+1$ for H_m with information set I_m if and only if no two matrices $(M_i^{-1})^*$ and $(M_j^{-1})^*$ for $i \neq j$ have a row in common. Moreover, any subset $P_k \subseteq P_s$ of size $k+1$ is a k-*PD*-set for $k \in \{1, \ldots, s\}$.*

Proof. Suppose that the set $P_s = \{M_i \ : \ 0 \leq i \leq s\}$ satisfies that no two matrices $(M_i^{-1})^*$ and $(M_j^{-1})^*$ for $i \neq j$ have a row in common. Let $E = \{v_1, \ldots, v_s\} \subseteq \{1\} \times \mathbb{Z}_2^m$ be a set of s different column vectors of the generator matrix G_m regarded as row vectors, which represents a set of s error positions. Assume we cannot move all the error positions to the check set by any element of P_s. Then, for each $i \in \{0, \ldots, s\}$, there is a $v \in E$ such that $vM_i \in I_m$. In other words, there is at least one error position that remains in the information set I_m after applying any permutation of P_s. Note that there are $s+1$ values for i, but only s elements in E. Therefore, $vM_i \in I_m$ and $vM_j \in I_m$ for some $v \in E$ and $i \neq j$. Suppose $vM_i = w_r$ and $vM_j = w_t$, for $w_r, w_t \in I_m$. Then, $v = w_r M_i^{-1} = w_t M_j^{-1}$. Taking into account the form of the vectors in the information set $I_m = \{w_1, \ldots, w_{m+1}\}$, by multiplying for such inverse matrices M_i^{-1} and M_j^{-1}, we get the first row or a certain addition between the first row and another row of each matrix. Thus, we obtain that $(M_i^{-1})^*$ and $(M_j^{-1})^*$ have a row in common, contradicting our assumption. Let $P_k \subseteq P_s$ of size $k+1$. If this set satisfies the condition on the inverse matrices and we suppose that it is not a k-PD-set, we arrive at a contradiction in the same way as before.

Conversely, suppose that the set $P_s = \{M_i \ : \ 0 \leq i \leq s\}$ forms an s-PD-set for H_m, but does not satisfy the condition on the inverse matrices. Thus, some $v \in \{w_1, \ldots, w_{2^m}\}$ must be the rth row of $(M_i^{-1})^*$ and the tth row of $(M_j^{-1})^*$ for some $r, t \in \{1, \ldots, m+1\}$, $i, j \in \{0, \ldots, s\}$. In other words, we have that $v = e_r(M_i^{-1})^* = e_t(M_j^{-1})^*$. Therefore, $v = w_r M_i^{-1} = w_t M_j^{-1}$, where $w_r, w_t \in I_m$, and thus we obtain that $vM_i = w_r$ and $vM_j = w_t$. These equalities implies that the vector v, which represents an error position, cannot be moved to the check set by the permutations defined by the matrices M_i and M_j. Let $L = \{l \ : \ 0 \leq l \leq s, \ l \neq i, j\}$. For each $l \in L$, choose a row v_l of $(M_l^{-1})^*$. It is clear that $v_l = e_t(M_l^{-1})^* = w_t M_l^{-1}$, so $v_l M_l = w_t \in I_m$. Finally, since some of the v_l may repeat, we obtain a set $E = \{v_l \ : \ l \in L\} \cup \{v\}$ of size at most s. Nevertheless, no matrix in P_s will map every member of E into the check set, a fact that contradicts our assumption. □

Example 24. *The set of matrices* $P_2 = \{\mathrm{Id}_5, M_1, M_2\}$, *where*

$$M_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \qquad M_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and Id_5 *is the* 5×5 *identity matrix, is a* 2-PD-*set of size* 3 *for the binary linear Hadamard code* H_4 *of length* 16 *with generator matrix (2.10) given in Example 6, by Theorem 23. We now compute the matrices* $\mathrm{Id}_5^*, (M_1^{-1})^*$ *and* $(M_2^{-1})^*$ *to check the condition given in Theorem 23. First, note that* $P_2 \subseteq \mathrm{PAut}(H_4) \subseteq \mathrm{GL}(5,2)$. *The inverse matrices of* M_1 *and* M_2 *are*

$$M_1^{-1} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad \textit{and} \quad M_2^{-1} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

respectively. Then, it is straightforward to check that the matrices Id_5^*,

$$(M_1^{-1})^* = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \textit{and} \quad (M_2^{-1})^* = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

have no rows in common.

In addition, note that no s-PD-*set of size* $s+1$ *can be found for* $s \geq 3$ *since* $f_4 = 2$. *Notice also that* $f_4 = 2 < 3 = t_4$, *where* t_4 *is the error-correcting capability of* H_4. *Finally, also remark that the matrices of* P_2 *can be regarded as elements of* $\mathrm{Sym}(16)$. *In this case, we obtain* $\{\mathrm{id}, \sigma_1, \sigma_2\}$ *as a* 2-PD-*set, where*

$$\sigma_1 = (1, 14, 11, 9, 6, 10, 13, 3, 15, 5, 16, 2, 12, 8)(4, 7),$$
$$\sigma_2 = (1, 14, 11, 2, 7, 9, 5, 12, 3, 16, 13, 6)(4, 15, 8, 10).$$

Let S be an s-PD-set of size $s + 1$. The set S is a *nested* s-PD-set if there is an ordering of the elements of S, $S = \{\sigma_0, \ldots, \sigma_s\}$, such that $S_i = \{\sigma_0, \ldots, \sigma_i\} \subseteq S$ is an i-PD-set of size $i + 1$, for all $i \in \{0, \ldots, s\}$. Note that $S_i \subset S_j$ if $0 \leq i < j \leq s$ and $S_s = S$.

From Theorem 23, we have two important consequences. The first one is related to how to obtain nested s-PD-sets and the second one provides another proof of Lemma 17.

Corollary 25. *Let m be an integer, $m \geq 4$. If P_s is an s-PD-set of size $s+1$ for the binary linear Hadamard code H_m, then any ordering of the elements of P_s gives nested k-PD-sets for $k \in \{1, \ldots, s\}$.*

Corollary 26. *Let m be an integer, $m \geq 4$. If P_s is an s-PD-set of size $s+1$ for the binary linear Hadamard code H_m, then $s \leq f_m = \left\lfloor \frac{2^m - m - 1}{1+m} \right\rfloor$.*

Proof. Following the condition on sets of matrices to be s-PD-sets of size $s+1$, given by Theorem 23, we have to obtain certain $s+1$ matrices with no rows in common. Note that the number of possible vectors of length $m+1$ over $\mathbb{Z}_2$ with 1 in the first coordinate is 2^m. Thus, taking this fact into account and counting the number of rows of each one of these $s+1$ matrices, we have that $(s+1)(m+1) \leq 2^m$, so $s+1 \leq \frac{2^m}{m+1}$ and finally $s \leq f_m$. □

3.2 Explicit construction of s-PD-sets of size $s+1$

In this section, by using Theorem 23, we give an explicit construction of s-PD-sets of minimum size $s+1$ for H_m, for all $m \geq 4$ and $2 \leq s \leq f_m$. We follow a similar technique to the one described for simplex codes in [FKM12].

Lemma 27. *Let $K = \mathbb{Z}_2[x]/(f(x))$, where $f(x) \in \mathbb{Z}_2[x]$ is a primitive polynomial of degree m. Let $\alpha \in K$ be a root of $f(x)$. Then $\alpha^{i+1} - \alpha^i, \ldots, \alpha^{i+m} - \alpha^i$ are linearly independent over $\mathbb{Z}_2$ for all $i \in \{0, \ldots, 2^m - 2\}$.*

Proof. The elements $\alpha^{i+1} - \alpha^i, \ldots, \alpha^{i+m} - \alpha^i$ are linearly independent over $\mathbb{Z}_2$ for all $i \in \{0, \ldots, 2^m - 2\}$, if and only if $\alpha - 1, \ldots, \alpha^m - 1$ are linearly independent over $\mathbb{Z}_2$. Suppose that there are $\lambda_j \in \mathbb{Z}_2$ such that $\sum_{j=1}^{m} \lambda_j(\alpha^j - 1) = 0$. Then, multiplying by α^i, we obtain that $\sum_{j=1}^{m} \lambda_j(\alpha^{i+j} - \alpha^i) = 0$, so $\lambda_j = 0$ for all $j \in \{1, \ldots, m\}$, since $\{\alpha^{i+1} - \alpha^i, \ldots, \alpha^{i+m} - \alpha^i\}$ are linearly independent. Analogously, assume that there are $\lambda_j \in \mathbb{Z}_2$ such that $\sum_{j=1}^{m} \lambda_j(\alpha^{i+j} - \alpha^i) = \alpha^i[\sum_{j=1}^{m} \lambda_j(\alpha^j - 1)] = 0$. Since $\alpha^i \in K \backslash \{0\}$, it follows that $\sum_{j=1}^{m} \lambda_j(\alpha^j - 1) = 0$, so $\lambda_j = 0$ for all $j \in \{1, \ldots, m\}$, since $\{\alpha - 1, \ldots, \alpha^m - 1\}$ are linearly independent.

Note that $\alpha^m - 1 = \sum_{j=1}^{m-1} \mu_j \alpha^j$ for some μ_j. Moreover, this summation has an odd number of nonzero terms, since $f(x)$ is irreducible. Let $\mu =$

$(\mu_1, \ldots, \mu_{m-1}) \in \mathbb{Z}_2^{m-1}$. Note that in vectorial notation $\alpha^j - 1 = e_1 + e_j$, $j \in \{1, \ldots, m-1\}$, and $\alpha^m - 1 = \sum_{j=1}^{m-1} \mu_j e_{j+1}$. Finally, it is easy to see that the $m \times m$ binary matrix

$$\begin{pmatrix} \mathbf{1} & \mathrm{Id}_{m-1} \\ 0 & \mu \end{pmatrix},$$

which has as rows $\alpha - 1, \ldots, \alpha^m - 1$, has determinant $\sum_{j=1}^{m-1} \mu_j = 1 \neq 0$. □

For $i \in \{1, \ldots, f_m\}$, we consider the $(m+1) \times (m+1)$ binary matrices

$$N_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & \alpha^{m-1} \end{pmatrix} \quad \text{and} \quad N_i = \begin{pmatrix} 1 & \alpha^{(m+1)i-1} \\ 0 & \alpha^{(m+1)i} - \alpha^{(m+1)i-1} \\ \vdots & \vdots \\ 0 & \alpha^{(m+1)i+m-1} - \alpha^{(m+1)i-1} \end{pmatrix}.$$

Theorem 28. *Let $P_s = \{M_i : 0 \leq i \leq s\}$, where $M_i = N_i^{-1}$. Then, P_s is an s-PD-set of size $s+1$ for the binary linear Hadamard code H_m of length 2^m with information set I_m, for all $m \geq 4$ and $2 \leq s \leq f_m$.*

Proof. Clearly, $N_0 \in \mathrm{PAut}(H_m)$, since it is the identity matrix. By Lemma 27, $N_i \in \mathrm{PAut}(H_m)$ for all $i \in \{1, \ldots, f_m\}$. The rows of the matrices $N_0^*, \ldots, N_{f_m}^*$ are the elements of the form $(1, a)$, for all $a \in \{0, 1, \alpha, \ldots, \alpha^{f_m(m+1)+m-1}\}$, which are all different, since α is primitive and $f_m(m+1) + m - 1 \leq 2^m - 2$. By Theorem 23, the result follows. □

Example 29. *Let H_4 be the binary linear Hadamard code of length 16 with generator matrix (2.10) given in Example 6. Let $K = \mathbb{Z}_2[x]/(x^4 + x + 1)$ and α a root of $x^4 + x + 1$. Recall that $f_4 = 2$. Let N_0, N_1 and N_2 be the following matrices:*

$$N_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \alpha \\ 0 & \alpha^2 \\ 0 & \alpha^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$N_1 = \begin{pmatrix} 1 & \alpha^4 \\ 0 & \alpha^5 - \alpha^4 \\ 0 & \alpha^6 - \alpha^4 \\ 0 & \alpha^7 - \alpha^4 \\ 0 & \alpha^8 - \alpha^4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix},$$

$$N_2 = \begin{pmatrix} 1 & \alpha^9 \\ 0 & \alpha^{10}-\alpha^9 \\ 0 & \alpha^{11}-\alpha^9 \\ 0 & \alpha^{12}-\alpha^9 \\ 0 & \alpha^{13}-\alpha^9 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

The matrices $N_0 = \mathrm{Id}_5$, *where* Id_5 *is the* 5×5 *identity matrix,* N_1 *and* N_2 *are elements of* $\mathrm{PAut}(H_4) \subseteq \mathrm{GL}(5,2)$, *thus they are the matrices* $M_0 = N_0^{-1}$, $M_1 = N_1^{-1}$ *and* $M_2 = N_2^{-1}$. *By Theorem 28,* $P_2 = \{N_0^{-1}, N_1^{-1}, N_2^{-1}\}$ *is a* 2-PD-*set of size* 3 *for* H_4, *where* $N_0^{-1} = \mathrm{Id}_5$,

$$N_1^{-1} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad N_2^{-1} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

It is straightforward to check that the matrices $N_0^* = \mathrm{Id}_5^*$,

$$N_1^* = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad N_2^* = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

have no rows in common. In case we need to consider P_2 *as a subset of* $\mathrm{Sym}(16)$, *the corresponding set of permutations is* $\{\mathrm{id}, \sigma_1, \sigma_2\}$, *where*

$$\begin{aligned} \sigma_1 &= (1,16,6)(2,12,14,9,4,7)(3,13,10,5,15,8), \\ \sigma_2 &= (1,16,11)(2,14,4,9,7,12)(3,10,15,5,8,13). \end{aligned}$$

Example 30. *Let* H_5 *be the binary linear Hadamard code of length 32 with generator matrix constructed as in (2.9). Let* $K = \mathbb{Z}_2[x]/(x^5+x^2+1)$ *and* α *a root of* x^5+x^2+1. *In this case, the matrices* N_i, $0 \le i \le f_5 = 4$, *are* $N_0 = \mathrm{Id}_6$,

$$N_1 = \begin{pmatrix} 1 & \alpha^5 \\ 0 & \alpha^6-\alpha^5 \\ 0 & \alpha^7-\alpha^5 \\ 0 & \alpha^8-\alpha^5 \\ 0 & \alpha^9-\alpha^5 \\ 0 & \alpha^{10}-\alpha^5 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix},$$

$$N_2 = \begin{pmatrix} 1 & \alpha^{11} \\ 0 & \alpha^{12}-\alpha^{11} \\ 0 & \alpha^{13}-\alpha^{11} \\ 0 & \alpha^{14}-\alpha^{11} \\ 0 & \alpha^{15}-\alpha^{11} \\ 0 & \alpha^{16}-\alpha^{11} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

$$N_3 = \begin{pmatrix} 1 & \alpha^{17} \\ 0 & \alpha^{18}-\alpha^{17} \\ 0 & \alpha^{19}-\alpha^{17} \\ 0 & \alpha^{20}-\alpha^{17} \\ 0 & \alpha^{21}-\alpha^{17} \\ 0 & \alpha^{22}-\alpha^{17} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix},$$

$$N_4 = \begin{pmatrix} 1 & \alpha^{23} \\ 0 & \alpha^{24}-\alpha^{23} \\ 0 & \alpha^{25}-\alpha^{23} \\ 0 & \alpha^{26}-\alpha^{23} \\ 0 & \alpha^{27}-\alpha^{23} \\ 0 & \alpha^{28}-\alpha^{23} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

By Theorem 28, $P_4 = \{N_i^{-1} : 0 \leq i \leq 4\}$ is a 4-PD*-set of size* 5 *for H_5. In [Sen09, KMM10], the* 4-PD*-set provided for H_5 has size* 17.

3.3 Recursive constructions of s-PD-sets

In this section, given an s-PD-set of size l for the binary linear Hadamard code H_m of length 2^m, where $l \geq s+1$, we show how to construct recursively an s-PD-set of the same size for $H_{m'}$ of length $2^{m'}$ for all $m' > m$.

Given a matrix $M \in \mathrm{PAut}(H_m)$ and an integer $\kappa \geq 1$, we define the matrix $M(\kappa) \in \mathrm{PAut}(H_{m+\kappa})$ as

$$M(\kappa) = \begin{pmatrix} M & \mathbf{0} \\ \mathbf{0} & \mathrm{Id}_\kappa \end{pmatrix},$$

where Id_κ denotes the $\kappa \times \kappa$ identity matrix.

Proposition 31. *Let m be an integer, $m \geq 4$, and let $P_s = \{M_i : 0 \leq i \leq s\}$ be an s-PD-set of size $s+1$ for H_m with information set I_m. Then, $Q_s = \{(M_i^{-1}(\kappa))^{-1} : 0 \leq i \leq s\}$ is an s-PD-set of size $s+1$ for $H_{m+\kappa}$ with information set $I_{m+\kappa}$, for any $\kappa \geq 1$.*

Proof. Since P_s is an s-PD-set for H_m, matrices $(M_1^{-1})^*, \ldots, (M_s^{-1})^*$ have no rows in common by Theorem 23. Therefore, it is straightforward to check that matrices $(M_1^{-1}(\kappa))^*, \ldots, (M_s^{-1}(\kappa))^*$ have no rows in common either. Moreover, $M_i^{-1}(\kappa) \in \mathrm{PAut}(H_{m+\kappa})$, for all $i \in \{1, \ldots, s\}$. Thus, applying again Theorem 23, we have that Q_s is an s-PD-set for $H_{m+\kappa}$. □

Note that the bound f_{m+1} for H_{m+1} cannot be achieved recursively from an s-PD-set for H_m, since the recursive construction given by Proposition 31 works for a given fixed s, while increasing the length of the Hadamard code.

Example 32. *Let* $P_2 = \{\mathrm{Id}_5, M_1, M_2\}$ *be the* 2-PD-*set of size* 3 *for* H_4*, given in Example 24. Then matrices* $M_1^{-1}(1)$ *and* $M_2^{-1}(1)$ *are*

$$M_1^{-1}(1) = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } M_2^{-1}(1) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

It is clear that matrices Id_6^*,

$$(M_1^{-1}(1))^* = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \text{ and}$$

$$(M_2^{-1}(1))^* = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

have no rows in common. Therefore, by Proposition 31, we have that the set $Q_2 = \{\mathrm{Id}_6, (M_1^{-1}(1))^{-1}, (M_2^{-1}(1))^{-1}\}$ *is a* 2-PD-*set of size* 3 *for the binary linear Hadamard code* H_5 *of length* 32*, constructed as in (2.9).*

Note that $f_5 = 4$*, as shown in Table 2.1. However, we cannot obtain an* s*-PD-set of size* $s+1$ *for* H_5 *with* $3 \leq s \leq 4$ *recursively by applying Propostion 31. Theorem 23 may be used to achieve the* 4-PD-*set of size* 5 *for* H_5*, as viewed in Example 30.*

The above recursive construction only holds when the size of the s-PD-set is exactly $s+1$. Next, we show a second recursive construction which holds when the size of the s-PD-set is any integer l, $l \geq s+1$, unlike the first recursive construction. Now, the elements of $\mathrm{PAut}(H_m)$ will be regarded as permutations of coordinate positions, that is, as elements of $\mathrm{Sym}(2^m)$ instead of matrices of $\mathrm{GL}(m+1,2)$.

It is well known that a generator matrix G_{m+1} for the binary linear Hadamard code H_{m+1} of length 2^{m+1} can be constructed as follows:

$$G_{m+1} = \begin{pmatrix} G_m & G_m \\ \mathbf{0} & \mathbf{1} \end{pmatrix}, \tag{3.1}$$

where G_m is a generator matrix for the binary linear Hadamard code H_m of length 2^m.

Given two permutations $\sigma_1 \in \mathrm{Sym}(n_1)$ and $\sigma_2 \in \mathrm{Sym}(n_2)$, we define $(\sigma_1|\sigma_2) \in \mathrm{Sym}(n_1+n_2)$, where σ_1 acts on the coordinates $\{1,\ldots,n_1\}$ and σ_2 on $\{n_1+1,\ldots,n_1+n_2\}$.

Proposition 33. *Let m be an integer, $m \geq 4$, and S be an s-PD-set of size l for H_m with information set I. Then, $(S|S) = \{(\sigma|\sigma) : \sigma \in S\}$ is an s-PD-set of size l for H_{m+1} constructed from (3.1), with any information set $I' = I \cup \{i+2^m\}$, $i \in I$.*

Proof. Since I is an information set for H_m, we have that $|(H_m)_I| = 2^{m+1}$. Since H_{m+1} is constructed from (3.1), it follows that $H_{m+1} = \{(x,x),(x,\bar{x}) : x \in H_m\}$, where $\bar{x}$ is the complementary vector of x. A vector and its complementary have different values in each coordinate, so $|(H_{m+1})_{I\cup\{i\}})| = 2^{m+2}$, for all $i \in \{2^m+1,\ldots,2^{m+1}\}$. Thus, any set of the form $I' = I\cup\{i+2^m\}$, $i \in I$, is an information set for H_{m+1}.

If $\sigma \in \mathrm{PAut}(H_m)$, then $\sigma(x) = z \in H_m$ for all $x \in H_m$. Therefore, since $(\sigma|\sigma)(x,x) = (z,z)$ and $(\sigma|\sigma)(x,\bar{x}) = (z, z+\sigma(\mathbf{1})) = (z,\bar{z})$, we can conclude that $(\sigma|\sigma) \in \mathrm{PAut}(H_{m+1})$.

Let $e = (a,b) \in \mathbb{Z}_2^{2n}$, where $a = (a_1,\ldots,a_n)$, $b = (b_1,\ldots,b_n) \in \mathbb{Z}_2^n$, and $n = 2^m$. Finally, we will prove that for every $e \in \mathbb{Z}_2^{2n}$ with $\mathrm{wt}(e) \leq s$, there is $(\sigma|\sigma) \in (S|S)$ such that $(\sigma|\sigma)(e)_{I'} = \mathbf{0}$. Let $c = (c_1,\ldots,c_n)$ be the binary vector defined as follows: $c_i = 1$ if and only if $a_i = 1$ or $b_i = 1$, for all $i \in \{1,\ldots,n\}$. Note that $\mathrm{wt}(c) \leq s$, since $\mathrm{wt}(e) \leq s$. Taking into account that S is an s-PD-set with respect to I, there is $\sigma \in S$ such that $\sigma(c)_I = \mathbf{0}$. Therefore, we also have that $(\sigma|\sigma)(a,b)_{I\cup J} = \mathbf{0}$, where $J = \{i+n : i \in I\}$. The result follows trivially since $I' \subseteq I \cup J$. □

Example 34. *Let* $S = \{\mathrm{id}, \sigma_1, \sigma_2\}$ *be the* 2-PD-*set of size* 3 *for* H_4*, given in Example 29. By Proposition 33, the set* $(S|S) = \{\mathrm{id}, (\sigma_1|\sigma_1), (\sigma_2|\sigma_2)\}$*, where*

$$\begin{aligned}(\sigma_1|\sigma_1) = \;& (1,16,6)(2,12,14,9,4,7)(3,13,10,5,15,8)\\ & (17,32,22)(18,28,30,25,20,23)(19,29,26,21,31,24),\\ (\sigma_2|\sigma_2) = \;& (1,16,11)(2,14,4,9,7,12)(3,10,15,5,8,13)\\ & (17,32,27)(18,30,20,25,23,28)(19,26,31,21,24,29),\end{aligned}$$

is a 2-PD-*set of size* 3 *for the binary linear Hadamard code* H_5 *constructed from (3.1) by using the generator matrix (2.10), with any information set of the form* $I' = \{1,2,3,4,5,i\}$*,* $i \in \{17,\ldots,21\}$*. Note that* H_5 *is generated by*

$$G_5 = \begin{pmatrix} G_4 & G_4 \\ \mathbf{0} & \mathbf{1} \end{pmatrix}, \tag{3.2}$$

where G_4 *is the generator matrix (2.10) of* H_4*.*

Example 35. *Let* S *be a* 3-PD-*set for the binary linear Hadamard code* H_4 *with information set* I*. In fact, we can just say that such a set* S *is a* PD-*set for* H_4 *since* $t_4 = 3$*. Moreover, since* $f_4 = 2$*, the size of* S *must be greater than* 4*. Indeed, by Propisition 15,*

$$|S| \geq \left\lceil \frac{16}{11} \left\lceil \frac{15}{10} \left\lceil \frac{14}{10} \right\rceil \right\rceil \right\rceil = 5.$$

Proposition 33 can be used to produce a 3-PD-*set for any* H_m *constructed by using (3.1) recursively from the given* 3-PD-*set* S *for* H_4*. Nevertheless, Proposition 31 becomes useless in this case. A* PD-*set* S *of size 16 for the binary linear Hadamard code* H_4 *can be found in [BBFV15, Example 3].*

Chapter 4

PD-sets for $\mathbb{Z}_4$-linear Hadamard codes

This chapter establishes similar results for (nonlinear) $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ of length 2^m and type $2^\gamma 4^\delta$, where $m = \gamma + 2\delta - 1$, to those presented in Chapter 3 for binary linear Hadamard codes. In Section 4.1, we regard the permutation automorphism group of $\mathcal{H}_{\gamma,\delta}$ as a certain subset, denoted by $\pi(\mathcal{L})$, of the general linear group $\mathrm{GL}(\gamma+\delta, 4)$ and we provide a criterion on subsets of matrices of $\pi(\mathcal{L})$ to obtain an s-PD-set of minimum size $s+1$ for $H_{\gamma,\delta}$. In Section 4.2, using this criterion we give explicit constructions of s-PD-sets of size $s+1$ for all $\delta \geq 3$ and $2 \leq s \leq \lfloor \frac{2^{2\delta-2}-\delta}{\delta} \rfloor \leq f_m$ for these codes. Finally, in Section 4.3, we define two recursive constructions to acquire s-PD-sets of size $l \geq s+1$ for $H_{\gamma+i,\delta+j}$ of length 2^{m+i+2j} and type $2^{\gamma+i}4^{\delta+j}$ for all $i, j \geq 0$ from a given s-PD-set of the same size for $H_{\gamma,\delta}$.

It is worth mentioning that the s-PD-sets constructed in this chapter represent the first ones suitable for applying permutation decoding to nonlinear codes. The results given in Sections 4.1 and 4.3 were presented at "21st Conference on Applications of Computer Algebra" in Kalamata, Greece, 2015 [BV15]. All results presented in this chapter, including the proofs, are going to be published in [BV16a] together with the results given in Chapter 3 and some results from Section 2.7.

4.1 First criterion to find s-PD-sets of size $s+1$

Let $\mathcal{H}_{\gamma,\delta}$ be the quaternary linear Hadamard code of length $\beta = 2^{m-1}$ and type $2^\gamma 4^\delta$, where $m = \gamma + 2\delta - 1$, and let $H_{\gamma,\delta} = \Phi(\mathcal{H}_{\gamma,\delta})$ be the corresponding $\mathbb{Z}_4$-linear code of length $2\beta = 2^m$. Recall that a generator matrix $\mathcal{G}_{\gamma,\delta}$ for $\mathcal{H}_{\gamma,\delta}$

can be constructed by using the following recursive constructions, starting from $\mathcal{G}_{0,1} = (1)$:

$$\mathcal{G}_{\gamma+1,\delta} = \begin{pmatrix} \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} \\ \mathbf{0} & \mathbf{2} \end{pmatrix},$$
$$\mathcal{G}_{\gamma,\delta+1} = \begin{pmatrix} \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} & \mathcal{G}_{\gamma,\delta} \\ \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \end{pmatrix},$$

An ordered set $\mathcal{I} = \{i_1, \ldots, i_{\gamma+\delta}\} \subseteq \{1, \ldots, \beta\}$ of $\gamma + \delta$ coordinate positions is said to be a *quaternary information set* for a quaternary linear code $\mathcal{C}$ of type $2^\gamma 4^\delta$ if $|\mathcal{C}_\mathcal{I}| = 2^\gamma 4^\delta$. If the elements of $\mathcal{I}$ are ordered in such a way that $|\mathcal{C}_{\{i_1,\ldots,i_\delta\}}| = 4^\delta$, then it is easy to see that the set $\Phi(\mathcal{I})$, defined as

$$\begin{aligned} \Phi(\mathcal{I}) &= \{\varphi_1(i_1), \varphi_2(i_1), \ldots, \varphi_1(i_\delta), \varphi_2(i_\delta)\} \cup \{\varphi_1(i_{\delta+1}), \ldots, \varphi_1(i_{\delta+\gamma})\} \\ &= \{2i_1 - 1, 2i_1, \ldots, 2i_\delta - 1, 2i_\delta, 2i_{\delta+1} - 1, \ldots, 2i_{\delta+\gamma} - 1\}, \end{aligned}$$

is an information set for $C = \Phi(\mathcal{C})$.

Example 36. *The set $\mathcal{I} = \{1\}$ is a quaternary information set for $\mathcal{H}_{0,1}$, so $\Phi(\mathcal{I}) = \{1, 2\}$ is an information set for $H_{0,1} = \Phi(\mathcal{H}_{0,1})$.*

For the quaternary linear code $\mathcal{H}_{2,2}$ generated by the matrix (2.7) given in Example 5, the set $\mathcal{I} = \{1, 2, 5, 9\}$ is a quaternary information set and $|(\mathcal{H}_{2,2})_{\{1,2\}}| = 4^2$, so $\Phi(\mathcal{I}) = \{1, 2, 3, 4\} \cup \{9, 17\}$ is an information set for $H_{2,2} = \Phi(\mathcal{H}_{2,2})$.

In general, there is not a unique way to obtain a quaternary information set for $\mathcal{H}_{\gamma,\delta}$. The following result provides a recursive and simple form to obtain such a set.

Proposition 37. *Let $\mathcal{I}$ be a quaternary information set for the quaternary linear Hadamard code $\mathcal{H}_{\gamma,\delta}$ of length $\beta = 2^{m-1}$ and type $2^\gamma 4^\delta$, where $m = \gamma + 2\delta - 1$. Then $\mathcal{I} \cup \{\beta + 1\}$ is a quaternary information set for the codes $\mathcal{H}_{\gamma+1,\delta}$ and $\mathcal{H}_{\gamma,\delta+1}$, which are obtained from $\mathcal{H}_{\gamma,\delta}$ by applying (2.11) and (2.12), respectively.*

Proof. Since $|\mathcal{H}_{\gamma+1,\delta}| = 2^{\gamma+1}4^\delta$ and $|\mathcal{H}_{\gamma,\delta+1}| = 2^\gamma 4^{\delta+1}$, it is clear that a quaternary information set for codes $\mathcal{H}_{\gamma+1,\delta}$ and $\mathcal{H}_{\gamma,\delta+1}$ should have $\gamma+\delta+1 = |\mathcal{I}|+1$ coordinate positions.

Taking into account that $\mathcal{H}_{\gamma,\delta+1}$ is constructed from (2.12), we have that $\mathcal{H}_{\gamma,\delta+1} = \{(u, u, u, u), (u, u + \mathbf{1}, u + \mathbf{2}, u + \mathbf{3}), (u, u + \mathbf{2}, u, u + \mathbf{2}), (u, u + \mathbf{3}, u + \mathbf{2}, u + \mathbf{1}) : u \in \mathcal{H}_{\gamma,\delta}\}$. Vectors u, $u + \mathbf{1}$, $u + \mathbf{2}$, and $u + \mathbf{3}$ have different values

in each coordinate, so $|(\mathcal{H}_{\gamma,\delta+1})_{\mathcal{I}\cup\{i\}}| = 2^\gamma 4^{\delta+1}$ for all $i \in \{\beta+1,\ldots,2\beta,3\beta+1,\ldots,4\beta\}$. In particular, $\mathcal{I}\cup\{\beta+1\}$ is a quaternary information set for $\mathcal{H}_{\gamma,\delta+1}$.

A similar argument holds for $\mathcal{H}_{\gamma+1,\delta}$. Since $\mathcal{H}_{\gamma+1,\delta}$ is constructed from (2.11), we have that $\mathcal{H}_{\gamma+1,\delta} = \{(u,u),(u,u+\mathbf{2}) : u \in \mathcal{H}_{\gamma,\delta}\}$. Vectors u and $u+\mathbf{2}$ have different values in each coordinate, so $|(\mathcal{H}_{\gamma+1,\delta})_{\mathcal{I}\cup\{i\}}| = 2^{\gamma+1}4^\delta$ for all $i \in \{\beta+1,\ldots,2\beta\}$. Therefore, $\mathcal{I}\cup\{\beta+1\}$ is a quaternary information set for $\mathcal{H}_{\gamma+1,\delta}$. □

Although the quaternary information set $\mathcal{I}\cup\{\beta+1\}$, given by Proposition 37, is the same for $\mathcal{H}_{\gamma+1,\delta}$ and $\mathcal{H}_{\gamma,\delta+1}$, the information set for the corresponding binary codes $H_{\gamma+1,\delta}$ and $H_{\gamma,\delta+1}$ are different, $I' = \Phi(\mathcal{I})\cup\{2\beta+1\}$ and $I'' = \Phi(\mathcal{I})\cup\{2\beta+1,2\beta+2\}$, respectively.

As for binary linear codes, we can label the ith coordinate position of a quaternary linear code $\mathcal{C}$, with the ith column of a generator matrix $\mathcal{G}$ of $\mathcal{C}$. Thus, any quaternary information set $\mathcal{I}$ for $\mathcal{C}$ can also be considered as a set of vectors representing the positions in $\mathcal{I}$. Then, by Proposition 37, we have that the set $\mathcal{I}_{\gamma,\delta} = \{e_1, e_1+e_2,\ldots,e_1+e_\delta, e_1+2e_{\delta+1},\ldots,e_1+2e_{\gamma+\delta}\}$ is a suitable quaternary information set for $\mathcal{H}_{\gamma,\delta}$. Depending on the context, $\mathcal{I}_{\gamma,\delta}$ will be considered as a subset of $\{1,\ldots,\beta\}$ or $\{1\}\times\mathbb{Z}_4^{\delta-1}\times\{0,2\}^\gamma$.

Example 38. *Let $\mathcal{H}_{0,3}$ be the quaternary linear Hadamard code of length 16 with generator matrix*

$$\mathcal{G}_{0,3} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \end{pmatrix},$$

obtained by applying (2.12) two times starting from $\mathcal{G}_{0,1} = (1)$. The set $\mathcal{I}_{0,3} = \{1,2,5\}$, or equivalently the set of vectors $\mathcal{I}_{0,3} = \{e_1, e_1+e_2, e_1+e_3\} = \{(1,0,0),(1,1,0),(1,0,1)\}$, is a quaternary information set for $\mathcal{H}_{0,3}$.

By applying (2.11) and (2.12) over $\mathcal{G}_{0,3}$, we obtain matrices $\mathcal{G}_{1,3}$ and $\mathcal{G}_{0,4}$ that generate the quaternary linear Hadamard codes $\mathcal{H}_{1,3}$ and $\mathcal{H}_{0,4}$ of length 32 and 64, respectively. By Proposition 37, it follows that $\mathcal{I}_{0,3}\cup\{17\} = \{1,2,5,17\}$ is a quaternary information set for $\mathcal{H}_{1,3}$ and $\mathcal{H}_{0,4}$. Although the quaternary information set is the same for both codes $\mathcal{H}_{1,3}$ and $\mathcal{H}_{0,4}$, it is important to note that in terms of vectors representing these positions, we have that

$$\begin{aligned} \mathcal{I}_{1,3} &= \{e_1, e_1+e_2, e_1+e_3, e_1+2e_4\} \\ &= \{(1,0,0,0),(1,1,0,0),(1,0,1,0),(1,0,0,2)\} \end{aligned}$$

and

$$\begin{aligned}\mathcal{I}_{0,4} &= \{e_1, e_1+e_2, e_1+e_3, e_1+e_4\}\\ &= \{(1,0,0,0),(1,1,0,0),(1,0,1,0),(1,0,0,1)\}.\end{aligned}$$

Finally, $I' = \Phi(\mathcal{I}_{0,3}) \cup \{33\} = \{1,2,3,4,9,10,33\}$ *and* $I'' = \Phi(\mathcal{I}_{0,3}) \cup \{33,34\} = \{1,2,3,4,9,10,33,34\}$ *are information sets for the* $\mathbb{Z}_4$*-linear Hadamard codes* $H_{1,3}$ *and* $H_{0,4}$*, respectively.*

Let $\mathcal{C}$ be a quaternary linear code of length β and type $2^\gamma 4^\delta$, and let $C = \Phi(\mathcal{C})$ be the corresponding $\mathbb{Z}_4$-linear code of length 2β. Let $\Phi : \mathrm{Sym}(\beta) \to \mathrm{Sym}(2\beta)$ be the map defined as

$$\Phi(\tau)(i) = \begin{cases} 2\tau(i/2), & \text{if } i \text{ is even,}\\ 2\tau((i+1)/2) - 1 & \text{if } i \text{ is odd,}\end{cases}$$

for all $\tau \in \mathrm{Sym}(\beta)$ and $i \in \{1,\ldots,2\beta\}$. Given a subset $\mathcal{S} \subseteq \mathrm{Sym}(\beta)$, we define the set $\Phi(\mathcal{S}) = \{\Phi(\tau) : \tau \in \mathcal{S}\} \subseteq \mathrm{Sym}(2\beta)$. It is easy to see that if $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{C}) \subseteq \mathrm{Sym}(\beta)$, then $\Phi(\mathcal{S}) \subseteq \mathrm{PAut}(C) \subseteq \mathrm{Sym}(2\beta)$.

Lemma 39. *The map* $\Phi : \mathrm{Sym}(\beta) \to \mathrm{Sym}(2\beta)$ *is a group monomorphism.*

Proof. We need to check that $\Phi(\sigma\tau) = \Phi(\sigma)\Phi(\tau)$ for all $\tau, \sigma \in \mathrm{PAut}(\mathcal{C})$. If i is even, it follows that $(\Phi(\sigma)\Phi(\tau))(i) = \Phi(\sigma)(2\tau(i/2)) = 2\sigma\left((2\tau(i/2))/2\right) = 2\sigma\tau(i/2) = \Phi(\sigma\tau)(i)$. Otherwise, $(\Phi(\sigma)\Phi(\tau))(i) = \Phi(\sigma)\left(2\tau\left((i+1)/2\right) - 1\right) = 2\sigma\tau((i+1)/2) - 1 = \Phi(\sigma\tau)(i)$. Finally, it is easy to check that Φ is injective. □

Let $\mathrm{GL}(k, \mathbb{Z}_4)$ denote the general linear group of degree k over $\mathbb{Z}_4$ and let $\mathcal{L}$ be the set consisting of all matrices over $\mathbb{Z}_4$ of the following form:

$$\begin{pmatrix} 1 & \eta & 2\theta \\ \mathbf{0} & A & 2X \\ \mathbf{0} & Y & B \end{pmatrix},$$

where $A \in \mathrm{GL}(\delta-1, \mathbb{Z}_4), B \in \mathrm{GL}(\gamma, \mathbb{Z}_4)$, X is a matrix over $\mathbb{Z}_4$ of size $(\delta-1) \times \gamma$, Y is a matrix over $\mathbb{Z}_4$ of size $\gamma \times (\delta-1)$, $\eta \in \mathbb{Z}_4^{\delta-1}$ and $\theta \in \mathbb{Z}_4^{\gamma}$.

Lemma 40. *The set* $\mathcal{L}$ *is a subgroup of* $\mathrm{GL}(\gamma+\delta, \mathbb{Z}_4)$.

Proof. We first need to check that $\mathcal{L} \subseteq \mathrm{GL}(\gamma+\delta, \mathbb{Z}_4)$, i.e., that $\det(\mathcal{M}) \in \{1,3\}$ for all $\mathcal{M} \in \mathcal{L}$. Note that if $\mathcal{M}' \in \mathrm{GL}(k, \mathbb{Z}_4)$, then $\mathcal{M} = \mathcal{M}' +$

$2\mathcal{R} \in \mathrm{GL}(k, \mathbb{Z}_4)$ for any $\mathcal{R}$. Thus, since $\det(\mathcal{M}') \in \{1, 3\}$, we have that $\det(\mathcal{M}) \in \{1, 3\}$, where

$$\mathcal{M}' = \begin{pmatrix} 1 & \eta & \mathbf{0} \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & Y & B \end{pmatrix}.$$

It is straightforward to check that $\mathcal{MN} \in \mathcal{L}$ for all $\mathcal{M}, \mathcal{N} \in \mathcal{L}$. □

Let ζ be the map from $\mathbb{Z}_4$ to $\mathbb{Z}_4$ defined as $\zeta(0) = \zeta(2) = 0, \zeta(1) = \zeta(3) = 1$. This map can be extended to matrices over $\mathbb{Z}_4$ by applying ζ to each one of their entries. Let π be the map from $\mathcal{L}$ to $\mathcal{L}$ defined as

$$\pi(\mathcal{M}) = \begin{pmatrix} 1 & \eta & 2\theta \\ \mathbf{0} & A & 2X \\ \mathbf{0} & \zeta(Y) & \zeta(B) \end{pmatrix},$$

and let $\pi(\mathcal{L}) = \{\pi(\mathcal{M}) : \mathcal{M} \in \mathcal{L}\} \subseteq \mathrm{GL}(\gamma + \delta, \mathbb{Z}_4)$. By Lemma 40, it is clear that $\pi(\mathcal{L})$ is a group with the operation $*$ defined as $\mathcal{M} * \mathcal{N} = \pi(\mathcal{MN})$ for all $\mathcal{M}, \mathcal{N} \in \pi(\mathcal{L})$. By the proof of Theorem 2 in [KV15], it is easy to see that the permutation automorphism group $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ is isomorphic to $\pi(\mathcal{L})$. Thus, from now on, we identify $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ with this group.

Recall that we can label the ith coordinate position of $\mathcal{H}_{\gamma,\delta}$ with the ith column vector w_i of the generator matrix $\mathcal{G}_{\gamma,\delta}$ constructed via (2.11) and (2.12), $i \in \{1, \ldots, \beta\}$. Therefore, again, any matrix $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ can be seen as a permutation of coordinate positions $\tau \in \mathrm{Sym}(\beta)$, such that $\tau(i) = j$ as long as $w_j = w_i\mathcal{M}$, $i, j \in \{1, \ldots, \beta\}$. For any $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$, we define $\Phi(\mathcal{M}) = \Phi(\tau) \in \mathrm{Sym}(2\beta)$, and for any $\mathcal{P} \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$, we consider $\Phi(\mathcal{P}) = \{\Phi(\mathcal{M}) : \mathcal{M} \in \mathcal{P}\} \subseteq \mathrm{Sym}(2\beta)$.

Proposition 41 ([PPV14, KV15])**.** *Let $\mathcal{H}_{\gamma,\delta}$ be the quaternary linear Hadamard code of length $2^{\gamma+\delta-2}$ and type $2^\gamma 4^\delta$. The order of its permutation automorphism group is*

$$|\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})| = 2^{\frac{3(\delta-1)^2}{2}+\frac{3(\delta-1)}{2}+2\gamma(\delta-1)+\frac{\gamma^2}{2}+\frac{\gamma}{2}} \prod_{i=1}^{\gamma}(2^i-1) \prod_{j=1}^{\delta-1}(2^j-1). \quad (4.1)$$

Proof. The formula (4.1) can be deduced from the one computed for the order of the monomial automorphism group $\mathrm{MAut}(\mathcal{H}_{\gamma,\delta})$ in [KV15]. A counting argument in the group $\pi(\mathcal{L})$ also reveals (4.1). The cardinality of the group $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ can also be computed by the recursive formula given in [PPV14]. □

Lemma 42. *Let $\mathcal{H}_{\gamma,\delta}$ be the quaternary linear Hadamard code of length β and type $2^\gamma 4^\delta$ and let $\mathcal{P} \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$. Then, $\Phi(\mathcal{P})$ is an s-PD-set for $H_{\gamma,\delta}$ with information set $\Phi(\mathcal{I}_{\gamma,\delta})$ if and only if for every s-set $\mathcal{E}$ of column vectors of $\mathcal{G}_{\gamma,\delta}$ there is $\mathcal{M} \in \mathcal{P}$ such that $\{g\mathcal{M} : g \in \mathcal{E}\} \cap \mathcal{I}_{\gamma,\delta} = \emptyset$.*

Proof. If $\Phi(\mathcal{P})$ is an s-PD-set with respect to the information set $\Phi(\mathcal{I}_{\gamma,\delta})$, then for every s-set $E \subseteq \{1,\ldots,2\beta\}$, there is $\tau \in \mathcal{P} \subseteq \mathrm{Sym}(\beta)$ such that $\Phi(\tau)(E) \cap \Phi(\mathcal{I}_{\gamma,\delta}) = \emptyset$. For every s-set $\mathcal{E} \subseteq \{1,\ldots,\beta\}$, let $E_o = \{2i-1 : i \in \mathcal{E}\}$. We know that there is $\tau \in \mathcal{P}$ such that $\Phi(\tau)(E_o) \cap \Phi(\mathcal{I}_{\gamma,\delta}) = \emptyset$. By the definition of Φ, we also have that $\tau(\mathcal{E}) \cap \mathcal{I}_{\gamma,\delta} = \emptyset$, which is equivalent to the statement.

Conversely, we assume that for every s-set $\mathcal{E} \subseteq \{1,\ldots,\beta\}$, there is $\tau \in \mathcal{P} \subseteq \mathrm{Sym}(\beta)$ such that $\tau(\mathcal{E}) \cap \mathcal{I}_{\gamma,\delta} = \emptyset$. For every s-set $E \subseteq \{1,\ldots,2\beta\}$, let $\mathcal{E}_o$ be an s-set such that $\{i : \varphi_1(i) \in E \text{ or } \varphi_2(i) \in E\} \subseteq \mathcal{E}_o$, where $\varphi_1(i) = 2i-1$ and $\varphi_2(i) = 2i$. Since there is $\tau \in \mathcal{P}$ such that $\tau(\mathcal{E}_o) \cap \mathcal{I}_{\gamma,\delta} = \emptyset$, we have that $\Phi(\tau)(E) \cap \Phi(\mathcal{I}_{\gamma,\delta}) = \emptyset$. □

A slight modification of the proof of Lemma 42 leads to a more general result that holds for any quaternary linear code, not only for the family of quaternary linear Hadamard codes.

Proposition 43. *Let $\mathcal{C}$ be a quaternary linear code and let $\mathcal{I}$ be a quaternary information set for $\mathcal{C}$. Suppose that $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{C})$. Then $\mathcal{S}$ satisfies that for each s-set $\mathcal{E} \subseteq \{1,\ldots,\beta\}$ there is $\tau \in \mathcal{S}$ such that $\tau(\mathcal{E}) \cap \mathcal{I} = \emptyset$ if and only if $\Phi(\mathcal{S}) \subseteq \mathrm{PAut}(C)$ satisfies that for each s-set $E \subseteq \{1,\ldots,2\beta\}$ there is $\sigma \in \Phi(\mathcal{S})$ such that $\sigma(E) \cap \Phi(\mathcal{I}) = \emptyset$.*

Let $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ and let m_i be the ith row of $\mathcal{M}$, $i \in \{1,\ldots,\delta+\gamma\}$. We define $\mathcal{M}^*$ as the matrix where the first row is m_1 and the ith row is $m_1 + m_i$ for $i \in \{2,\ldots,\delta\}$ and $m_1 + 2m_i$ for $i \in \{\delta+1,\ldots,\delta+\gamma\}$.

Theorem 44. *Let $\mathcal{H}_{\gamma,\delta}$ be the quaternary linear Hadamard code of type $2^\gamma 4^\delta$. Let $\mathcal{P}_s = \{\mathcal{M}_i : 0 \le i \le s\}$ be a set of $s+1$ matrices in $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$. Then, $\Phi(\mathcal{P}_s)$ is an s-PD-set of size $s+1$ for $H_{\gamma,\delta}$ with information set $\Phi(\mathcal{I}_{\gamma,\delta})$ if and only if no two matrices $(\mathcal{M}_i^{-1})^*$ and $(\mathcal{M}_j^{-1})^*$ for $i \neq j$ have a row in common.*

Proof. The result can be proved easily by Lemma 42 and an argument similar to the proof of Theorem 23. However, we include the detailed proof for the convenience of the reader.

Suppose that the set $\mathcal{P}_s = \{\mathcal{M}_i : 0 \le i \le s\}$ satisfies that no two matrices $(\mathcal{M}_i^{-1})^*$ and $(\mathcal{M}_j^{-1})^*$ for $i \neq j$ have a row in common. Let $\mathcal{E} =$

$\{g_1, \ldots, g_s\} \subseteq \{1\} \times \mathbb{Z}_4^{\delta-1} \times \{0,2\}^\gamma$ be a set of s different column vectors of the generator matrix $\mathcal{G}_{\gamma,\delta}$. Assume that $\Phi(\mathcal{P}_s)$ is not an s-PD-set for $H_{\gamma,\delta}$ with information set $\Phi(\mathcal{I}_{\gamma,\delta})$. By Lemma 42, it follows that for each $i \in \{0, \ldots, s\}$, there is a $g \in \mathcal{E}$ such that $g\mathcal{M}_i \in \mathcal{I}_{\gamma,\delta} \subseteq \{1\} \times \mathbb{Z}_4^{\delta-1} \times \{0,2\}^\gamma$. Note that there are $s+1$ values for i, but only s elements in $\mathcal{E}$. Therefore, $g\mathcal{M}_i \in \mathcal{I}_{\gamma,\delta}$ and $g\mathcal{M}_j \in \mathcal{I}_{\gamma,\delta}$ for some $g \in \mathcal{E}$ and $i \neq j$. Suppose $g\mathcal{M}_i = w_r$ and $g\mathcal{M}_j = w_t$, for $w_r, w_t \in \mathcal{I}_{\gamma,\delta}$. Then, $g = w_r\mathcal{M}_i^{-1} = w_t\mathcal{M}_j^{-1}$. Taking into account the form of the vectors in the information set $\mathcal{I}_{\gamma,\delta}$, by multiplying for such inverse matrices $\mathcal{M}_i^{-1}$ and $\mathcal{M}_j^{-1}$, we get the first row or a certain addition between the first row and another row of each matrix. Thus, we obtain that $(\mathcal{M}_i^{-1})^*$ and $(\mathcal{M}_j^{-1})^*$ have a row in common, contradicting our assumption. Let $\mathcal{P}_k \subseteq \mathcal{P}_s$ of size $k+1$. If this set satisfies the condition on the inverse matrices and we suppose that it is not a k-PD-set, we arrive to a contradiction in the same way as before.

Conversely, suppose that $\Phi(\mathcal{P}_s)$ is an s-PD-set for $H_{\gamma,\delta}$ with information set $\Phi(\mathcal{I}_{\gamma,\delta})$, but does not satisfy the condition on the inverse matrices. Thus, some $g \in \{1\} \times \mathbb{Z}_4^{\delta-1} \times \{0,2\}^\gamma$ must be the rth row of $(\mathcal{M}_i^{-1})^*$ and the tth row of $(\mathcal{M}_j^{-1})^*$ for some $r, t \in \{1, \ldots, \gamma+\delta\}$, $i, j \in \{0, \ldots, s\}$. In other words, we have that $g = e_r(\mathcal{M}_i^{-1})^* = e_t(\mathcal{M}_j^{-1})^*$ Therefore, $g = w_r\mathcal{M}_i^{-1} = w_t\mathcal{M}_j^{-1}$, where $w_r, w_t \in \mathcal{I}_{\gamma,\delta}$. Finally, we obtain that $g\mathcal{M}_i = w_r$ and $g\mathcal{M}_j = w_t$. Let $L = \{l \ : \ 0 \leq l \leq s, \ l \neq i, j\}$. For each $l \in L$, choose a row g_l of the matrix $(\mathcal{M}_l^{-1})^*$. It is clear that $g_l = e_t(\mathcal{M}_l^{-1})^* = w_t\mathcal{M}_l^{-1}$, so $g_l\mathcal{M}_l = w_t \in \mathcal{I}_{\gamma,\delta}$. Finally, since some of the g_l may repeat, we obtain a set $\mathcal{E} = \{g_l \ : \ l \in L\} \cup \{g\}$ of size at most s. Nevertheless, no matrix in $\mathcal{P}_s$ will map every member of $\mathcal{E}$ out of the quaternary information set $\mathcal{I}_{\gamma,\delta}$, fact that contradicts our assumption by Lemma 42. □

Corollary 45. *Let $\mathcal{P}_s$ be a set of $s+1$ matrices in $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$. If $\Phi(\mathcal{P}_s)$ is an s-PD-set of size $s+1$ for $H_{\gamma,\delta}$, then any ordering of elements in $\Phi(\mathcal{P}_s)$ provides nested k-PD-sets for $k \in \{1, \ldots, s\}$.*

Corollary 46. *Let $\mathcal{P}_s$ be a set of $s+1$ matrices in $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$. If $\Phi(\mathcal{P}_s)$ is an s-PD-set of size $s+1$ for $H_{\gamma,\delta}$, then $s \leq f_{\gamma,\delta}$, where*

$$f_{\gamma,\delta} = \left\lfloor \frac{2^{\gamma+2\delta-2} - \gamma - \delta}{\gamma+\delta} \right\rfloor.$$

Proof. Following the condition on sets of matrices to be s-PD-sets of size $s+1$, given by Theorem 44, we have to obtain certain $s+1$ matrices with no rows in common. Since the rows of length $\delta+\gamma$ must have 1 in the first

coordinate, and elements from $\{0, 2\}$ in the last γ coordinates, the number of possible rows is $4^{\delta-1}2^{\gamma} = 2^{\gamma+2\delta-2}$. Thus, taking this fact into account and counting the number of rows of each one of these $s+1$ matrices, we have that $(s+1)(\gamma+\delta) \leq 2^{\gamma+2\delta-2}$, and the result follows. □

Table 4.1 shows the values of the upper bound $f_{\gamma,\delta}$, along with the values of the bounds f_m and t_m for $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$, with $0 \leq \gamma \leq 5$ and $3 \leq \delta \leq 6$. Note that $f_{\gamma,\delta} \leq f_m \leq t_m$, where $m = \gamma + 2\delta - 1$.

Example 47. *Let* $\mathcal{P}_6 = \{\mathrm{Id}_4, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_6\} \subseteq \mathrm{PAut}(\mathcal{H}_{1,3})$, *where*

$$\mathcal{M}_1 = \begin{pmatrix} 1 & 1 & 2 & 2 \\ 0 & 3 & 1 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad \mathcal{M}_2 = \begin{pmatrix} 1 & 1 & 3 & 0 \\ 0 & 2 & 1 & 2 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \mathcal{M}_3 = \begin{pmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 3 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix},$$

$$\mathcal{M}_4 = \begin{pmatrix} 1 & 1 & 3 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad \mathcal{M}_5 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \mathcal{M}_6 = \begin{pmatrix} 1 & 1 & 2 & 2 \\ 0 & 1 & 1 & 2 \\ 0 & 2 & 3 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

The inverses of the aforesaid matrices, with respect to the operation $*$ *of the group* $\pi(\mathcal{L})$ *are*

$$\mathcal{M}_1^{-1} = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 0 & 0 & 3 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \mathcal{M}_2^{-1} = \begin{pmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 \end{pmatrix},$$

$$\mathcal{M}_3^{-1} = \begin{pmatrix} 1 & 3 & 2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 3 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad \mathcal{M}_4^{-1} = \begin{pmatrix} 1 & 3 & 1 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix},$$

$$\mathcal{M}_5^{-1} = \begin{pmatrix} 1 & 3 & 3 & 0 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \mathcal{M}_6^{-1} = \begin{pmatrix} 1 & 3 & 3 & 2 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Note that these matrices in general do not work as inverse elements under the usual product of matrices. For example,

$$\mathcal{M}_2\mathcal{M}_2^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 2 & 3 \end{pmatrix},$$

whereas $\mathcal{M}_2 * \mathcal{M}_2^{-1} = \mathrm{Id}_4$. *By Theorem 44,* $\Phi(\mathcal{P}_6) \subseteq \mathrm{PAut}(H_{1,3})$ *is a* 6-PD*-set of size* 7 *for the* $\mathbb{Z}_4$*-linear Hadamard code* $H_{1,3}$ *of length* 64 *with information set* $\Phi(\mathcal{I}_{1,3}) = \{1, 2, 3, 4, 9, 10, 33\}$ *given in Example 38. Indeed, note that matrices* Id_4^*,

$$(\mathcal{M}_1^{-1})^* = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 0 & 2 \\ 1 & 3 & 0 & 2 \\ 1 & 2 & 3 & 0 \end{pmatrix}, \quad (\mathcal{M}_2^{-1})^* = \begin{pmatrix} 1 & 0 & 3 & 0 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 2 \\ 1 & 2 & 3 & 2 \end{pmatrix},$$

$$(\mathcal{M}_3^{-1})^* = \begin{pmatrix} 1 & 3 & 2 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 2 \end{pmatrix}, \quad (\mathcal{M}_4^{-1})^* = \begin{pmatrix} 1 & 3 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 2 & 0 \\ 1 & 1 & 3 & 2 \end{pmatrix},$$

$$(\mathcal{M}_5^{-1})^* = \begin{pmatrix} 1 & 3 & 3 & 0 \\ 1 & 0 & 1 & 2 \\ 1 & 3 & 2 & 2 \\ 1 & 3 & 1 & 2 \end{pmatrix}, \quad (\mathcal{M}_6^{-1})^* = \begin{pmatrix} 1 & 3 & 3 & 2 \\ 1 & 2 & 2 & 2 \\ 1 & 3 & 0 & 0 \\ 1 & 1 & 3 & 0 \end{pmatrix}.$$

have no rows in common.

Since $\gamma = 1$ *and* $\delta = 3$, *we have that* $m = \gamma + 2\delta - 1 = 6$, *which means that* $H_{1,3}$ *is of length* 2^6. *Note that the bound* $f_6 = 8$ *is not attained. Since* $f_{1,3} = 7$, *a* 8-PD*-set of size* 9 *for* $H_{1,3}$ *cannot be obtained by using Theorem 44.*

Finally, mention that matrices in the set $\mathcal{P}_6$ *have been found by using the new* MAGMA *function* `PDSetHadamardCodeZ4(δ, m)` *with* $\delta = 3$, $m = 6$ *and the nondeterministic method (see Section 6.4.4). Due to this, a different execution of the function may provide a different set of matrices.*

We know that the $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ of length 2^m with $\delta = 1$ or $\delta = 2$ are equivalent to the binary linear Hadamard codes H_m of length 2^m [Kro01]. However, the results given in Section 3.2 for these codes will always be better than the ones obtained by using Theorem 44, since $f_{\gamma,\delta} \leq f_m$, where $m = \gamma + 2\delta - 1$.

Example 48. *In Example 29, a* 2-PD-*set of size* 3 *for* H_4 *is given. The code* H_4 *is equivalent to both* $\mathbb{Z}_4$*-linear Hadamard codes* $H_{1,2}$ *and* $H_{3,1}$*. However, a* 2-PD-*set of size* 3 *is not achievable by using Theorem 44, since* $f_{1,2} = f_{3,1} = 1$*.*

Example 49. *A* 4-PD-*set of size* 5 *for* H_5 *can be constructed by Theorem 28, since* $f_5 = 4$*. However, considering* H_5 *as the Gray map image of* $\mathcal{H}_{2,2}$ *or* $\mathcal{H}_{4,1}$*, no more than a* 3-PD-*set of size* 4 *can be found by using Theorem 44, since* $f_{4,1} = 2$ *and* $f_{2,2} = 3$*.*

4.2 Explicit construction of s-PD-sets of size $s+1$

In this section, by using Theorem 44, we give an explicit construction of s-PD-sets of minimum size $s+1$ for $H_{0,\delta}$, for all $\delta \geq 3$ and $2 \leq s \leq f_{0,\delta}$. We follow a similar technique to the one described for simplex codes in [FKM12] and for binary linear Hadamard codes in Section 3.2.

Let $\mathcal{R} = \mathrm{GR}(4^{\delta-1})$ be the Galois extension of dimension $\delta - 1$ over $\mathbb{Z}_4$. It turns out that as in the case of finite fields, Galois rings can be constructed as quotients of the associated polynomial over the base ring. For more information about some basic facts related to Galois rings, refer to [Mac60, HP03, Wan97, Wan03].

Let $\mathbb{Z}_4[x]$ and $\mathbb{Z}_2[x]$ be the polynomial ring over $\mathbb{Z}_4$ and $\mathbb{Z}_2$, respectively. Let $\mu : \mathbb{Z}_4[x] \to \mathbb{Z}_2[x]$ be the map that performs a modulo 2 reduction of the coefficients of $h(x) \in \mathbb{Z}_4[x]$. Recall that a monic polynomial $h(x) \in \mathbb{Z}_4[x]$ is said to be a primitive basic irreducible polynomial if $\mu(h(x))$ is primitive over $\mathbb{Z}_2[x]$. In [BHK92, Table I] and [Wan97, Table 6.1], all primitive basic irreducible polynomials for degrees 3 to 10 are provided.

It is known that $\mathcal{R}$ is isomorphic to $\mathbb{Z}_4[x]/(h(x))$, where $h(x)$ is a monic basic irreducible polynomial of degree $\delta - 1$. Let $f(x) \in \mathbb{Z}_2[x]$ be a primitive polynomial of degree $\delta - 1$. Let $\ell = 2^{\delta-1} - 1$. There is a unique primitive basic irreducible polynomial $h(x)$ dividing $x^\ell - 1$ in $\mathbb{Z}_4[x]$ and such that $\mu(h(x)) = f(x)$, where μ is the map that performs modulo 2 to all coefficients of $h(x)$. Let α be a root of $h(x)$.

There are two canonical forms to represent the $4^{\delta-1}$ elements of $\mathcal{R}$, the so-called additive and multiplicative representations. In the first one, each

element $r \in \mathcal{R}$ can be uniquely expressed as

$$r = \sum_{i=0}^{\delta-2} b_i \alpha^i,$$

where $b_i \in \mathbb{Z}_4$. Note that in this representation, we may identify $r \in \mathcal{R}$ with $(b_0, \ldots, b_{\delta-2}) \in \mathbb{Z}_4^{\delta-1}$. Next, we introduce the multiplicative representation. The subset $\mathcal{T} = \{0, 1, \alpha, \ldots, \alpha^{\ell-1}\} \subseteq \mathcal{R}$ is called the Teichmuller set and it contains the elements $r \in \mathcal{R}$ satisfying that $r^\ell = 1$. It is well known that any $r \in \mathcal{R}$ can be written uniquely as $r = a + 2b$, where $a, b \in \mathcal{T}$.

We take $\mathcal{R}$ as the following ordered set:

$$\begin{aligned}\mathcal{R} &= \{r_1, \ldots, r_{4^{\delta-1}}\} \\ &= \{0 + 2 \cdot 0, \ldots, \alpha^{\ell-1} + 2 \cdot 0, \ldots, 0 + 2 \cdot \alpha^{\ell-1}, \ldots, \alpha^{\ell-1} + 2 \cdot \alpha^{\ell-1}\}. \quad (4.2)\end{aligned}$$

Since $|\mathcal{R}|/\delta = f_{0,\delta} + 1$, we can form $f_{0,\delta} + 1$ disjoints sets of $\mathcal{R}$ of size δ. For $i \in \{0, \ldots, f_{0,\delta}\}$, we consider the $\delta \times \delta$ quaternary matrices

$$\mathcal{N}_i^* = \begin{pmatrix} 1 & r_{\delta i+1} \\ \vdots & \vdots \\ 1 & r_{\delta(i+1)} \end{pmatrix}.$$

Theorem 50. *Let $\mathcal{P}_s = \{\mathcal{M}_i : 0 \le i \le s\}$, where $\mathcal{M}_i = \mathcal{N}_i^{-1}$. Then, $\Phi(\mathcal{P}_s)$ is an s-PD-set of size $s+1$ for the $\mathbb{Z}_4$-linear Hadamard code $H_{0,\delta}$ of length $2^{2\delta-1} = 2^m$ with information set $\Phi(\mathcal{I}_{0,\delta})$, for all $\delta \ge 3$ and $2 \le s \le f_{0,\delta} = f_m$.*

Proof. We need to prove that $r_{\delta i+2} - r_{\delta i+1}, \ldots, r_{\delta(i+1)} - r_{\delta i+1}$ are linearly independent over $\mathbb{Z}_4$, for all $i \in \{0, \ldots, f_{0,\delta}\}$, in order to guarantee that $\mathcal{N}_i \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$. Note that these vectors are not zero divisors [HKC+94]. Since $\alpha^\ell = 1$, $\{r_{\delta i+2} - r_{\delta i+1}, \ldots, r_{\delta(i+1)} - r_{\delta i+1}\}$ is one of the following three sets:

$$\begin{aligned}
L_1 = {} & \{1, \ldots, \alpha^{\delta-2}\}, \\
L_2 = {} & \{\alpha^{k+1} - \alpha^k, \ldots, \alpha^{k+\delta-1} - \alpha^k\}, \text{for some } k \in \{0, \ldots, \ell-1\}, \\
L_3 = {} & \{\alpha^{k+1} - \alpha^k, \ldots, \alpha^{\ell-1} - \alpha^k, -\alpha^k + 2(b_j - b_i), \alpha^\ell - \alpha^k + 2(b_j - b_i), \ldots, \\
& \alpha^{k+\delta-2} - \alpha^k + 2(b_j - b_i)\}, \text{for some } b_i, b_j \in \mathcal{T} \text{ and } k \in \{0, \ldots, \ell-1\}.
\end{aligned}$$

The elements in L_1 are clearly linearly independent over $\mathbb{Z}_4$. Now, we prove that the same property is satisfied in L_2. Assume on the contrary that there are some $\lambda_i \neq 0$, $i \in \{1, \ldots, \delta-1\}$, such that $\sum \lambda_i(\alpha^{k+i} - \alpha^k) = 0$. If $\lambda_i \in \{1, 3\}$ for at least one $i \in \{1, \ldots, \delta-1\}$, we get a contradiction.

Indeed, if we take modulo 2 in the previous linear combination, we obtain that $\sum \bar{\lambda}_i(\bar{\alpha}^{k+i} - \bar{\alpha}^k) = 0$, where $\bar{\lambda}_i \in \mathbb{Z}_2$ and at least one $\bar{\lambda}_i \neq 0$. This is a contradiction by Lemma 27. On the other hand, if $\lambda_i \in \{0, 2\}$ for all $i \in \{1, \ldots, \delta - 1\}$ and there is at least one $\lambda_i = 2$, then $\sum 2\lambda_i'(\alpha^{k+i} - \alpha^k) = 2[\sum \lambda_i'(\alpha^{k+i} - \alpha^k)] = 0$, where $\lambda_i' \in \{0, 1\}$ and at least one $\lambda_i' = 1$. Hence, $\sum \lambda_i'(\alpha^{k+i} - \alpha^k) = 2\lambda$ for some $\lambda \in \mathcal{R}$, that is, it is a zero divisor. By taking modulo 2, we obtain a contradiction again by Lemma 27.

We show that the elements in $L_3 = \{v_1, \ldots, v_{\delta-1}\}$ are also linearly independent over $\mathbb{Z}_4$ by using a slight modification of the previous argument. Suppose that there is at least one $\lambda_i \neq 0$, $i \in \{1, \ldots, \delta - 1\}$, such that $\sum \lambda_i v_i = 0$. By taking modulo 2, we obtain that $\bar{\lambda}_{\delta-1}\bar{\alpha}^k + \sum \bar{\lambda}_i(\bar{\alpha}^{k+i} - \bar{\alpha}^k) = \bar{\alpha}^k[\bar{\lambda}_{\delta-1} + \sum \bar{\lambda}_i(\bar{\alpha}^i - 1)] = 0$. Since $\bar{\alpha}^k$ is a unit, it follows that $\bar{\lambda}_{\delta-1} + \sum \bar{\lambda}_i(\bar{\alpha}^i - 1) = 0$, which gives a contradiction if $\lambda_i \in \{1, 3\}$ for at least one index, since $1, \bar{\alpha} - 1, \ldots, \bar{\alpha}^{\delta-2} - 1$ are linearly independent over $\mathbb{Z}_2$. If $\lambda_i \in \{0, 2\}$ for all $i \in \{1, \ldots, \delta - 1\}$, we get a contradiction by applying a similar argument to the one used above.

Finally, by construction, the matrices $\mathcal{N}_0^*, \ldots, \mathcal{N}_s^*$ have no rows in common and the result follows by Theorem 44. □

Note that the bound f_m is always attained for $H_{0,\delta}$ despite the elements of the f_m-PD-set belong to the subgroup $\Phi(\mathrm{PAut}(\mathcal{H}_{0,\delta})) \leq \mathrm{PAut}(H_{0,\delta})$, since $f_m = f_{0,\delta}$.

Example 51. *Let $\mathcal{H}_{0,3}$ be the quaternary linear Hadamard code of length* 16 *and type* $2^0 4^3$. *Let $\mathcal{R} = \mathbb{Z}_4[x]/(h(x))$, where $h(x) = x^2 + x + 1$. Note that $h(x)$ is a primitive basic irreducible polynomial dividing $x^3 - 1$ in $\mathbb{Z}_4[x]$. Let α be a root of $h(x)$. Then, $\mathcal{T} = \{0, 1, \alpha, \alpha^2\}$ and $\mathcal{R} = \{r_1, \ldots, r_{16}\} = \{0, 1, \alpha, 3 + 3\alpha, 2, 3, 2 + \alpha, 1 + 3\alpha, 2\alpha, 1 + 2\alpha, 3\alpha, 3 + \alpha, 2 + 2\alpha, 3 + 2\alpha, 2 + 3\alpha, 1 + \alpha\}$. Let $\mathcal{P}_4 = \{\mathcal{N}_0^{-1}, \mathcal{N}_1^{-1}, \mathcal{N}_2^{-1}, \mathcal{N}_3^{-1}, \mathcal{N}_4^{-1}\}$, where $\mathcal{N}_0 = \mathrm{Id}_3$, and $\mathcal{N}_1$, $\mathcal{N}_2$, $\mathcal{N}_3$ and $\mathcal{N}_4$ are the following matrices:*

$$\begin{pmatrix} 1 & 3 & 3 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 & 1 \\ 0 & 3 & 2 \\ 0 & 2 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 2 \\ 0 & 3 & 1 \\ 0 & 2 & 3 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 1 & 2 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Note that $\mathcal{P}_4 \subseteq \mathrm{PAut}(\mathcal{H}_{0,3})$. The set $\mathcal{P}_4$ regarded as a subset of $\mathrm{Sym}(16)$ *is* $\{\mathrm{id}, \tau_1, \tau_2, \tau_3, \tau_4\}$, *where*

$$\begin{aligned} \tau_1 &= (1, 12, 13, 8, 9, 4, 5, 16)(2, 15, 14, 11, 10, 7, 6, 3), \\ \tau_2 &= (1, 13, 11, 7)(2, 8, 12, 14)(3, 15, 9, 5)(4, 6, 10, 16), \\ \tau_3 &= (1, 8, 5, 10)(2, 9, 16, 13)(3, 14, 7, 4)(6, 15, 12, 11), \\ \tau_4 &= (1, 11)(2, 12)(3, 9)(4, 10)(5, 15)(6, 16)(7, 13)(8, 14). \end{aligned}$$

By Theorem 50, $S = \Phi(\mathcal{P}_4) = \{\mathrm{id}, \sigma_1, \sigma_2, \sigma_3, \sigma_4\} \subseteq \mathrm{Sym}(32)$, *where*

$$\begin{aligned}
\sigma_1 = {} & (1,23,25,15,17,7,9,31)(2,24,26,16,18,8,10,32)\\
& (3,29,27,21,19,13,11,5)(4,30,28,22,20,14,12,6),\\
\sigma_2 = {} & (1,25,21,13)(2,26,22,14)(3,15,23,27)(4,16,24,28)\\
& (5,29,17,9)(6,30,18,10)(7,11,19,31)(8,12,20,32),\\
\sigma_3 = {} & (1,15,9,19)(2,16,10,20)(3,17,31,25)(4,18,32,26)\\
& (5,27,13,7)(6,28,14,8)(11,29,23,21)(12,30,24,22),\\
\sigma_4 = {} & (1,21)(2,22)(3,23)(4,24)(5,17)(6,18)(7,19)(8,20)\\
& (9,29)(10,30)(11,31)(12,32)(13,25)(14,26)(15,27)(16,28),
\end{aligned}$$

is a 4-PD-*set of size* 5 *for* $H_{0,3}$. *Note that* $\mathcal{N}_0^* = \mathrm{Id}_3^*$, *and the following matrices* $\mathcal{N}_1^*, \ldots, \mathcal{N}_4^*$

$$\begin{pmatrix} 1 & 3 & 3 \\ 1 & 2 & 0 \\ 1 & 3 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 & 1 \\ 1 & 1 & 3 \\ 1 & 0 & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 2 \\ 1 & 0 & 3 \\ 1 & 3 & 1 \end{pmatrix}, \quad \textit{and} \quad \begin{pmatrix} 1 & 2 & 2 \\ 1 & 3 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

have no rows in common.

Finally, mention that matrices in the set $\mathcal{P}_4$ *can also be obtained by using the new* MAGMA *function* `PDSetHadamardCodeZ4(δ, m)` *with* $\delta = 3$, $m = 5$ *and the deterministic method (see Section 6.4.4). Unlike Example 47, the set* $\mathcal{P}_4$ *provided in this example is the constant outcome of this function.*

4.3 Recursive constructions of s-PD-sets

In this section, given an s-PD-set of size l for the $\mathbb{Z}_4$-linear Hadamard code $H_{\gamma,\delta}$ of length 2^m and type $2^\gamma 4^\delta$, where $m = \gamma + 2\delta - 1$ and $l \geq s + 1$, we show how to construct recursively an s-PD-set of the same size for $H_{\gamma+i,\delta+j}$ of length 2^{m+i+2j} and type $2^{\gamma+i}4^{\delta+j}$ for all $i, j \geq 0$.

Next, a first recursive construction considering the elements of $\mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ as matrices in $\mathrm{GL}(\gamma+\delta, \mathbb{Z}_4)$ is provided. This construction can be seen as a natural generalization of the technique used for binary linear Hadamard codes in Section 3.3.

Given a matrix

$$\mathcal{M} = \begin{pmatrix} 1 & \eta & 2\theta \\ \mathbf{0} & A & 2X \\ \mathbf{0} & \zeta(Y) & \zeta(B) \end{pmatrix} \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$$

and an integer $\kappa \geq 1$, we define

$$\mathcal{M}(\kappa) = \begin{pmatrix} 1 & \eta & \mathbf{0} & 2\theta \\ \mathbf{0} & A & \mathbf{0} & 2X \\ \mathbf{0} & \mathbf{0} & \mathrm{Id}_\kappa & \mathbf{0} \\ \mathbf{0} & \zeta(Y) & \mathbf{0} & \zeta(B) \end{pmatrix}.$$

Proposition 52. *Let $\mathcal{P}_s = \{\mathcal{M}_0, \ldots, \mathcal{M}_s\} \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ such that $\Phi(\mathcal{P}_s)$ is an s-PD-set of size $s+1$ for $H_{\gamma,\delta}$ with information set $\Phi(\mathcal{I}_{\gamma,\delta})$. Then, $\mathcal{Q}_s = \{(\mathcal{M}_0^{-1}(\kappa))^{-1}, \ldots, \mathcal{M}_s^{-1}(\kappa))^{-1}\} \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma+i,\delta+j})$ and $\Phi(\mathcal{Q}_s)$ is an s-PD-set of size $s+1$ for $H_{\gamma+i,\delta+j}$ with information set $\Phi(\mathcal{I}_{\gamma+i,\delta+j})$, for any $i, j \geq 0$ such that $i + j = \kappa \geq 1$.*

Proof. Note that if $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$, then $\mathcal{M}(\kappa) \in \mathrm{GL}(\gamma+\delta+\kappa, \mathbb{Z}_4)$. Taking this into account, together with the fact that Id_κ can split as

$$\mathrm{Id}_\kappa = \begin{pmatrix} \mathrm{Id}_j & \mathbf{0} \\ \mathbf{0} & \mathrm{Id}_i \end{pmatrix},$$

where $i + j = \kappa \geq 1$, it is clear that $\mathcal{M}^{-1}(\kappa) \in \mathrm{PAut}(\mathcal{H}_{\gamma+i,\delta+j})$ and so its inverse. Thus, $\mathcal{Q}_s \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma+i,\delta+j})$. Finally, repeated rows in the matrices $(\mathcal{M}_0^{-1}(\kappa))^*, \ldots, (\mathcal{M}_s^{-1}(\kappa))^*$ cannot occur, since this fact would imply repeated rows in the matrices $(\mathcal{M}_0^{-1})^*, \ldots, (\mathcal{M}_s^{-1})^*$ by construction. The result follows from Theorem 44. □

Example 53. *Let $\mathcal{P}_4 = \{\mathcal{M}_0, \ldots, \mathcal{M}_4\} \subseteq \mathrm{PAut}(\mathcal{H}_{0,3})$ be the set, given in Example 51, such that $\Phi(\mathcal{P}_4)$ is a 4-PD-set of size 5 for $H_{0,3}$. By Proposition 52, $\mathcal{Q}_4 = \{\mathcal{M}_i^{-1}(1))^{-1} \ : \ 0 \leq i \leq 4\}$ is contained in both $\mathrm{PAut}(\mathcal{H}_{1,3})$ and $\mathrm{PAut}(\mathcal{H}_{0,4})$. Moreover, $\Phi(\mathcal{Q}_4)$ is a 4-PD-set of size 5 for $H_{1,3}$ and $H_{0,4}$. Nevertheless, note that the construction of $(\mathcal{M}_i^{-1}(1))^*$ depends on the group where $\mathcal{M}_i^{-1}(1)$ is considered. For example, the matrix*

$$\mathcal{M}_1^{-1}(1) = \begin{pmatrix} 1 & 3 & 3 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

considered as an element in $\mathrm{PAut}(\mathcal{H}_{1,3})$ and $\mathrm{PAut}(\mathcal{H}_{0,4})$ leads, respectively, to

$$(\mathcal{M}_1^{-1}(1))^* = \begin{pmatrix} 1 & 3 & 3 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 3 & 3 & 2 \end{pmatrix} \text{ and } \quad (\mathcal{M}_1^{-1}(1))^* = \begin{pmatrix} 1 & 3 & 3 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 3 & 3 & 1 \end{pmatrix}.$$

As for binary linear Hadamard codes, a second recursive construction considering the elements of $\mathrm{PAut}(H_{\gamma,\delta})$ as permutations of coordinate positions, that is as elements of $\mathrm{Sym}(2^m)$, can also be provided. This construction can also be seen as a generalization of the one described in Section 3.3 for binary linear Hadamard codes.

We define $(\sigma_1|\sigma_2|\sigma_3|\sigma_4) \in \mathrm{Sym}(n_1+n_2+n_3+n_4)$, for any four permutations $\sigma_i \in \mathrm{Sym}(n_i)$, $i \in \{1,\ldots,4\}$, in the same way as we defined $(\sigma_1|\sigma_2) \in \mathrm{Sym}(n_1+n_2)$ in Section 3.3.

Proposition 54. *Let S be an s-PD-set of size l for $H_{\gamma,\delta}$ of length n and type $2^\gamma 4^\delta$ with information set I. Then, $(S|S) = \{(\sigma|\sigma) : \sigma \in S\}$ is an s-PD-set of size l for $H_{\gamma+1,\delta}$ of length $2n$ and type $2^{\gamma+1}4^\delta$ constructed from (2.11) and the Gray map, with any information set $I' = I \cup \{i+n\}$, $i \in I$.*

Proof. Since $H_{\gamma+1,\delta} = \{(x,x),(x,\bar{x}) : x \in H_{\gamma,\delta}\}$, where $\bar{x}$ is the complementary vector of x, the result follows using the same argument as in the proof of Proposition 33. By the proof of Proposition 37, we can add any of the coordinate positions of $\{i+n : i \in I\}$ to I in order to form a suitable information set I' for $H_{\gamma+1,\delta}$. □

Let $2S = 2^1 S$ denote the set $(S|S)$ and, recursively, $2^i S = 2(2^{i-1}S)$.

Corollary 55. *Let $\mathcal{P}_s = \{\mathcal{M}_i : 0 \le i \le s\}$, where $\mathcal{M}_i = \mathcal{N}_i^{-1}$. Then, $2^\gamma \Phi(\mathcal{P}_s)$ is an s-PD-set of size $s+1$ for the $\mathbb{Z}_4$-linear Hadamard code $H_{\gamma,\delta}$, for all $\gamma \ge 0$, $\delta \ge 3$ and $2 \le s \le f_{0,\delta}$.*

Proof. By Theorem 50 and Proposition 54, we can construct $f_{0,\delta}$-PD-sets of size $f_{0,\delta}+1$ for $H_{\gamma,\delta}$, for all $\gamma \ge 0$ and $\delta \ge 3$. □

Proposition 54 cannot be generalized directly for $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta+1}$ constructed from (2.12) and the Gray map. Note that if S is an s-PD-set for $H_{\gamma,\delta}$, then $(S|S|S|S) = \{(\sigma|\sigma|\sigma|\sigma) : \sigma \in S\}$ is not always an s-PD-set for $H_{\gamma,\delta+1}$, since in general $(\sigma|\sigma|\sigma|\sigma) \notin \mathrm{PAut}(H_{\gamma,\delta})$. For example, $\sigma = (1,5)(2,8,3,6,4,7) \in \mathrm{PAut}(H_{0,2}) \subseteq \mathrm{Sym}(8)$, but $\pi = (\sigma|\sigma|\sigma|\sigma) \notin \mathrm{PAut}(H_{0,3}) \subseteq \mathrm{Sym}(32)$, since $\pi(\Phi((0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3))) = \Phi((0,0,0,0,0,0,2,0,2,2,2,2,2,2,0,2,0)) \notin H_{0,3}$.

Proposition 56. *Let $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ such that $\Phi(\mathcal{S})$ is an s-PD-set of size l for $H_{\gamma,\delta}$ of length n and type $2^\gamma 4^\delta$ with information set I. Then, $\Phi((\mathcal{S}|\mathcal{S}|\mathcal{S}|\mathcal{S})) = \{\Phi((\tau|\tau|\tau|\tau)) : \tau \in \mathcal{S}\}$ is an s-PD-set of size l for $H_{\gamma,\delta+1}$ of length $4n$ and type $2^\gamma 4^{\delta+1}$ constructed from (2.12) and the Gray map, with any information set $I'' = I \cup \{i+n, j+n\}$, $i,j \in I$ and $i \ne j$.*

Proof. Since $\mathcal{H}_{\gamma,\delta+1}$ is constructed from (2.12), $\mathcal{H}_{\gamma,\delta+1} = \{(u,u,u,u),(u,u+\mathbf{1},u+\mathbf{2},u+\mathbf{3}),(u,u+\mathbf{2},u,u+\mathbf{2}),(u,u+\mathbf{3},u,u+\mathbf{1}) : u \in \mathcal{H}_{\gamma,\delta}\}$. It is easy to see that if $\tau \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$, then $(\tau|\tau|\tau|\tau) \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta+1})$.

Let $\sigma = \Phi(\tau)$. Finally, we need to prove that for every $e \in \mathbb{Z}_2^{4n}$ with $\mathrm{wt}(e) \leq s$, there is $(\sigma|\sigma|\sigma|\sigma) \in \Phi((\mathcal{S}|\mathcal{S}|\mathcal{S}|\mathcal{S}))$ such that $(\sigma|\sigma|\sigma|\sigma)(e)_{I''} = \mathbf{0}$, where $I'' \subseteq \{1,\ldots,4n\}$ is an information set for $H_{\gamma,\delta+1}$ with $\gamma + 2(\delta+1)$ coordinate positions. Using a similar argument to that given in the proofs of Propositions 33 and 54, the result follows. Moreover, by the proof of Proposition 37, any $I'' = I \cup \{i+n, j+n\}$ with $i,j \in I$ and $i \neq j$ is a suitable information set for $H_{\gamma,\delta+1}$. □

Corollary 57. *Let $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ such that $\Phi(\mathcal{S})$ is an s-PD-set of size l for $H_{\gamma,\delta}$ of length 2^m and type $2^\gamma 4^\delta$ with information set I. Then, $\Phi(2^{i+2j}\mathcal{S})$ is an s-PD-set of size l for $H_{\gamma+i,\delta+j}$ of length 2^{m+i+2j} and type $2^{\gamma+i}4^{\delta+j}$ with information set obtained by applying recursively Proposition 37, for all $i,j \geq 0$.*

Proof. The result comes trivially by applying Propositions 37, 54 and 56. □

4.4 Computational results

Using the functions implemented in the new MAGMA package (see Section 6.4), it is possible to improve easily the result given by Corollary 55 for $H_{\gamma,\delta}$ with $\gamma > 0$, that is, to obtain s-PD-sets of size $s+1$ for $f_{0,\delta} < s \leq f_{\gamma,\delta}$ by using a nondeterministic method. Table 4.1 summarizes these computational results for the codes $H_{\gamma,\delta}$ with $3 \leq \delta \leq 6$ and $1 \leq \gamma \leq 5$. Specifically, for each one of these codes, the values of $f_{0,\delta}, f_{\gamma,\delta}, f_m$ and t_m are shown, where $m = \gamma + 2\delta - 1$, together with the maximum s for which an s-PD-set of size $s+1$ has been found. Even when the nondeterministic method fails to quickly find a $f_{\gamma,\delta}$-PD-set of minimum size $f_{\gamma,\delta}+1$, the bound $f_{\gamma,\delta}$ may be attained as shown in the following examples.

Example 58. *Let the ordered set $\mathcal{R}$ and the matrices $\mathcal{N}_0^*,\ldots,\mathcal{N}_3^*$ be as in Example 51. Define $\bar{r} = (1,r) \in \{1\} \times \mathbb{Z}_4^2$ for all $r \in \mathcal{R}$. Let $\mathcal{P}_7 = \{\mathcal{A}_i^{-1} : 0 \leq i \leq 7\}$, where $\mathcal{A}_i^*$ are the following matrices:*

$$\begin{pmatrix} \mathcal{N}_0^* & \mathbf{0} \\ \bar{r}_{13} & 2 \end{pmatrix}, \quad \begin{pmatrix} \mathcal{N}_1^* & \mathbf{0} \\ \bar{r}_{16} & 2 \end{pmatrix}, \quad \begin{pmatrix} \mathcal{N}_2^* & \mathbf{0} \\ \bar{r}_{15} & 2 \end{pmatrix}, \quad \begin{pmatrix} \mathcal{N}_3^* & \mathbf{0} \\ \bar{r}_{14} & 2 \end{pmatrix},$$

$$\begin{pmatrix} \mathcal{N}_0^* & \mathbf{2} \\ \bar{r}_{13} & 0 \end{pmatrix}, \quad \begin{pmatrix} \mathcal{N}_1^* & \mathbf{2} \\ \bar{r}_{16} & 0 \end{pmatrix}, \quad \begin{pmatrix} \mathcal{N}_2^* & \mathbf{2} \\ \bar{r}_{15} & 0 \end{pmatrix}, \quad \begin{pmatrix} \mathcal{N}_3^* & \mathbf{2} \\ \bar{r}_{14} & 0 \end{pmatrix}.$$

By Theorem 44, one can easily check that $\Phi(\mathcal{P}_7) \subseteq \mathrm{PAut}(H_{1,3})$ *is a* 7-PD*-set of size* 8 *for* $H_{1,3}$ *of length* 64 *with information set* $\Phi(\mathcal{I}_{1,3}) = \{1, 2, 3, 4, 9, 10, 33\}$. *Since* $f_{1,3} = 7$, *no better* s-PD*-sets of size* $s+1$ *can be provided for* $H_{1,3}$ *by using Theorem 44. However, an* 8-PD*-set of size* 9 *could be theoretically found in* $\mathrm{PAut}(H_{1,3})$ *since* $f_6 = 8$.

Example 59. *Let* $\mathcal{H}_{1,4}$ *be the quaternary linear Hadamard code of length 128 and type* $2^1 4^4$. *Let*

$$\begin{aligned} \mathcal{R} &= \mathbb{Z}_4[x]/(h(x)) \\ &= \{r_1, \ldots, r_{64}\}, \end{aligned}$$

as described in (4.2), where $h(x) = x^3 + 2x^2 + x + 3$. *Note that* $h(x)$ *is a primitive basic irreducible polynomial dividing* $x^7 - 1$ *in* $\mathbb{Z}_4[x]$. *Let* $\mathcal{P}_{24} = \{\mathcal{M}_i : 0 \le i \le 24\}$, *where* $(\mathcal{M}_i^{-1})^*$ *are the following matrices:*

$$\begin{pmatrix} 1 & r_1 & 0 \\ 1 & r_2 & 0 \\ 1 & r_3 & 0 \\ 1 & r_4 & 0 \\ 1 & r_{57} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_5 & 0 \\ 1 & r_6 & 0 \\ 1 & r_7 & 0 \\ 1 & r_8 & 0 \\ 1 & r_{61} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{10} & 0 \\ 1 & r_9 & 0 \\ 1 & r_{11} & 0 \\ 1 & r_{12} & 0 \\ 1 & r_{58} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{14} & 0 \\ 1 & r_{13} & 0 \\ 1 & r_{15} & 0 \\ 1 & r_{16} & 0 \\ 1 & r_{62} & 2 \end{pmatrix},$$

$$\begin{pmatrix} 1 & r_{19} & 0 \\ 1 & r_{18} & 0 \\ 1 & r_{17} & 0 \\ 1 & r_{20} & 0 \\ 1 & r_{59} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{23} & 0 \\ 1 & r_{22} & 0 \\ 1 & r_{21} & 0 \\ 1 & r_{24} & 0 \\ 1 & r_{63} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{28} & 0 \\ 1 & r_{26} & 0 \\ 1 & r_{27} & 0 \\ 1 & r_{25} & 0 \\ 1 & r_{60} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{32} & 0 \\ 1 & r_{30} & 0 \\ 1 & r_{31} & 0 \\ 1 & r_{29} & 0 \\ 1 & r_{64} & 2 \end{pmatrix},$$

$$\begin{pmatrix} 1 & r_{33} & 0 \\ 1 & r_{34} & 0 \\ 1 & r_{35} & 0 \\ 1 & r_{36} & 0 \\ 1 & r_{49} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{37} & 0 \\ 1 & r_{38} & 0 \\ 1 & r_{39} & 0 \\ 1 & r_{40} & 0 \\ 1 & r_{53} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{42} & 0 \\ 1 & r_{41} & 0 \\ 1 & r_{43} & 0 \\ 1 & r_{44} & 0 \\ 1 & r_{50} & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{46} & 0 \\ 1 & r_{45} & 0 \\ 1 & r_{47} & 0 \\ 1 & r_{48} & 0 \\ 1 & r_{54} & 2 \end{pmatrix},$$

$$\begin{pmatrix} 1 & r_1 & 2 \\ 1 & r_2 & 2 \\ 1 & r_3 & 2 \\ 1 & r_4 & 2 \\ 1 & r_{57} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_5 & 2 \\ 1 & r_6 & 2 \\ 1 & r_7 & 2 \\ 1 & r_8 & 2 \\ 1 & r_{61} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{10} & 2 \\ 1 & r_9 & 2 \\ 1 & r_{11} & 2 \\ 1 & r_{12} & 2 \\ 1 & r_{58} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{14} & 2 \\ 1 & r_{13} & 2 \\ 1 & r_{15} & 2 \\ 1 & r_{16} & 2 \\ 1 & r_{62} & 0 \end{pmatrix},$$

$$\begin{pmatrix} 1 & r_{19} & 2 \\ 1 & r_{18} & 2 \\ 1 & r_{17} & 2 \\ 1 & r_{20} & 2 \\ 1 & r_{59} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{23} & 2 \\ 1 & r_{22} & 2 \\ 1 & r_{21} & 2 \\ 1 & r_{24} & 2 \\ 1 & r_{63} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{28} & 2 \\ 1 & r_{26} & 2 \\ 1 & r_{27} & 2 \\ 1 & r_{25} & 2 \\ 1 & r_{60} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{32} & 2 \\ 1 & r_{30} & 2 \\ 1 & r_{31} & 2 \\ 1 & r_{29} & 2 \\ 1 & r_{64} & 0 \end{pmatrix},$$

$$\begin{pmatrix} 1 & r_{33} & 2 \\ 1 & r_{34} & 2 \\ 1 & r_{35} & 2 \\ 1 & r_{36} & 2 \\ 1 & r_{49} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{37} & 2 \\ 1 & r_{38} & 2 \\ 1 & r_{39} & 2 \\ 1 & r_{40} & 2 \\ 1 & r_{53} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{42} & 2 \\ 1 & r_{41} & 2 \\ 1 & r_{43} & 2 \\ 1 & r_{44} & 2 \\ 1 & r_{50} & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & r_{46} & 2 \\ 1 & r_{45} & 2 \\ 1 & r_{47} & 2 \\ 1 & r_{48} & 2 \\ 1 & r_{54} & 0 \end{pmatrix},$$

$$\begin{pmatrix} 1 & r_{51} & 0 \\ 1 & r_{52} & 0 \\ 1 & r_{55} & 0 \\ 1 & r_{56} & 0 \\ 1 & r_{51} & 2 \end{pmatrix}.$$

Note that the aforementioned matrices have no rows in common; the corresponding matrices $\mathcal{M}_i^{-1}$ are invertible in the group $\pi(\mathcal{L})$, $i \in \{0, \ldots, 24\}$; and $\mathcal{M}_i \in \mathrm{PAut}(\mathcal{H}_{1,4})$. By Theorem 44, the set $\Phi(\mathcal{P}_{24})$ is a 24-PD-set of size 25 for the $\mathbb{Z}_4$-linear Hadamard code $H_{1,4} = \Phi(\mathcal{H}_{1,4})$ of length 256 with information set $\Phi(\mathcal{I}_{1,4}) = \{1, 2, 3, 4, 9, 10, 33, 34, 65\}$. Since $f_{1,4} = 24$, no better s-PD-sets of size $s+1$ can be provided for $H_{1,4}$ by using Theorem 44. However, an 27-PD-set of size 28 could be theoretically found in $\mathrm{PAut}(H_{1,4})$ since $f_8 = 27$.

From Examples 58 and 59 we conjecture that $f_{\gamma,\delta}$-PD-sets of size $f_{\gamma,\delta}+1$ can be found by using only elements in $\Phi(\mathrm{PAut}(\mathcal{H}_{\gamma,\delta}))$.

δ	γ	$f_{0,\delta}$	s	$f_{\gamma,\delta}$	f_m	t_m
3	0	4	**4**	4	4	7
	1	4	**6**	7	8	15
	2	4	**10**	11	15	31
	3	4	**16**	20	27	63
	4	4	**26**	35	50	127
	5	4	**42**	63	92	255
4	0	15	**15**	15	15	31
	1	15	**23**	24	27	63
	2	15	**36**	41	50	127
	3	15	**56**	72	92	255
	4	15	**91**	127	169	511
	5	15	**150**	226	314	1023
5	0	50	**50**	50	50	127
	1	50	**72**	84	92	255
	2	50	**116**	145	169	511
	3	50	**187**	255	314	1023
	4	50	**312**	454	584	2047
	5	50	**518**	818	1091	4095
6	0	169	**169**	169	169	511
	1	169	**230**	291	314	1023
	2	169	**377**	511	584	2047
	3	169	**630**	909	1091	4095
	4	169	**1040**	1637	2047	8191
	5	169	**1784**	2977	3854	16383

Table 4.1: Maximum s for which s-PD-sets of size $s+1$ are found computationally for some codes $H_{\gamma,\delta}$

Chapter 5

PD-sets for $\mathbb{Z}_4$-linear codes

In this chapter, we focus on finding s-PD-sets of size $s+1$ for $\mathbb{Z}_4$-linear in general. In Section 5.1, we introduce a theorem that states sufficient conditions for a permutation $\sigma \in \mathrm{PAut}(C)$ to generate an s-PD-set $S = \{\sigma^i \ : \ 1 \leq i \leq s+1\}$ of size $s+1$ for a $\mathbb{Z}_4$-linear code C. In Section 5.2, by using this theorem, we obtain new s-PD-sets of size $s+1$ for the $\mathbb{Z}_4$-linear Hadamard code $H_{\gamma,\delta}$ for all $\delta \geq 4$ and $1 < s \leq 2^\delta - 3$. These new sets are generated by a single permutation, unlike the s-PD-sets for $\mathbb{Z}_4$-linear Hadamard codes given in Chapter 4. Finally, in Section 5.3, we obtain s-PD-sets of size $s+1$, for all $m \geq 5$ and $1 < s \leq \lambda - 1$, for the $\mathbb{Z}_4$-linear Kerdock code of length 2^m such that $2^{m-1}-1$ is not prime, where λ is the greatest divisor of $2^{m-1}-1$ satisfying $\lambda \leq 2^{m-1}/m$.

The results given in this chapter were presented at "IEEE International Symposium on Information Theory" in Barcelona, Spain, 2016 [BV16b]. They are planning to be published in [BV16c] together with some results from Section 2.7.

5.1 Second criterion to find s-PD-sets of size $s+1$

Let C be a $\mathbb{Z}_4$-linear code. Next, we introduce the main theorem of this section that provides sufficient conditions for a permutation $\sigma \in \mathrm{PAut}(C)$ to generate an s-PD-set $S = \{\sigma^i \ : \ 1 \leq i \leq s+1\}$ of size $s+1$.

In what follows, we denote the order of an element A in a finite group G by $\mathrm{ord}(A)$.

Theorem 60. *Let $\mathcal{C}$ be a quaternary linear code of length β and type $2^\gamma 4^\delta$ with quaternary information set $\mathcal{I}$ and let s be a positive integer. If $\tau \in \mathrm{PAut}(\mathcal{C})$ has at least $\gamma+\delta$ disjoint cycles of length $s+1$ such that there is exactly one quaternary information position per cycle of length $s+1$, then $S = \{\Phi(\tau^i)\}_{i=1}^{s+1}$ is an s-PD-set of size $s+1$ for the $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$ with information set $\Phi(\mathcal{I})$. Moreover, any ordering of the elements of S gives a nested r-PD-set for any $r \in \{1,\ldots,s\}$.*

Proof. The permutation $\tau \in \mathrm{PAut}(\mathcal{C})$ can be written as

$$\begin{aligned}\tau = \quad & (i_1, x_2, \ldots, x_{(s+1)})(i_2, x_{(s+1)+2}, \ldots, x_{2(s+1)}) \cdots \\ & (i_{\gamma+\delta}, x_{(\gamma+\delta-1)(s+1)+2}, \ldots, x_{(\gamma+\delta)(s+1)})\tau',\end{aligned} \tag{5.1}$$

where $\mathcal{I} = \{i_1, \ldots, i_{\gamma+\delta}\}$ is the quaternary information set for $\mathcal{C}$ and $\tau' \in \mathrm{Sym}(\beta)$. We consider the elements of $\mathcal{I}$ ordered in such a way that $|\mathcal{C}_{\{i_1,\ldots,i_\delta\}}| = 4^\delta$. Note that each cycle $(i_\epsilon, x_{(\epsilon-1)(s+1)+2}, \ldots, x_{\epsilon(s+1)})$, $\epsilon \in \{1,\ldots,\gamma+\delta\}$, of $\tau \in \mathrm{PAut}(\mathcal{C})$ splits into two disjoint cycles of the same length via Φ, that is,

$$\begin{aligned}&\Phi((i_\epsilon, x_{(\epsilon-1)(s+1)+2}, \ldots, x_{\epsilon(s+1)})) = \\ &(2i_\epsilon - 1, 2x_{(\epsilon-1)(s+1)+2} - 1, \ldots, 2x_{\epsilon(s+1)} - 1) \\ &(2i_\epsilon, 2x_{(\epsilon-1)(s+1)+2}, \ldots, 2x_{\epsilon(s+1)}).\end{aligned}$$

Furthermore, the information positions of the set $I = \Phi(\mathcal{I})$ are also placed in different cycles of length $s+1$ of the permutation $\sigma = \Phi(\tau)$. There is again one information position per cycle of length $s+1$, with the exception of the cycles of the form $(2i_\epsilon, 2x_{(\epsilon-1)(s+1)+2} - 1, \ldots, 2x_{\epsilon(s+1)})$ for all $\epsilon \in \{\delta+1,\ldots,\gamma+\delta\}$.

Let $S = \{\sigma^i\}_{i=1}^{s+1}$ and let $P = \{1,\ldots,2\beta\}$ be the set of all coordinate positions. We define the set $A_i = \{j \in P : \sigma^i(j) \in I\}$ for each $i \in \{1,\ldots,s+1\}$. Note that $|A_i| = \gamma + 2\delta$ and $A_i \cap A_j = \emptyset$ for all $i,j \in \{1,\ldots,s+1\}$, $i \neq j$. We have to prove that every s-set of coordinate positions, denoted by $J = \{j_1,\ldots,j_s\} \subseteq P$, is moved out of I by at least one element of S. Note that a coordinate position in J cannot be in two different sets A_i, $i \in \{1,\ldots,s+1\}$. In the worst-case scenario, for each $k \in \{1,\ldots,s\}$, $j_k \in A_{l_k}$ for some $l_k \in \{1,\ldots,s+1\}$. However, since $|J| = s$ and $|S| = s+1$, we can always assure that there is $\varphi \in S$ such that $\varphi(J) \cap I = \emptyset$. Thus, by Lemma 39, $S = \{\Phi(\tau)^i\}_{i=1}^{s+1} = \{\Phi(\tau^i)\}_{i=1}^{s+1}$ is an s-PD-set of size $s+1$ for the $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$ with information set $I = \Phi(\mathcal{I})$.

Using the same argument as before, we can prove that any subset $R \subseteq S$ of size $r+1$ is an r-PD-set for C, $r \in \{1,\ldots,s\}$. Let $J = \{j_1,\ldots,j_r\} \subseteq P$ be an r-set of coordinate positions. Since $|J| = r$ and $|R| = r+1$, again it is clear that there is $\varphi \in R$ such that $\varphi(J) \cap I = \emptyset$. □

Corollary 61. *Let S be an s-PD-set of size $s+1$ for a $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$ of length 2β and type $2^\gamma 4^\delta$ as in Theorem 60. Then $s+1$ divides the order of $\mathrm{PAut}(\mathcal{C})$ and $s \leq f_{\mathcal{C}}$, where*

$$f_{\mathcal{C}} = \lfloor (\beta - \gamma - \delta)/(\gamma + \delta) \rfloor .$$

Proof. We know that $S = \{\Phi(\tau^i)\}_{i=1}^{s+1}$ for a certain $\tau \in \mathrm{PAut}(\mathcal{C}) \subseteq \mathrm{Sym}(\beta)$ with at least $\gamma + \delta$ cycles of length $s+1$. A counting argument shows that the condition $(s+1)(\gamma+\delta) \leq \beta$ must be met. Then $s \leq f_{\mathcal{C}}$.

Let $\tau = \tau_1 \cdots \tau_l$ be considered as a product of its disjoint cycles. Since $\mathrm{ord}(\tau) = \mathrm{lcm}(\mathrm{ord}(\tau_1), \ldots, \mathrm{ord}(\tau_l))$, it follows that $\mathrm{ord}(\tau_j)$ divides $\mathrm{ord}(\tau)$ for all $j \in \{1, \ldots, l\}$. On the other hand, $\mathrm{ord}(\tau)$ must divide $|\mathrm{PAut}(\mathcal{C})|$. Since $\mathrm{ord}(\tau_j) = s+1$ for at least $\gamma + \delta$ cycles, the result follows. □

By Corollary 46, if $S = \Phi(\mathcal{S})$, where $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$, is an s-PD-set of size $s+1$ for $H_{\gamma,\delta} = \Phi(\mathcal{H}_{\gamma,\delta})$, then $s \leq f_{\gamma,\delta}$, where

$$f_{\gamma,\delta} = \left\lfloor (2^{\gamma+2\delta-2} - \gamma - \delta)/(\gamma + \delta) \right\rfloor .$$

Furthermore, by Lemma 17, if $S \subseteq \mathrm{PAut}(H_{\gamma,\delta})$ is an s-PD-set of size $s+1$ for $H_{\gamma,\delta}$, then $s \leq f_m$, where

$$f_m = \lfloor (2^m - m - 1)/(1+m) \rfloor .$$

Note that $f_{\gamma,\delta} \leq f_m$, where $m = \gamma + 2\delta - 1$. Moreover, $f_{\mathcal{H}_{\gamma,\delta}} = f_{\gamma,\delta}$ despite the fact that $f_{\mathcal{H}_{\gamma,\delta}}$ takes into account the restrictions given by Theorem 60.

Example 62. *Let $\mathcal{H}_{0,3}$ be the quaternary linear Hadamard code of length 16 and type $2^0 4^3$ with generator matrix $\mathcal{G}_{0,3}$ given in Example 38 and obtained by applying (2.12) two times starting from $\mathcal{G}_{0,1} = (1)$. The matrix $\mathcal{G}_{0,3}$ is also the matrix (2.5) given in Example 4. Let*

$$\tau = (1, 16, 11, 6)(2, 7, 12, 13)(3, 14, 9, 8)(4, 5, 10, 15) \in \mathrm{PAut}(\mathcal{H}_{0,3})$$

[PPV14]. Note that τ has four disjoint cycles of length four. By Proposition 37, $\mathcal{I} = \{1, 2, 5\}$ is a quaternary information set for $\mathcal{H}_{0,3}$. Moreover, note that each quaternary information position in $\mathcal{I}$ is in a different cycle of τ. Let $\sigma = \Phi(\tau) \in \mathrm{PAut}(H_{0,3}) \subseteq \mathrm{Sym}(32)$, where $H_{0,3} = \Phi(\mathcal{H}_{0,3})$. Thus, by Lemma 39 and Theorem 60, $S = \{\sigma, \sigma^2, \sigma^3, \sigma^4\} \subseteq \mathrm{PAut}(H_{0,3})$ is a 3-PD-set of size 4 for the $\mathbb{Z}_4$-linear Hadamard code $H_{0,3}$ with information set $\Phi(\mathcal{I}) = \{1, 2, 3, 4, 9, 10\}$. Note that $H_{0,3}$ is the smallest $\mathbb{Z}_4$-linear Hadamard code which is nonlinear.

In this case, we have that $f_5 = f_{0,3} = 4$. *For example, a* 4-PD-*set of size* 5 *for* $H_{0,3}$ *is found in Example 51. Note that it is enough to consider permutations in the subgroup* $\Phi(\mathrm{PAut}(\mathcal{H}_{0,3})) \leq \mathrm{PAut}(H_{0,3})$ *to achieve* f_5. *However, a* 4-PD-*set of size 5 cannot be generated by a single permutation* $\sigma \in \mathrm{PAut}(H_{0,3})$ *by using Theorem 60, since* 5 *does not divide* $|\mathrm{PAut}(\mathcal{H}_{0,3})| = 2^9 \cdot 3$ *[PPV14] (or by Proposition 41).*

Example 63. *Let* $\mathcal{H}_{1,3}$ *be the quaternary linear Hadamard code of length* 32 *and type* $2^1 4^3$ *with generator matrix*

$$\mathcal{G}_{1,3} = \begin{pmatrix} \mathcal{G}_{0,3} & \mathcal{G}_{0,3} \\ \mathbf{0} & \mathbf{2} \end{pmatrix}$$

obtained by applying (2.11) over the matrix $\mathcal{G}_{0,3}$ *given in Example 62. Let* $\tau \in \mathrm{PAut}(\mathcal{H}_{1,3}) \subseteq \mathrm{Sym}(32)$ *be*

$$\begin{aligned} \tau = \ & (1,24,26,15,3,22,28,13)(2,23,27,14,4,21,25,16) \\ & (5,11,32,20,7,9,30,18)(6,10,29,19,8,12,31,17), \end{aligned}$$

which has four disjoint cycles of length eight. By Proposition 37, $\mathcal{I} = \{1,2,5,17\}$ *is a quaternary information set for* $\mathcal{H}_{1,3}$. *Each quaternary information position in* $\mathcal{I}$ *is in a different cycle of* τ. *Let* $\sigma = \Phi(\tau) \in \mathrm{PAut}(H_{1,3}) \subseteq \mathrm{Sym}(64)$, *where* $H_{1,3} = \Phi(\mathcal{H}_{1,3})$. *Thus, by Lemma* 39 *and Theorem* 60, $S = \{\sigma^i\}_{i=1}^{8}$ *is a* 7-PD-*set of size* 8 *for the* $\mathbb{Z}_4$-*linear Hadamard code* $H_{1,3}$ *with information set* $\Phi(\mathcal{I}) = \{1,2,3,4,9,10,33\}$. *Note that* $H_{1,3}$ *is a binary nonlinear code, since* $\delta \geq 3$ *[Kro01].*

Since $f_{1,3} = 7$, *no better* s-PD-*sets of size* $s+1$ *can be found by using permutations in the subgroup* $\Phi(\mathrm{PAut}(\mathcal{H}_{1,3})) \leq \mathrm{PAut}(H_{1,3})$. *However, an* 8-PD-*set of size* 9 *could be theoretically found in* $\mathrm{PAut}(H_{1,3})$, *since* $f_6 = 8$. *Unlike Example 62, there is need to invoke the method presented in Chapter 4 to obtain an* s-PD-*set of size* $s+1$ *achieving the upper bound* $f_{1,3}$ *for* $H_{1,3}$.

5.2 Explicit construction for $\mathbb{Z}_4$-linear Hadamard codes

In this section, we give an explicit construction of s-PD-sets of minimum size $s+1$ for $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ with $\delta \geq 4$ by finding a permutation $\tau \in \mathrm{PAut}(\mathcal{H}_{\gamma,\delta})$ that satisfies the conditions of Theorem 60. The results obtained in this section regarding the order of elements in $\mathrm{PAut}(\mathcal{H}_{0,\delta})$ complement the comprehensive study of this permutation automorphism group started in [PPV14].

In Chapter 4, we show that $\mathrm{PAut}(\mathcal{H}_{0,\delta})$, the permutation automorphism group of $\mathcal{H}_{0,\delta}$, is isomorphic to the following set of matrices over $\mathbb{Z}_4$, denoted in this chapter by $\mathcal{A}_\delta$:

$$\mathcal{A}_\delta = \left\{ \begin{pmatrix} 1 & \eta \\ \mathbf{0} & A \end{pmatrix} \ : \ A \in \mathrm{GL}(\delta-1, \mathbb{Z}_4),\ \eta \in \mathbb{Z}_4^{\delta-1} \right\} \subseteq \mathrm{GL}(\delta, \mathbb{Z}_4).$$

When dealing with the order of nonsingular matrices, the following results are useful.

Lemma 64 ([Max51])**.** *Let p be a prime. Let P be a partition of n,*

$$n = N + n_1 + \cdots + n_h,$$

with $N \geq 0$, $h \geq 1$, $2 \leq n_1 < n_2 < \cdots < n_h$. Let $p\{y\}$ designate the first in the series $1, p, p^2, \ldots$ that equals or exceeds y. Then the orders of the nonsingular matrices (mod p) *of size $n \times n$ are*

$$\begin{aligned} f &= p\{n\}(p-1) \\ g_P &= p\{N\}\,\mathrm{lcm}(p^{n_1}-1, p^{n_2}-1, \ldots, p^{n_h}-1) \end{aligned}$$

and their divisors, taken over all possible partitions P.

Lemma 65 ([Max51])**.** *The orders of the nonsingular matrices* (mod p^a) *of size $n \times n$ for $n > 1$ are $p^{a-1}f, p^{a-1}g_P$, and their divisors, with f and g_P as in Lemma 64.*

Corollary 66. *The maximum order of a matrix $A \in \mathrm{GL}(k, \mathbb{Z}_4)$ is $2(2^k-1)$.*

Proof. By Lemma 64, one can easily check that $\max\{\mathrm{ord}(A) : A \in \mathrm{GL}(k, \mathbb{Z}_2)\} = 2^k - 1$. Therefore, by Lemma 65, $\max\{\mathrm{ord}(A) : A \in \mathrm{GL}(k, \mathbb{Z}_4)\} = 2(2^k-1)$. □

It is well known that the order of a permutation $\tau \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$ (or equivalently, the corresponding matrix $\mathcal{M} \in \mathcal{A}_\delta$) must divide the order of $\mathrm{PAut}(\mathcal{H}_{0,\delta})$. We can strengthen this result by studying the possible orders of a matrix $\mathcal{M} \in \mathcal{A}_\delta$.

Lemma 67. *Let $A \in \mathrm{GL}(k, \mathbb{Z}_4)$. Then*

$$\mathrm{ord}\begin{pmatrix} 1 & \eta \\ \mathbf{0} & A \end{pmatrix} = \lambda \cdot \mathrm{ord}(A), \text{ where } \lambda \in \{1, 2, 4\}.$$

Proof. Denote the order of A by r. Since

$$\begin{pmatrix} 1 & \eta \\ \mathbf{0} & A \end{pmatrix}^{\lambda r+i} =$$

$$\begin{pmatrix} 1 & \eta[(\lambda+1)\,\mathrm{Id}+\cdots+(\lambda+1)A^{i-1}+\lambda A^{i}+\cdots+\lambda A^{r-1}] \\ \mathbf{0} & A^{\lambda r+i} \end{pmatrix},$$

the order of $\begin{pmatrix} 1 & \eta \\ \mathbf{0} & A \end{pmatrix}$ must be a multiple of r. Moreover, $\lambda \in \{1,2,4\}$ since

$$\begin{pmatrix} 1 & \eta \\ \mathbf{0} & A \end{pmatrix}^{4r} = \begin{pmatrix} 1 & 4\eta(\mathrm{Id}+\cdots+A^{r-1}) \\ \mathbf{0} & A^{4r} \end{pmatrix} = \mathrm{Id}_{(k+1)\times(k+1)}\,.$$

□

Lemma 68. *Let $\tau \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$. Then*

$$\mathrm{ord}(\tau) \in \{\lambda{\cdot}\mathrm{ord}(A) : A \in \mathrm{GL}(\delta{-}1,\mathbb{Z}_4),\ \lambda \in \{1,2,4\},\ \lambda{\cdot}\mathrm{ord}(A) \le 2(2^{\delta}{-}1)\}.$$

Proof. By Corollary 66, $\mathrm{ord}(\tau)$ is always bounded by $2(2^{\delta}-1)$, since $\mathcal{A}_{\delta} \subseteq \mathrm{GL}(\delta,\mathbb{Z}_4)$. This fact together with Lemma 67 lead us to the result. □

In any case, if $\eta = \mathbf{0}$, then

$$\mathrm{ord}\begin{pmatrix} 1 & \eta \\ \mathbf{0} & A \end{pmatrix} = \mathrm{ord}(A).$$

Example 69. *We give the possible orders of a permutation τ in the permutation automorphism group $\mathrm{PAut}(\mathcal{H}_{0,4})$ of the quaternary linear Hadamard code $\mathcal{H}_{0,4}$ of length 64 and type $2^0 4^4$. We know that $\mathrm{ord}(\tau)$ must divide $|\mathrm{PAut}(\mathcal{H}_{0,4})| = 2^{18}\cdot 3\cdot 7$ [PPV14] (or by Proposition 41). By Lemma 64, taking $p=2$ and $n=3$, we obtain that the orders of all matrices in $\mathrm{GL}(3,\mathbb{Z}_2)$ are $\{1,2,3,4,7\}$. Indeed, there exist only two admissible partitions as in Lemma 64, denoted here by P_1 and P_2, obtaining respectively,*

$$3 = N + n_1 = 0 + 3,$$
$$3 = N + n_1 = 1 + 2.$$

Then, $f = p\{3\} = 4, g_{P_1} = 7$, and $g_{P_2} = 3$. By Lemma 65, the orders of all matrices in $\mathrm{GL}(3,\mathbb{Z}_4)$ are $\{1,2,3,4,6,7,8,14\}$. By Lemma 68,

$$\mathrm{ord}(\tau) \in \{1,2,3,4,6,7,8,14,16,24,28\}.$$

Nevertheless, 16, 24, *and* 28 *are not suitable orders for* $\tau \in \mathrm{PAut}(\mathcal{H}_{0,4})$*. Indeed, an equivalent study of the orders of elements in* $\mathrm{GL}(4,\mathbb{Z}_4)$ *shows that these cannot be the order of an element in* $\mathrm{GL}(4,\mathbb{Z}_4)$*. Note that*

$$\begin{aligned} 14 &= \max\{\mathrm{ord}(A) : A \in \mathrm{GL}(3,\mathbb{Z}_4)\} \\ &= \max\{\mathrm{ord}(\tau) : \tau \in \mathrm{PAut}(\mathcal{H}_{0,4})\}. \end{aligned}$$

Note that in order to obtain the best s-PD-sets by using Theorem 60, the more suitable candidates are the high order permutations. It does not seem very restrictive to focus on obtaining permutations $\tau \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$ of order $2(2^{\delta-1}-1)$ to acquire suitable candidates for $\mathcal{H}_{0,\delta}$, in view of the results obtained for $\delta = 4$ in Example 69.

Although we can characterize when a permutation $\tau \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$ has order $2(2^{\delta-1}-1)$, we do not know its cyclic structure. Recall that, in order to apply Theorem 60, we need a permutation $\tau \in \mathrm{PAut}(\mathcal{H}_{0,\delta}) \subseteq \mathrm{Sym}(\beta)$ with at least δ disjoint cycles of the same length. Next, we show how some known results on maximum length sequences over $\mathbb{Z}_4$ can be used to solve this question.

Let $f(x) = x^k - a_{k-1}x^{k-1} - \cdots - a_1 x - a_0 \in \mathbb{Z}_4[x]$. The kth-order homogeneous linear recurrence relation over $\mathbb{Z}_4$ with characteristic polynomial $f(x)$ is

$$s_{n+k} = a_{k-1}s_{n+k-1} + \cdots + a_1 s_{n+1} + a_0 s_n, \;\; n = 0, 1, \ldots \tag{5.2}$$

Let $\{s_n\}_{n=0}^{\infty}$ be a nonzero sequence over $\mathbb{Z}_4$ satisfying (5.2). For each $n \geq 0$, we define the tuple $\mathbf{s}_n = (s_n, \ldots, s_{n+k-1})$ over $\mathbb{Z}_4$. In the language of feedback shift registers, $\mathbf{s}_n$ is called the *n-state vector*.

The set of all nonzero sequences $\{s_n\}_{n=0}^{\infty}$ over $\mathbb{Z}_4$ satisfying (5.2) whose characteristic polynomial $f(x)$ is a primitive basic irreducible polynomial dividing $x^{2(2^k-1)} - 1$ in $\mathbb{Z}_4[x]$ is called Family $\mathcal{B}$ in [ZF15]. It is known that there are $2^{k-1}+1$ cyclically distinct periodic sequences in each Family $\mathcal{B}$: 2^{k-1} of them with common period $2(2^k-1)$ and one with period 2^k-1 [BHK92]. Moreover, the sequence with period 2^k-1 is the unique containing only zero-divisors. Examples of primitive basic irreducible polynomials $f(x) \in \mathbb{Z}_4[x]$ suitable for constructing sequences of Family $\mathcal{B}$ for degrees 3 to 10 can be found in [BHK92, Table III] and also in [ZF15, Table 5.8].

Example 70. *Let* $f(x) = x^3 + x + 1 \in \mathbb{Z}_4[x]$*. It is easy to see that* $\mu(f(x)) = x^3 + x + 1 \in \mathbb{Z}_2[x]$ *is primitive over* $\mathbb{Z}_2[x]$*, so* $f(x)$ *is a primitive basic irreducible polynomial. It is also easy to check that* $f(x)$ *divides* $x^{14} - 1$*.*

Consider the linear recurrence relation over $\mathbb{Z}_4$ with characteristic polynomial $f(x)$, that is,

$$s_{n+3} = 3s_{n+1} + 3s_n, \quad n = 0, 1, \ldots \tag{5.3}$$

We now give all nonzero sequences $\{s_n\}_{n=0}^{\infty}$ over $\mathbb{Z}_4$ satisfying (5.3) whose characteristic polynomial is $f(x)$. By different choices of initial state vectors $\mathbf{s}_0$, we obtain the following sequences $\{s_n\}_{n=0}^{\infty}$:

$\mathbf{s}_0$	$\{s_n\}_{n=0}^{\infty}$
100	10030113203213
300	30010331201231
110	11023233322303
210	21013302121112
200	2002022

Note that we show only the nonrepeated part of those sequences. Note also that there are 5 periodic sequences in the Family $\mathcal{B}$ generated by $f(x)$. Four of them have period $2(2^3-1) = 14$ and the remaining one has period $2^4-1 = 7$. As we can see, this last sequence is the one with initial state vector $\mathbf{s}_0 = (2,0,0)$ and the unique containing only zero-divisors (only zeros and twos).

Let $\overline{\mathcal{G}_{0,\delta}}$ denote the generator matrix $\mathcal{G}_{0,\delta}$, described in Section 2.3 for the quaternary linear Hadamard code $\mathcal{H}_{0,\delta}$ of length $\beta = 4^{\delta-1}$ and type 4^{δ}, without the first row $(1,\ldots,1)$. Let w_i denote the ith column vector of $\overline{\mathcal{G}_{0,\delta}}$. Since $\overline{\mathcal{G}_{0,\delta}}$ has as columns all the different vectors from $\mathbb{Z}_4^{\delta-1}$, each state vector $\mathbf{s}_n \in \mathbb{Z}_4^{\delta-1}$ in the different sequences $\{s_n\}_{n=0}^{\infty}$ from the Family $\mathcal{B}$ generated by a primitive basic irreducible polynomial $f(x)$ of degree $\delta-1$ represents a column vector w_i of $\overline{\mathcal{G}_{0,\delta}}$. Sequences $\{s_n\}_{n=0}^{\infty}$ are all different and they do not share any state vector, so the correspondence between state vectors $\mathbf{s}_n$ and column vectors w_i is one-to-one. Thus, define $\tau_f \in \mathrm{PAut}(\mathcal{H}_{0,\delta}) \subseteq \mathrm{Sym}(\beta)$ as follows: $\tau(i) = j$ as long as w_i and w_j are consecutive state vectors of a sequence $\{s_n\}_{n=0}^{\infty}$.

Proposition 71. *Let $\mathcal{H}_{0,\delta}$ be the quaternary linear Hadamard code of length $\beta = 4^{\delta-1}$ and type 4^{δ} with $\delta \geq 4$ generated by $\mathcal{G}_{0,\delta}$. Let $f(x) \in \mathbb{Z}_4[x]$ be a primitive basic irreducible polynomial of degree $\delta-1$ dividing $x^{2(2^{\delta-1}-1)}-1$ in $\mathbb{Z}_4[x]$. Then, the permutation τ_f has order $\mathrm{ord}(\tau_f) = 2(2^{\delta-1}-1)$ in the group $\mathrm{PAut}(\mathcal{H}_{0,\delta})$. Moreover, τ_f has $2^{\delta-2}+1$ disjoint cycles, where $2^{\delta-2}$ of them have length $2(2^{\delta-1}-1)$ and one of them has length $2^{\delta-1}-1$. Finally, the permutation τ_f has only one fixed point.*

Proof. There are $2^{\delta-2}+1$ sequences in the Family $\mathcal{B}$ defined by the polynomial $f(x) = x^{\delta-1} - a_{\delta-2}x^{\delta-2} - \cdots - a_1 x - a_0 \in \mathbb{Z}_4[x]$: $2^{\delta-2}$ of them with common period $2(2^{\delta-1}-1) = 2^\delta - 2$ and the remaining one with period $2^{\delta-1}-1$. All different state vectors $\mathbf{s}_0, \ldots, \mathbf{s}_\ell$, where $\ell \in \{2^{\delta-1}-2, 2^\delta - 3\}$, from the same sequence $\{s_n\}_{n=0}^{\infty}$ define a disjoint cycle of the permutation $\tau_f \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$. Thus,

$$\tau_f = \prod_{i=1}^{2^{\delta-2}+1} \tau_i$$

where $\tau_1, \ldots, \tau_{2^{\delta-2}}$ have length $2(2^{\delta-1}-1)$ and $\tau_{2^{\delta-2}+1}$ has length $2^{\delta-1}-1$. Moreover,

$$\begin{aligned} \mathrm{ord}(\tau_f) &= \mathrm{lcm}(\{\mathrm{ord}(\tau_i) : 1 \le i \le 2^{\delta-2}+1\}) \\ &= \mathrm{lcm}(\{2(2^{\delta-1}-1), 2^{\delta-1}-1\}) \\ &= 2(2^{\delta-1}-1). \end{aligned} \quad .$$

Finally, since

$$\begin{aligned} 2^{\delta-2}[2(2^{\delta-1}-1)] + (2^{\delta-1}-1) &= (2^{\delta-1}+1)(2^{\delta-1}-1) \\ &= 2^{2\delta-2} - 1 \\ &= \beta - 1, \end{aligned}$$

the permutation τ_f has only one fixed point. Indeed, $\tau_f(1) = 1$. □

Corollary 72. *Let $f(x) \in \mathbb{Z}_4[x]$ be a primitive basic irreducible polynomial of degree $\delta - 1$ dividing $x^\lambda - 1$ in $\mathbb{Z}_4[x]$, where $\lambda = 2(2^{\delta-1}-1)$ and $\delta \ge 4$. Let $\mathcal{I}$ be a quaternary information set for $\mathcal{H}_{0,\delta}$ with exactly one quaternary information position per cycle of length λ of τ_f. Then $S = \{\Phi(\tau_f^i)\}_{i=1}^{\lambda}$ is a $(\lambda-1)$-PD-set of size λ for $H_{0,\delta} = \Phi(\mathcal{H}_{0,\delta})$ with information set $\Phi(\mathcal{I})$.*

Proof. By Proposition 71, τ_f has $2^{\delta-2}$ disjoint cycles of length λ. Since $|\mathcal{H}_{0,\delta}| = 4^\delta$, the set $\mathcal{I}$ has size δ. Note that $2^{\delta-2} \ge \delta$ if and only if $\delta \ge 4$. By Theorem 60, the result holds. □

Example 73. *Let $f(x) \in \mathbb{Z}_4[x]$ be the primitive basic irreducible polynomial dividing $x^{14} - 1$ in $\mathbb{Z}_4[x]$ introduced in Example 70. By Propisition 71, the permutation $\tau_f \in \mathrm{PAut}(\mathcal{H}_{0,4}) \in \mathrm{Sym}(64)$ has order 14. In addition, its cyclic structure behaves as follows: four disjoint cycles of length 14 and one cycle of length 7. Note that*

$$\begin{aligned} \tau_f = \; & (\mathbf{2}, 49, 13, 20, 21, 54, 46, 12, 51, 45, 28, 55, 30, 8) \\ & (3, 33, 9, 35, 41, 43, 11) \\ & (4, 17, \mathbf{5}, 50, 61, 32, 40, 10, 19, 37, 58, 31, 56, 14) \\ & (\mathbf{6}, 34, 57, 47, 60, 63, 64, 48, 44, 59, 15, 52, 29, 24) \\ & (7, \mathbf{18}, 53, 62, 16, 36, 25, 39, 26, 23, 22, 38, 42, 27). \end{aligned}$$

δ	s	$f_{0,\delta} = f_m$
3	**3**	4
4	**13**	15
5	**29**	50
6	**61**	169
7	**125**	584
8	**253**	2047

Table 5.1: Maximum s for which an s-PD-set of size $s+1$ generated by a single permutation is found for some codes $H_{0,\delta}$.

It is easy to check that $\mathcal{I} = \{2, 5, 6, 18\}$ is a quaternary information set for $\mathcal{H}_{0,4}$ with generator matrix $\mathcal{G}_{0,4}$ obtained by applying (2.12) three times starting from $\mathcal{G}_{0,1} = (1)$. Note that each quaternary information position in $\mathcal{I}$ is in a different cycle of length 14 of τ_f. By Corollary 72, $S = \{\Phi(\tau_f^i)\}_{i=1}^{14}$ is a 13-PD-set of size 14 for the $\mathbb{Z}_4$-linear Hadamard code $H_{0,4}$ with information set $I = \Phi(\mathcal{I}) = \{3, 4, 9, 10, 11, 12, 35, 36\}$. In practice, it is not difficult to find such a set $\mathcal{I}$. For example, computations in Magma *software package [BPPV16, BCFS16] shows that, in this case, there are 10752 suitable quaternary information sets.*

We have that $f_7 = f_{0,4} = 15$. An s-PD-set of size $s+1$, $s \in \{14, 15\}$, for $H_{0,4}$ cannot be obtained by using Theorem 60. Indeed, a permutation $\tau \in \mathrm{PAut}(\mathcal{H}_{0,4})$ as in Theorem 60 needs at least 4 disjoint cycles of length $s+1$, so $\mathrm{ord}(\tau) \geq 15$ if $s \in \{14, 15\}$. This is not possible by Example 69.

Table 5.1 shows the maximum values of s for which an s-PD-set of size $s+1$ can be constructed by using Corollary 72, together with the values of the upper bound $f_{0,\delta} = f_m$ attained by using Theorem 50 and shown in Table 4.1.

Towards a better understanding of the relation between sequences over $\mathbb{Z}_4$ generated by a polynomial $f(x)$ (the previously called Family $\mathcal{B}$ under some assumptions on $f(x)$) and the permutation $\tau_f \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$, we introduce the notion of companion matrices. In general, the *companion matrix* of a polynomial $f(x) = x^k - a_{k-1}x^{k-1} - \cdots - a_1 x - a_0$, denoted by A_f, is the $k \times k$

matrix defined as

$$A_f = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & 0 & \cdots & 0 & a_1 \\ 0 & 1 & 0 & \cdots & 0 & a_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a_{k-1} \end{pmatrix}. \tag{5.4}$$

Let $\mathcal{M}_f$ be the matrix

$$\mathcal{M}_f = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & A_f \end{pmatrix}.$$

Lemma 74. *If $f(x) = x^{\delta-1} - a_{\delta-2}x^{\delta-2} - \cdots - a_1 x - a_0 \in \mathbb{Z}_4[x]$ is a primitive basic irreducible polynomial of degree $\delta - 1$, then $\mathcal{M}_f \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$.*

Proof. If $f(x)$ is a primitive basic irreducible polynomial, then its independent coefficient $a_0 \in \{1, 3\}$. Indeed, since $\mu(f(x))$ is a primitive polynomial over $\mathbb{Z}_2[x]$, we can assure that $a_0 \equiv 1 \pmod 2$. On the other hand, $\mathcal{M}_f \in \mathrm{PAut}(\mathcal{H}_{0,\delta})$ as long as $A_f \in \mathrm{GL}(\delta - 1, \mathbb{Z}_4)$. Since $\det(A_f) = (-1)^{\delta-1}a_0 \in \{1, 3\}$, we obtain the desired result. □

Note that if A_f is the companion matrix of the polynomial $f(x)$ associated with (5.2), then it holds that $\mathbf{s}_n = \mathbf{s}_0 A_f^n$. Moreover, the matrix $\mathcal{M}_f$ viewed as a permutation of coordinate positions is precisely τ_f.

Example 75. *The companion matrix of the polynomial $f(x) = x^3 + x + 1 \in \mathbb{Z}_4[x]$ introduced in Example 70 is*

$$A_f = \begin{pmatrix} 0 & 0 & 3 \\ 1 & 0 & 3 \\ 0 & 1 & 0 \end{pmatrix}.$$

It is straightforward to check that the n-state vector $\mathbf{s}_n$, for any of the five different sequences $\{s_n\}_{n=0}^{\infty}$ presented in Example 70, may be obtained as $\mathbf{s}_0 A_f^n$, where $\mathbf{s}_0$ is the initial state vector we have used for that sequence. The matrix $\mathcal{M}_f \in \mathrm{PAut}(\mathcal{H}_{0,4}) \subseteq \mathrm{GL}(4, \mathbb{Z}_4)$ associated with the permutation $\tau_f \in \mathrm{PAut}(\mathcal{H}_{0,4}) \subseteq \mathrm{Sym}(\beta)$ introduced in Example 73 is then

$$\mathcal{M}_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

We can also generate s-PD-sets of size $s+1$ under the conditions of Theorem 60 for $H_{\gamma,\delta}$ with $\gamma > 0$. We proceed as follows: First, we compute an s-PD-set S of size $s+1$ for $H_{0,\delta}$, for example, by using Corollary 72. Then applying Proposition 54 recursively γ times over S, we obtain an s-PD-set of size $s+1$ for $H_{\gamma,\delta}$, with $\gamma > 0$.

Corollary 76. *Let $f(x) \in \mathbb{Z}_4[x]$ be a primitive basic irreducible polynomial of degree $\delta - 1$ dividing $x^\lambda - 1$ in $\mathbb{Z}_4[x]$, where $\lambda = 2(2^{\delta-1} - 1)$ and $\delta \geq 4$. Let $\mathcal{I}$ be a quaternary information set for $\mathcal{H}_{0,\delta}$ with exactly one quaternary information position per cycle of length λ of τ_f. Then $2^\gamma\{\Phi(\tau_f^i)\}_{i=1}^{\lambda}$ is a $(\lambda-1)$-PD-set of size λ for $H_{\gamma,\delta} = \Phi(\mathcal{H}_{\gamma,\delta})$ with information set obtained by applying recursively Proposition 37 over $\Phi(\mathcal{I})$.*

Example 77. *Let $\mathcal{H}_{1,4}$ be the quaternary linear Hadamard code of length 128 and type $2^1 4^4$ with generator matrix $\mathcal{G}_{1,4}$ obtained by applying (2.11) over the matrix $\mathcal{G}_{0,4}$ given in Example 73. Let $S \subseteq \mathrm{PAut}(H_{0,4})$ and $I \subseteq \{1, \ldots, 128\}$ as defined in the same example. Then $(S|S)$ is a 13-PD-set of size 14 for the $\mathbb{Z}_4$-linear Hadamard code $H_{1,4}$ with any information set of the form $I \cup \{128+i\}$, where $i \in I$, by Corollary 76.*

The set of all nonzero sequences $\{s_n\}_{n=0}^{\infty}$ over $\mathbb{Z}_4$ satisfying (5.2) whose characteristic polynomial $f(x)$ is a primitive basic irreducible polynomial dividing $x^{2^k-1} - 1$ in $\mathbb{Z}_4[x]$ (that is, the Hensel lift of a binary primitive polynomial) is called Family $\mathcal{A}$ in [BHK92]. In [BHK92, Table I] and [Wan97, Table 6.1], the primitive basic irreducible polynomials over $\mathbb{Z}_4$ up to degree 10 that are suitable for generating Family $\mathcal{A}$ are listed. Although this family may also be used to obtain s-PD-sets, they are not as good as those generated by Family $\mathcal{B}$ in this section. Specifically, a $(2^{\delta-1} - 2)$-PD-set of size $2^{\delta-1} - 1$ for $H_{\gamma,\delta}$ can be found by using Family $\mathcal{A}$, for all $\delta \geq 3$ and $\gamma \geq 0$.

5.3 Explicit construction for $\mathbb{Z}_4$-linear Kerdock codes

In this section, we obtain s-PD-sets of minimum size $s+1$, for all $m \geq 5$ and $1 < s \leq \lambda - 1$, for the $\mathbb{Z}_4$-linear Kerdock code of length 2^m such that $2^{m-1} - 1$ is not prime, where λ is the greatest divisor of $2^{m-1} - 1$ satisfying $\lambda \leq 2^{m-1}/m$.

Let $h(x)$ be a primitive basic irreducible polynomial of degree $m-1$ over $\mathbb{Z}_4$ such that $h(x)$ divides $x^n - 1$, $n = 2^{m-1} - 1$, and let $g(x)$ be the

reciprocal polynomial to the polynomial $(x^n-1)/((x-1)h(x))$. Recall that the shortened quaternary Kerdock code $\mathcal{K}_m^-$ is the quaternary linear cyclic code of length $n = 2^{m-1}-1$ with generator polynomial $g(x)$. The quaternary Kerdock code $\mathcal{K}_m$ is the code obtained from $\mathcal{K}_m^-$ by adding a zero-sum check symbol at the end of each codeword of $\mathcal{K}_m^-$. The $\mathbb{Z}_4$-linear Kerdock code K_m is defined to be the Gray map image $\Phi(\mathcal{K}_m)$ of the quaternary Kerdock code $\mathcal{K}_m$. Since $\mathcal{K}_m^-$ is a cyclic code of length $n = 2^{m-1}-1$, it is clear that $(1,\ldots,2^{m-1}-1) \in \mathrm{PAut}(\mathcal{K}_m^-)$. Therefore, by definition of $\mathcal{K}_m$, we also have that $(1,\ldots,2^{m-1}-1) \in \mathrm{PAut}(\mathcal{K}_m)$.

Corollary 78. *Let $\mathcal{K}_m$ be the quaternary Kerdock code of length 2^{m-1} and type 4^m such that $2^{m-1}-1$ is not a prime number. Let $\nu = (1,\ldots,2^{m-1}-1) \in \mathrm{PAut}(\mathcal{K}_m) \subseteq \mathrm{Sym}(2^{m-1})$. Let λ be the greatest divisor of $2^{m-1}-1$ such that $\lambda \leq 2^{m-1}/m$ and μ satisfying that $\lambda\mu = 2^{m-1}-1$. Then $S = \{\Phi(\nu^{i\cdot\mu})\}_{i=1}^{\lambda}$ is a $(\lambda-1)$-PD-set of size λ for $K_m = \Phi(\mathcal{K}_m)$ with information set $I = \{1,\ldots,2m\}$.*

Proof. Let $g(x) = g_0 + g_1x + \cdots + g_rx^r \in \mathbb{Z}_4[x]$ be a generator polynomial of $\mathcal{K}_m^-$, where $r = 2^{m-1}-m-1$. Then the following $m \times (2^{m-1}-1)$ matrix

$$\begin{pmatrix} g_0 & g_1 & \cdots & g_m & & & \\ & g_0 & g_1 & \cdots & g_m & & \\ & & \ddots & & & \ddots & \\ & & & g_0 & g_1 & \cdots & g_m \end{pmatrix} \tag{5.5}$$

is a generator matrix of $\mathcal{K}_m^-$. From matrix (5.5), it is clear that $\mathcal{I} = \{1,\ldots,m\}$ is a quaternary information set for $\mathcal{K}_m^-$ and so for its extended code $\mathcal{K}_m$. Thus, $I = \Phi(\mathcal{I}) = \{1,\ldots,2m\}$ becomes a valid information set for the $\mathbb{Z}_4$-linear Kerdock code K_m. Finally, the result holds since $\nu \in \mathrm{PAut}(\mathcal{K}_m$ and the permutation $\tau = \nu^\mu$ is under conditions of Theorem 60.

□

Example 79. *Let $\mathcal{K}_5$ be the quaternary Kerdock code of length 16 and type 4^5 with generator matrix*

$$\begin{pmatrix} 1 & 1 & 3 & 0 & 3 & 3 & 0 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 1 & 3 & 0 & 3 & 3 & 0 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 1 & 3 & 0 & 3 & 3 & 0 & 2 & 1 & 2 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 1 & 3 & 0 & 3 & 3 & 0 & 2 & 1 & 2 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 3 & 0 & 3 & 3 & 0 & 2 & 1 & 2 & 1 & 3 \end{pmatrix},$$

where $h(x) = x^4 + 2x^2 + 3x + 1$. *Note that* $\mathcal{I} = \{1, 2, 3, 4, 5\}$ *is a quaternary information set for* $\mathcal{K}_5$. *In this case, we have that* $\lambda = 3$ *and* $\mu = 5$. *Let* $\mathcal{S} = \{\nu^5, \nu^{10}, \nu^{15}\}$, *where* $\nu = (1, \ldots, 15)$. *Note that*

$$\tau = \nu^5 = (1, 6, 11)(2, 7, 12)(3, 8, 13)(4, 9, 14)(5, 10, 15)$$

has 5 disjoint cycles of length 3, where each quaternary information position in $\mathcal{I}$ *is placed in a different cycle of* τ. *Hence,* $S = \Phi(\mathcal{S})$ *is a* 2-PD-*set of size* 3 *for the* $\mathbb{Z}_4$-*linear Kerdock code* K_5 *of length* 32 *with information set* $\Phi(\mathcal{I}) = \{1, \ldots, 10\}$.

Theorem 60 provides the best s-PD-sets of size $s+1$ when the permutation $\tau \in \mathrm{PAut}(\mathcal{C})$ has the minimum number of disjoint cycles $|\mathcal{I}| = \gamma + \delta$, each one being of maximum length. Note that the parameters μ and λ, considered in Corollary 78, denote the number of disjoint cycles and the length of the cycles of the permutation $\tau = \nu^\mu$, respectively. Therefore, this corollary yields the best $(\lambda - 1)$-PD-sets of size λ when $\mu = m$, or equivalently, when $\lambda = f_{\mathcal{K}_m}$. For example, when $m = 5, 7, 11$ or 13. Note that $f_{\mathcal{K}_m} = f_m$.

Table 5.2 shows the maximum values of s for which an s-PD-set of size $s + 1$ for the $\mathbb{Z}_4$-linear Kerdock code K_m of length 2^m can be constructed by using Corollary 78, together with the values of the upper bound f_m shown in Table 2.2.

Prime numbers of type $2^{m-1} - 1$ are known as Mersenne primes and have been extensively studied. It is known that if $m-1$ is not prime, then $2^{m-1}-1$ is not a Mersenne prime. Hence, Corollary 78 can be applied to all nonprime values of $m - 1$. Despite this, there are also some prime values of $m - 1$ (for example, $m-1 = 11$) for which $2^{m-1}-1$ is not a prime number, so Corollary 78 can also be applied. Moreover, even for values of $m - 1$ for which we can not apply this corollary, there are permutations that verify the conditions of Theorem 60, as shown in the following example.

Example 80. *Let* $\mathcal{K}_6$ *be the quaternary Kerdock code of length* 32 *and type* 4^6. *Note that* $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ *is a quaternary information set for* $\mathcal{K}_5$. *The conditions of Corollary 78 are not fulfilled since* 31 *is a Mersenne prime. Nevertheless,*

$$\begin{aligned}\tau = \; & (1, 32, 9, 19, 25)(2, 18, 24, 15, 31)(3, 27, 23, 28, 12) \\ & (4, 8, 20, 30, 26)(5, 14, 16, 21, 13)(6, 10, 17, 29, 22)\end{aligned}$$

satisfies the conditions of Theorem 60 for $s = 4$. *Thus,* $S = \{\Phi(\tau^i)\}_{i=1}^5$ *is a* 4-PD-*set of size* 5 *for the binary Kerdock code* K_5 *of length* 64 *with information set* $\Phi(\mathcal{I})$.

m	s	f_m
5	**2**	2
7	**8**	8
9	**16**	27
10	**6**	50
11	**92**	92
12	**88**	169
13	**314**	314

Table 5.2: Maximum s for which an s-PD-set of size $s+1$ is found for some $\mathbb{Z}_4$-linear Kerdock codes K_m

Corollary 81. *Let $\mathcal{K}_m^-$ be the shortened quaternary Kerdock code of length $2^{m-1}-1$ and type 4^m such that $2^{m-1}-1$ is not a prime number. Let $\nu = (1,\ldots,2^{m-1}-1) \in \mathrm{PAut}(\mathcal{K}_m^-) \subseteq \mathrm{Sym}(2^{m-1}-1)$. Let λ be the greatest divisor of $2^{m-1}-1$ such that $\lambda \leq 2^{m-1}/m$ and μ satisfying that $\lambda\mu = 2^{m-1}-1$. Then $S = \{\Phi(\nu^{i\cdot\mu})\}_{i=1}^{\lambda}$ is a $(\lambda-1)$-PD-set of size λ for $K_m^- = \Phi(\mathcal{K}_m^-)$ with information set $I = \{1,\ldots,2m\}$.*

Chapter 6

MAGMA package implementation and performance analysis

In this chapter, we present the computational part of this dissertation. Most of the concepts and results described in the previous chapters have been implemented as function in the MAGMA Computational Algebra System `http://magma.maths.usyd.edu.au/magma/` [BCFS16]. In Section 6.1, we explain the general context for the functions implemented in this MAGMA software. For quaternary linear codes (also named in this chapter, codes over $\mathbb{Z}_4$) and their corresponding $\mathbb{Z}_4$-linear codes, besides the permutation decoding algorithm described in [BBFV15] or Section 2.6, three more known general decoding methods suitable for these codes have been implemented. For linear codes over finite fields, only the permutation decoding algorithm and related functions have been developed. In Section 6.2, we briefly review the three general decoding methods implemented for quaternary linear codes. In section 6.3, we provide a comparison of the performance of these four methods, including the permutation decoding method, when they are applied to $\mathbb{Z}_4$-linear Hadamard and Kerdock codes. In Section 6.4, we include the manual describing all implemented functions for codes over $\mathbb{Z}_4$. Most of them have been used in the performance analysis presented in Section 6.3. Finally, in Section 6.5, the manual of the implemented functions for linear codes over finite fields is also included.

6.1 MAGMA package implementation

MAGMA is a software system designed to solve computationally hard problems in algebra, number theory, geometry and combinatorics. In general,

MAGMA supports basic facilities for linear codes over finite fields [BCFS16, Chapter 158] and linear codes over integer residue rings and Galois rings [BCFS16, Chapter 161], including additional functionality for the special case of codes over $\mathbb{Z}_4$ [BCFS16, Chapter 162]. The members of the research group *Combinatoric, Coding and Security Group* (CCSG) have been working on extending the MAGMA system implementing new packages to work efficiently with $\mathbb{Z}_4$-linear codes [PPV12], $\mathbb{Z}_2\mathbb{Z}_4$-linear codes [BFP$^+$12], and nonlinear codes in general [PV14]. More specifically, version 1.4 of the package that expands the functionality for $\mathbb{Z}_4$-linear codes (equivalently, for codes over $\mathbb{Z}_4$) includes constructions for some families of codes over $\mathbb{Z}_4$; efficient functions for computing the rank and dimension of the kernel of any code over $\mathbb{Z}_4$; general functions for computing coset representatives for a subcode in a code over $\mathbb{Z}_4$; as well as functions for computing the permutation automorphism group for Hadamard and extended perfect codes over $\mathbb{Z}_4$, and their orders.

As part of this dissertation, new functionality related to the information space, information sets, syndrome space and coset leaders for codes over $\mathbb{Z}_4$ has been included in MAGMA. For example, now there are functions to obtain an information set and to check whether or not a set or coordinate position is an information set for a code over $\mathbb{Z}_4$. We have also implemented functions to compute the syndrome of a given vector and the coset leaders (vectors of minimal Lee weight in their cosets) for a code over $\mathbb{Z}_4$.

In the MAGMA default distribution (up to version V2.21), there was not any function available for decoding codes over $\mathbb{Z}_4$. In this sense, functions to decode by using different methods have been included in the extension package for these codes. Specifically, the syndrome, lifted, and coset decoding algorithms have been implemented. These decoding methods are briefly described in Section 6.2. For the permutation decoding, besides functions to simulate the decoding process by using this method, functions to check whether or not a set of permutations is an s-PD-set for a code with respect to an information set have been developed. Moreover, functions for constructing most of the s-PD-sets described in Chapters 4 and 5 for $\mathbb{Z}_4$-linear Hadamard and Kerdock codes have also been implemented.

The last version 2.0 of the package for $\mathbb{Z}_4$-linear codes together with a manual describing all functions can be downloaded from the CCSG web site `http://ccsg.uab.cat` [BPPV16]. It is also worth mentioning that MAGMA version V2.22 (from May 2016) and later contains this package by default [BCFS16, Chapter 162], so it is no longer necessary to install it. For the convenience of the reader, the detailed description of the implemented functions

for codes over $\mathbb{Z}_4$ can be also found in Section 6.4, where several examples showing how to use these functions are also included.

Unlike for codes over $\mathbb{Z}_4$, for linear codes over finite fields, the MAGMA default distribution already contained a function to decode. The implemented methods through this function are the syndrome and Euclidean decoding. If the code is alternant (BCH, Goppa, and Reed–Solomon codes, etc.), then the Euclidean algorithm is applied, and otherwise the syndrome decoding is performed. We have also extended the functionality for codes over finite fields, adding functions to simulate the permutation decoding method. There is also a new function to check whether or not a set of permutations is an s-PD-set for a linear code with respect to an information set. Finally, functions to construct the s-PD-sets given in Chapter 3 for the family of binary linear Hadamard codes and in [FKM12] for simplex codes over any finite field have also been implemented.

The above mentioned functions can be also downloaded from CCSG web site `http://ccsg.uab.cat` [BPPV16], and are also included in MAGMA version V2.22 (from May 2016) and later [BCFS16, Chapter 158]. Again, for the convenience of the reader, the detailed description of the implemented functions for codes over finite fields can also be found in Section 6.5. Along with the description of the functions, several examples showing how to use these functions are included.

In CCSG web site `http://ccsg.uab.cat`, apart from the files with the implemented functions and the manual, files to test the correctness of the implemented functions are included. Figure 6.1 shows the directory structure and files of this package. Note that all examples provided with the manual can also be found written in a text file, so they can be uploaded directly to MAGMA.

6.2 Implemented decoding methods

The hardest problem in the process of transmitting information is decoding. For linear codes, a general decoding algorithm is the *syndrome decoding* [MS77, Chapter 1]. For linear codes over Galois rings, there is also a general decoding method [GV98, BZ01], which will be referred as *lifted decoding*. For codes in general (not necessarily linear), an algorithm to decode based on the kernel and coset representation was proposed in [VZP15] and is called *coset decoding*. Finally, for some families of $\mathbb{Z}_4$-linear codes for which we know the

The "Codes Over Z4" package is composed of five directories:

/src: *The files to attach to Magma:*
- CodesOverZ4.m
- DecodeOverZ4.m
- PermDecodeOverFq.m

/license: *The license of the package.*
/doc: *The manual to use the package in pdf format.*
/test: *Several files to test the package.*
- CodesOverZ4_R_test.m
- testCodesOverZ4.m
- testDecodeOverZ4_part1.m
- testDecodeOverZ4_part2.m
- testPermDecodeOverFq

/examples: *Examples from the manual.*
- H2E1, H2E2, ..., H2E16
- H3

Figure 6.1: Directory structure and files for "Codes over Z4" package

existence of PD-sets, a permutation decoding algorithm can also be applied [BBFV15, BPV16, BV16a, BV16b].

This section briefly describes the first three algorithms mentioned above for decoding vectors from the ambient space over $\mathbb{Z}_4$, or the binary space under the Gray; that is, syndrome, lifted, and coset decoding. Permutation decoding is reviewed in Section 2.6 of this dissertation. These four decoding algorithms for codes over $\mathbb{Z}_4$ are the ones that have been implemented in MAGMA. The description of the functions can be found in Section 6.4, together with several examples showing how to use them.

Let $\mathcal{C}$ be a quaternary linear code of length n, type $2^\gamma 4^\delta$, and minimum Lee distance d.

6.2.1 Syndrome decoding

The general decoding algorithm for linear codes over finite fields, called syndrome decoding [MS77, Chapter1], can also be applied to quaternary linear codes.

The syndrome of a vector $u \in \mathbb{Z}_4^n$ with respect to a parity check matrix $\mathcal{H}$ of $\mathcal{C}$ is the vector $s = \mathcal{H}u \in \{0,2\}^\gamma \times \mathbb{Z}_4^{n-\gamma-\delta}$. We consider $\mathcal{H}$ as a parity

check matrix containing the minimum number of rows, where the first γ rows are of order two and the last $n-\gamma-\delta$ rows are of order four. Note that $\mathcal{C}$ consists of all vectors whose syndrome is equal to the zero vector. Moreover, every vector in $\{0,2\}^{\gamma} \times \mathbb{Z}_4^{n-\gamma-\delta}$ is a syndrome and there is a one-to-one correspondence between cosets of $\mathcal{C}$ and its syndromes. Let $\mathcal{C}_s$ be the coset of $\mathcal{C}$ consisting of all vectors in $\mathbb{Z}_4^n$ having syndrome s.

The syndrome decoding algorithm consists of computing a table pairing each possible syndrome $s \in \{0,2\}^{\gamma} \times \mathbb{Z}_4^{n-\gamma-\delta}$ with an error vector of minimum Lee weight e_s, called coset leader, in the coset $\mathcal{C}_s$. After receiving a vector $u \in \mathbb{Z}_4^n$, compute its syndrome $s = \mathcal{H}u$. Then, u is decoded as the codeword $c = u - e_s$.

Although creating the syndrome table is a one-time task, which is carried out before decoding the received vectors, sometimes it can be difficult to create and store it. Moreover, if it contains many elements, it can also be difficult to find the corresponding error vector from a given syndrome.

6.2.2 Lifted decoding

A general decoding algorithm for linear codes over Galois rings was presented in [BZ01]. We refer to this method as lifted decoding, since it is based on lifting decoding algorithms for a family of linear codes over a finite field. This method can be applied to quaternary linear codes, which maybe nonfree $\mathbb{Z}_4$-modules.

More specifically, this method comprises lifting decoding algorithms for two binary linear codes C_0 and C_1, such that $C_0 \subseteq C_1$, constructed from a generator matrix of $\mathcal{C}$. The binary codes C_0 and C_1 are known as the residue and torsion codes of $\mathcal{C}$. Let t_0 and t_1 be the error-correcting capability of C_0 and C_1, respectively. Assume the received vector $u = c + e$, where $c \in \mathcal{C}$ and $e \in \mathbb{Z}_4^n$ is the error vector. Then, all error vectors e such that $\tau_1 + \tau_3 \leq t_0$ and $\tau_2 + \tau_3 \leq t_1$, where τ_i is the number of occurrences of i in e, can be corrected.

For the binary linear codes C_0 and C_1, the general syndrome decoding method for codes over finite fields can be applied. However, in case we have more information about these codes, other more efficient methods can be used, as the Euclidean algorithm if they are known to be alternant codes (for example, BCH, Goppa, or Reed-Solomon codes) [MS77, Chapter 12].

6.2.3 Coset decoding

A general decoding algorithm for codes over finite fields (not necessarily linear) was introduced in [VZP15]. This method, called coset decoding, is based on representing the code as the union of cosets of a linear subcode called kernel [FPV08, FPV10] and computing the minimum Hamming weight for a family of linear codes.

The $\mathbb{Z}_4$-linear code $C = \Phi(\mathcal{C})$ is a binary code, which can be represented as

$$C = \bigcup_{i=0}^{t} (K + c_i),$$

where $K = \{x \in C \ : \ x + C = C\}$ is a binary linear subcode, $c_0, c_1, \ldots, c_t$ are coset representatives of C with respect to K (not necessarily of minimal Hamming weight in their cosets) and c_0 is the zero codeword. Note that if C is linear, then $C = K$. Moreover, note that the minimum Lee distance d of $\mathcal{C}$ coincides with the minimum Hamming weight of $C = \Phi(\mathcal{C})$.

After receiving a vector $u \in \mathbb{Z}_4^n$, the coset decoding algorithm considers the binary linear codes

$$K_i = K \cup (K + c_i + \Phi(u)),$$

$i \in \{0, \ldots, t\}$. If the minimum Hamming weight of $\cup_{i=0}^{t} K_i$ is less than the minimum Hamming weight of K, then u is decoded as the codeword c such that $\Phi(c) = \Phi(u) + e$, where e is a word of minimum Hamming weight of $\cup_{i=0}^{t} K_i$.

This decoding method is based on computing the minimum Hamming weight of $t+1$ linear codes K_i, $i \in \{0, \ldots, t\}$. Although it is known that the problem of computing the minimum Hamming weight for linear codes is NP-hard [Var97], the Brouwer-Zimmermann algorithm, implemented in MAGMA, can be used [Whi06, Zim96]. Then, for linear codes with a big codimension or nonlinear codes with a big kernel, this method can have better performance than syndrome decoding.

6.3 Performance analysis

This section aims to provide a comparison study of the performance of the three decoding method described in Section 6.2 and the permutation decoding method reviewed in Section 2.6, when they are applied to the (nonlinear)

$\mathbb{Z}_4$-linear Hadamard codes $H_{0,\delta}$ of length $2^{2\delta-1}$, $\delta \geq 3$, and the $\mathbb{Z}_4$-linear Kerdock codes K_m of length 2^m, $m \geq 5$.

In order to better compare the different decoding methods, we will focus only on a partial decoding by using the s-PD-sets of size $s+1$ found in the previous chapters. Specifically, we use the $f_{0,\delta}$-PD-set $\Phi(\mathcal{P}_{0,\delta})$ of size $f_{0,\delta}+1$ given by Theorem 50 for the $\mathbb{Z}_4$-linear Hadamard code $H_{0,\delta}$. Recall that $f_{0,\delta}$ is the greater s for which we can find s-PD-sets of size $s+1$ for $H_{0,\delta}$. For the $\mathbb{Z}_4$-linear Kerdock code K_m, the $(\lambda-1)$-PD-set S of size λ, where λ is the greatest divisor of $2^{m-1}-1$ such that $\lambda \leq 2^{m-1}/m$, obtained in Corollary 78 is employed. The time spent generating those sets is included in all tests.

Example 82. *Let $H_{0,3}$ be the (nonlinear) $\mathbb{Z}_4$-linear Hadamard code of length 32 and type $2^0 4^3$. Figure 6.2 displays the time in seconds to decode random received vectors with at most $f_{0,3} = 4$ errors by using permutation, syndrome, lifted, and coset decoding. Despite lifted decoding has a similar performance (in terms of time) than permutation decoding, it is the unique method that cannot correct all received vectors since $t_0 = t_1 = 3$. We observe a negligible amount of time generating the 4-PD-set obtained by Theorem 50, fact that facilitates permutation decoding to be the best method (both in terms of time and correctly decoded received vectors) when trying to correct up to 4 errors.*

For the $\mathbb{Z}_4$-linear Kerdock code K_5 of length 32 we obtain similar results. In this example, syndrome, lifted and permutation decoding behaves similarly while coset decoding is the method with the worst performance. Figure 6.3 summarizes these results, displaying the time in seconds to decode random received vectors with at most 2 errors, since a 2-PD-set of size 3 is used for the partial permutation decoding.

Although creating the syndrome table is a one-time task, which is carried out before decoding the received vectors, sometimes it can be difficult to create and store it. Moreover, if it contains many elements, it can be hard to find the corresponding error vector from a given syndrome. Thus, syndrome and lifted decoding are not suitable for decoding neither $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ with $\delta \geq 4$ nor Kerdock codes K_m with $m \geq 7$. Note that to correct up to s errors, the total number of syndromes is $2(\sum_{i=0}^{s} \binom{\beta}{i})$ for lifted decoding and $\sum_{i=0}^{s} \binom{2\beta}{i}$ for syndrome decoding.

Example 83. *Syndrome and lifted decoding become useless when trying to decode the $\mathbb{Z}_4$-linear Hadamard code $H_{0,4}$ and the $\mathbb{Z}_4$-linear Kerdock code K_7 both of length 128, since they have too many cosets. Note that the same problem appears for any $\mathbb{Z}_4$-linear Hadamard and Kerdock code of length 2^m*

δ	m	**Coset decoding**	**Permutation decoding**
3	5	153.08	9.85
4	7	26.53	19.13
5	9	101.75	64.92
6	11	628.43	243.97
7	13	5879.25	1079.62

Table 6.1: Time for decoding using $H_{0,\delta}$ of length 2^m, where $m = 2\delta - 1$.

m	**Coset decoding**	**Permutation decoding**
5	284.34	10.71
7	1473.69	23.03
9	44634.00	62.51
10	9669.70	93.13
11	61215.49	259.18

Table 6.2: Time for decoding using K_m of length 2^m.

with $m > 7$. Figures 6.4 and 6.5 display the time in seconds to decode random received vectors with at most $f_{0,4} = 15$ errors for $H_{0,4}$ and with at most 8 errors for K_7.

Example 84. *Let $H_{0,\delta}$ be the $\mathbb{Z}_4$-linear Hadamard code of length $2^{2\delta-1}$ and type $2^0 4^\delta$. Let K_m be the $\mathbb{Z}_4$-linear Kerdock code of length 2^m and type 4^m. Table 6.1 shows the time in seconds to decode 100,000 random received vectors with at most $s = f_{0,\delta}$ errors for $H_{0,\delta}$ of length $2^{2\delta-1}$ by using permutation and coset decoding for $\delta \in \{3, \ldots, 7\}$.*

Similarly, Table 6.2 shows the time in seconds to decode 100,000 random received vectors with at most $s = \lambda - 1$ errors for K_m by using permutation and coset decoding for $m \in \{5, 7, 9, 10, 11\}$. Permutation decoding has better performance than coset decoding for each $H_{0,\delta}$ with $\delta \geq 3$ and K_m with $m \geq 5$. The content of Tables 6.1 and 6.2 is partially included in Figures 6.6 and 6.7, respectively.

All tests have been performed in MAGMA version V2.21-4, running on a server with an Intel Xeon processor (clock speed 3.30GHz).

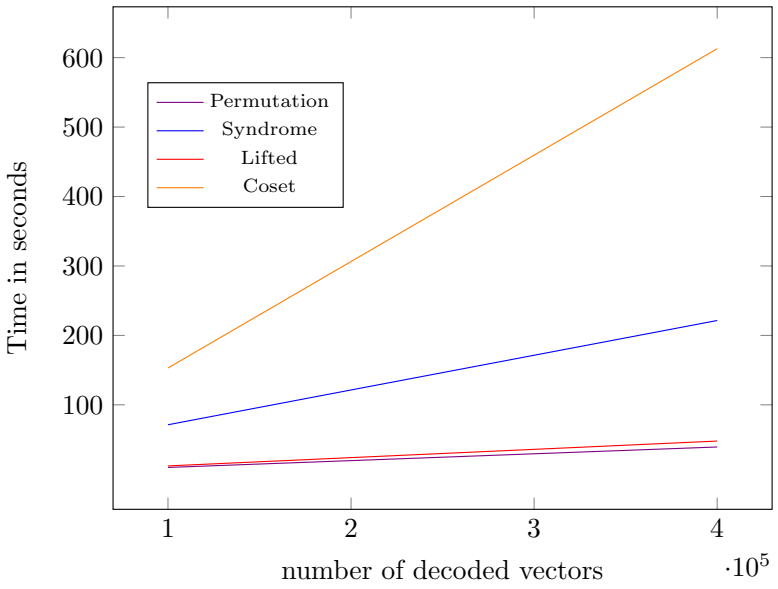

Figure 6.2: Time for decoding using the $\mathbb{Z}_4$-linear Hadamard code $H_{0,3}$

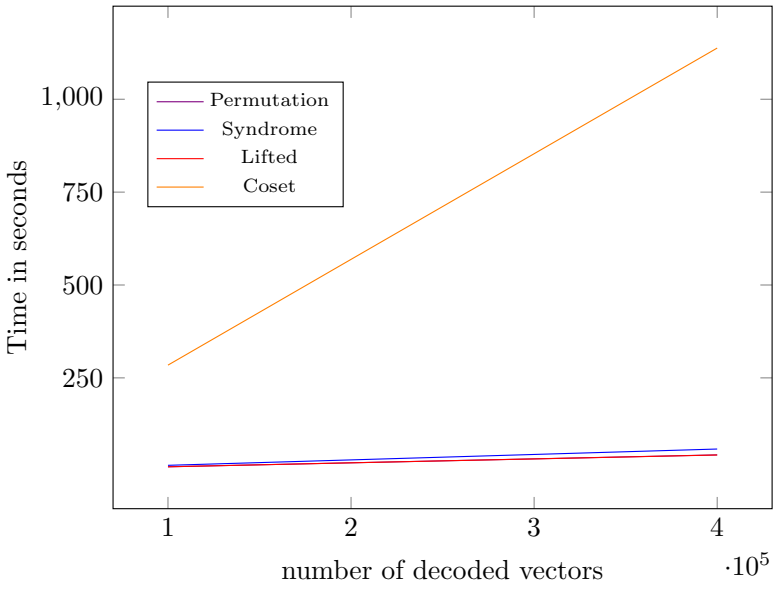

Figure 6.3: Time for decoding using the $\mathbb{Z}_4$-linear Kerdock code K_5

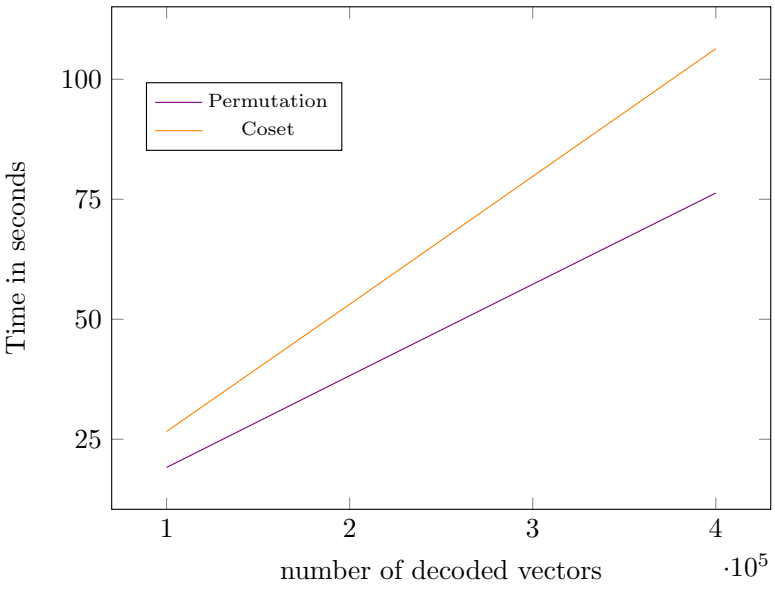

Figure 6.4: Time for decoding using the $\mathbb{Z}_4$-linear Hadamard code $H_{0,4}$

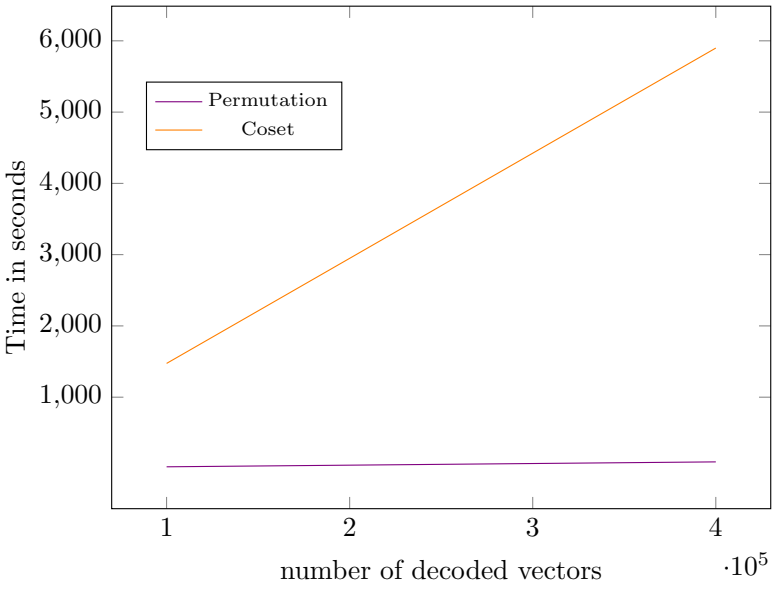

Figure 6.5: Time for decoding using the $\mathbb{Z}_4$-linear Kerdock code K_7

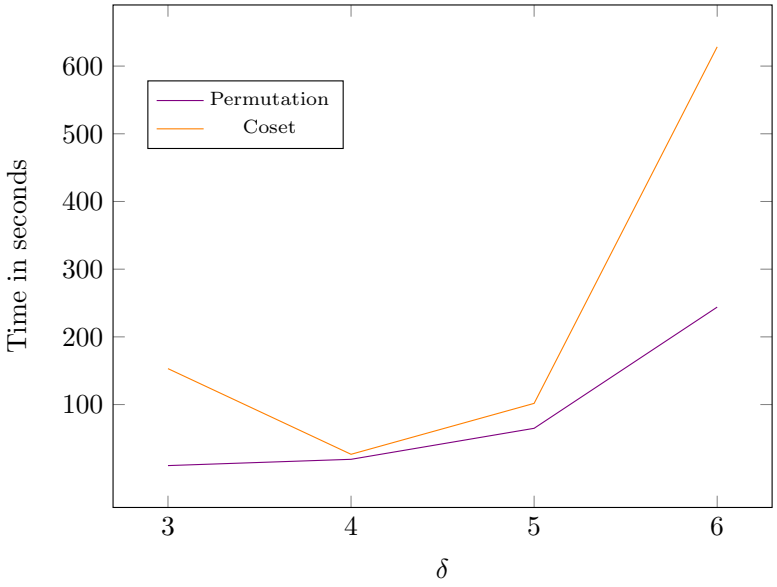

Figure 6.6: Time for decoding using the $\mathbb{Z}_4$-linear Hadamard code $H_{0,\delta}$

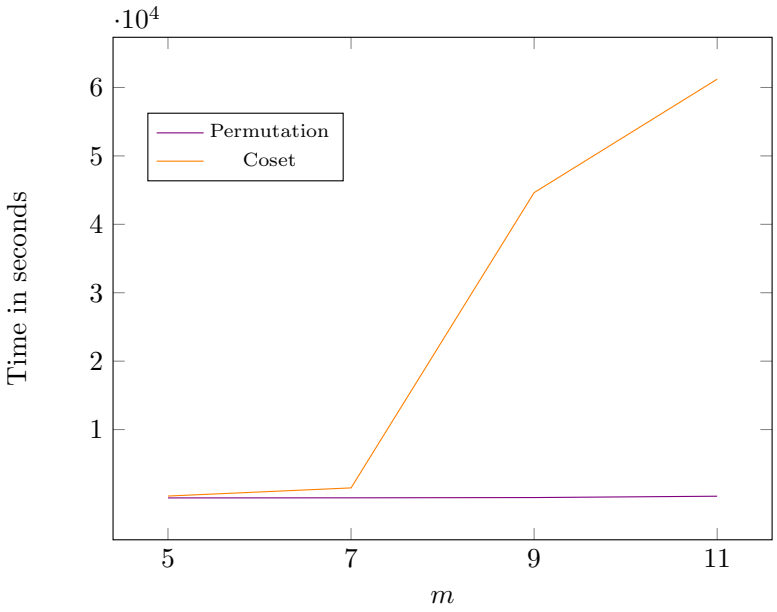

Figure 6.7: Time for decoding using the $\mathbb{Z}_4$-linear Kerdock code K_m

6.4 MAGMA functions for codes over $\mathbb{Z}_4$

This section supplies functions for decoding vectors from the ambient space of a code over $\mathbb{Z}_4$, or the corresponding space over $\mathbb{Z}_2$ under the Gray map, using four different algorithms: coset decoding, syndrome decoding, lifted decoding and permutation decoding. For the permutation decoding algorithm, if the parameter s is less than the error-correcting capability of the code, then a partial permutation decoding is performed. A function to know whether or not a set of permutations S is an s-PD-set for a code over $\mathbb{Z}_4$ with respect to an information set I is included. Moreover, functions for constructing most of the s-PD-sets described in Chapters 4 and 5 for $\mathbb{Z}_4$-linear Hadamard and Kerdock codes are also provided. Finally, functions related to information space, information sets, syndrome space and coset leaders for codes over $\mathbb{Z}_4$ can also be found. The section also provides detailed examples for all the implemented functions, in order to see better how they work.

6.4.1 Coset decoding

`CosetDecode(C, u : parameters)`

`MinWeightCode`	RNGINTELT	*Default* : -
`MinWeightKernel`	RNGINTELT	*Default* : -

Given a code C over $\mathbb{Z}_4$ of length n, and a vector u from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$, attempt to decode u with respect to C. If the decoding algorithm succeeds in computing a vector $u' \in C$ as the decoded version of $u \in V$, then the function returns `true`, u' and $\Phi(u')$, where Φ is the Gray map. If the decoding algorithm does not succeed in decoding u, then the function returns `false`, the zero vector in V and the zero vector in V_2.

The coset decoding algorithm considers the binary linear code $C_u = C_{bin} \cup (C_{bin} + \Phi(u))$, when $C_{bin} = \Phi(C)$ is linear. On the other hand, when C_{bin} is nonlinear, we have $C_{bin} = \bigcup_{i=0}^{t}(K_{bin} + \Phi(c_i))$, where $K_{bin} = \Phi(K_C)$, K_C is the kernel of C as a subcode over $\mathbb{Z}_4$, $[c_0, c_1, \ldots, c_t]$ are the coset representatives of C with respect to K_C (not necessarily of minimal weight in their cosets) and c_0 is the zero codeword. In this case, the algorithm considers the binary linear codes $K_0 = K_{bin} \cup (K_{bin} + \Phi(u))$, $K_1 = K_{bin} \cup (K_{bin} + \Phi(c_1) + \Phi(u))$, $\ldots$, $K_t = K_{bin} \cup (K_{bin} + \Phi(c_t) + \Phi(u))$.

If the parameter `MinWeightCode` is not assigned, then the minimum weight of C, which coincides with the minimum weight of C_{bin}, denoted by d, is computed. Note that the minimum distance of C_{bin} coincides with its minimum

weight. If C_{bin} is linear and the minimum weight of C_u is less than d, then $\Phi(u') = \Phi(u) + e$, where e is a word of minimum weight of C_u; otherwise, the decoding algorithm returns `false`. On the other hand, if C_{bin} is non-linear and the minimum weight of $\cup_{i=0}^{t} K_i$ is less than the minimum weight of K_{bin}, then $\Phi(u') = \Phi(u) + e$, where e is a word of minimum weight of $\cup_{i=0}^{t} K_i$; otherwise, the decoding algorithm returns `false`. If the parameter `MinWeightKernel` is not assigned, then the minimum Hamming weight of K_{bin} is computed.

`CosetDecode(C, Q : parameters)`

`MinWeightCode`	RNGINTELT	*Default* : -
`MinWeightKernel`	RNGINTELT	*Default* : -

Given a code C over $\mathbb{Z}_4$ of length n, and a sequence Q of vectors from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$, attempt to decode the vectors of Q with respect to C. This function is similar to the function `CosetDecode(C, u)` except that rather than decoding a single vector, it decodes a sequence of vectors and returns a sequence of booleans and two sequences of decoded vectors corresponding to the given sequence. The algorithm used and effect of the parameters `MinWeightCode` and `MinWeightKernel` are identical to those for the function `CosetDecode(C, u)`.

Example 85. *Starting with the Hadamard code C over $\mathbb{Z}_4$ of length* 16 *and type $2^0 4^3$, a codeword $c \in C$ is selected and then perturbed to give a vector u in the ambient space of C. The vector u is then decoded to recover c.*

```
> C := HadamardCodeZ4(3, 5);
> C;
((16, 4^3 2^0)) Linear Code over IntegerRing(4)
Generator matrix:
[1 0 3 2 0 3 2 1 3 2 1 0 2 1 0 3]
[0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3]
[0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3]
> d := MinimumLeeDistance(C);
> t := Floor((d-1)/2);
> t;
7

> c := C ! [1,1,1,1,2,2,2,2,3,3,3,3,0,0,0,0];
> c in C;
true
> u := c;
> u[5] := u[5] + 2;
```

```
> u[12] := u[12] + 1;
> u[13] := u[13] + 3;
> u[16] := u[16] + 2;
> c;
(1 1 1 1 2 2 2 2 3 3 3 3 0 0 0 0)
> u;
(1 1 1 1 0 2 2 2 3 3 3 0 3 0 0 2)
> grayMap := GrayMap(UniverseCode(Integers(4), Length(C)));
> grayMap(c-u);
(0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1)

> isDecoded, uDecoded := CosetDecode(C, u : MinWeightCode := d);
> isDecoded;
true
> uDecoded eq c;
true
```

6.4.2 Syndrome decoding

`SyndromeDecode(C, u)`

Given a code C over $\mathbb{Z}_4$ of length n, and a vector u from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$, attempt to decode u with respect to C. The decoding algorithm always succeeds in computing a vector $u' \in C$ as the decoded version of $u \in V$, and the function returns `true`, u' and $\Phi(u')$, where Φ is the Gray map. Although the function never returns `false`, the first output parameter `true` is given to be consistent with the other decoding functions.

The syndrome decoding algorithm consists of computing a table pairing each possible syndrome s with a vector of minimum Lee weight e_s, called coset leader, in the coset of C containing all vectors having syndrome s. After receiving a vector u, its syndrome s is computed using the parity check matrix. Then, u is decoded into the codeword $c = u - e_s$.

`SyndromeDecode(C, Q)`

Given a code C over $\mathbb{Z}_4$ of length n, and a sequence Q of vectors from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$, attempt to decode the vectors of Q with respect to C. This function is similar to the function `SyndromeDecode(C, u)` except that rather than decoding a single vector, it decodes a sequence of vectors and returns a sequence of booleans and two sequences of decoded vectors corresponding to the given sequence. The algorithm used is the same as that of function `SyndromeDecode(C, u)`.

Example 86. *The Hadamard code C over $\mathbb{Z}_4$ of length 8 and type $2^1 4^2$ is constructed. Next, information bits are encoded using C and three errors are introduced to give the vector u. Then u is decoded by calculating its syndrome and applying the map, given by the* `CosetLeaders` *function, to the syndrome to recover the original vector.*

```
> C := HadamardCodeZ4(2, 4);
> C;
((8, 4^2 2^1, 8)) Linear Code over IntegerRing(4)
Generator matrix:
[1 0 3 2 1 0 3 2]
[0 1 2 3 0 1 2 3]
[0 0 0 0 2 2 2 2]
> t := Floor((MinimumLeeDistance(C)-1)/2);
> t;
3

> R, V, f, fbin := InformationSpace(C);
> i := R![2,1,0];
> c := f(i);
> c;
(1 0 3 2 3 2 1 0)
> u := c;
> u[5] := u[5] + 3;
> u[7] := u[7] + 2;
> c;
(1 0 3 2 3 2 1 0)
> u;
(1 0 3 2 2 2 3 0)
> grayMap := GrayMap(UniverseCode(Integers(4), Length(C)));
> grayMap(c-u);
(0 0 0 0 0 0 0 0 1 0 0 1 1 0 0)

> isDecoded, uDecoded := SyndromeDecode(C, u);
> isDecoded;
true
> uDecoded eq c;
true

> L, mapCosetLeaders := CosetLeaders(C);
> errorVector := mapCosetLeaders(Syndrome(u, C));
> errorVector;
(0 0 0 0 3 0 2 0)
> u-errorVector eq c;
true
```

6.4.3 Lifted decoding

`LiftedDecode(C, u : parameters)`

`AlgMethod` MONSTGELT *Default* : "Euclidean"

Given a code C over $\mathbb{Z}_4$ of length n, and a vector u from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$, attempt to decode u with respect to C. If the decoding algorithm succeeds in computing a vector $u' \in C$ as the decoded version of $u \in V$, then the function returns `true`, u' and $\Phi(u')$, where Φ is the Gray map. If the decoding algorithm does not succeed in decoding u, then the function returns `false`, the zero vector in V and the zero vector in V_2 (in the Euclidean case it may happen that u' is not in C because there are too many errors in u to correct). The lifted decoding algorithm comprises lifting decoding algorithms for two binary linear codes C_0 and C_1, being the residue and torsion codes of C. Let t_0 and t_1 be the error-correcting capability of C_0 and C_1, respectively. Assume the received vector $u = c + e$, where $c \in C$ and $e \in V$ is the error vector. Then, the lifted decoding algorithm can correct all error vectors e such that $\tau_1 + \tau_3 \leq t_0$ and $\tau_2 + \tau_3 \leq t_1$, where τ_i is the number of occurrences of i in e.

In the decoding process, the function `Decode(C, u)` for linear codes is used. The accessible algorithms for linear codes are: syndrome decoding and a Euclidean algorithm, which operates on alternant codes (BCH, Goppa, and Reed–Solomon codes, etc.). If C_0 or C_1 is alternant, the Euclidean algorithm is used by default, but the syndrome algorithm will be used if the parameter `AlgMethod` is assigned the value `"Syndrome"`. For non-alternant codes C_0 and C_1, only syndrome decoding is possible, so the parameter `AlgMethod` is not relevant.

`LiftedDecode(C, Q : parameters)`

`AlgMethod` MONSTGELT *Default* : "Euclidean"

Given a code C over $\mathbb{Z}_4$ of length n, and a sequence Q of vectors from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$, attempt to decode the vectors of Q with respect to C. This function is similar to the function `LiftedDecode(C, u)` except that rather than decoding a single vector, it decodes a sequence of vectors and returns a sequence of booleans and two sequences of decoded vectors corresponding to the given sequence. The algorithm used and effect of the parameter `AlgMethod` are the same as for `LiftedDecode(C, u)`.

Example 87. *The Hadamard code C over $\mathbb{Z}_4$ of length 8 and type $2^1 4^2$ is constructed. Then an information word is encoded using C, three errors are*

introduced into the codeword c, and then c is recovered by using the lifted decoding algorithm.

```
> C := HadamardCodeZ4(2, 4);
> C;
((8, 4^2 2^1, 8)) Linear Code over IntegerRing(4)
Generator matrix:
[1 0 3 2 1 0 3 2]
[0 1 2 3 0 1 2 3]
[0 0 0 0 2 2 2 2]
> d := MinimumLeeDistance(C);
> t := Floor((d-1)/2);
> t;
3
> C0 := BinaryResidueCode(C);
> C1 := BinaryTorsionCode(C);
> t0 := Floor((MinimumDistance(C0)-1)/2);
> t1 := Floor((MinimumDistance(C1)-1)/2);
> t0, t1;
1 1
```

Using the lifted decoding, it is possible to correct all error vectors e such that $\tau_1 + \tau_3 \leq t_0 = 1$ *and* $\tau_2 + \tau_3 \leq t_1 = 1$*, where* τ_i *is the number of occurrences of i in e. The following statements show that it is not possible to correct the error vector* $e = (00003020)$ *since* $\tau_2 + \tau_3 = 2 > 1$*, but it is possible to correct the error vector* $e = (00001020)$ *since* $\tau_1 + \tau_3 = 1 \leq 1$ *and* $\tau_2 + \tau_3 = 1 \leq 1$*.*

```
> R, V, f, fbin := InformationSpace(C);
> i := R![2,1,0];
> c := f(i);
> c;
(1 0 3 2 3 2 1 0)

> u := c;
> u[5] := u[5] + 3;
> u[7] := u[7] + 2;
> c;
(1 0 3 2 3 2 1 0)
> u;
(1 0 3 2 2 2 3 0)
> e := u - c;
> e;
(0 0 0 0 3 0 2 0)

> isDecoded, uDecoded := LiftedDecode(C, u);
> isDecoded;
```

```
true
> uDecoded eq c;
false

> u := c;
> u[5] := u[5] + 1;
> u[7] := u[7] + 2;
> c;
(1 0 3 2 3 2 1 0)
> u;
(1 0 3 2 0 2 3 0)
> e := u - c;
> e;
(0 0 0 0 1 0 2 0)

> isDecoded, uDecoded := LiftedDecode(C, u);
> isDecoded;
true
> uDecoded eq c;
true
```

6.4.4 Permutation decoding

Let C be a code over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$ and $C_{bin} = \Phi(C)$, where Φ is the Gray map. A subset $S \subseteq \mathrm{Sym}(2n)$ is an s-PD-set for C_{bin} with respect to a subset of coordinate positions $I \subseteq \{1, \ldots, 2n\}$ if S is a subset of the permutation automorphism group of C_{bin}, I is an information set for C_{bin}, and every s-set of coordinate positions in $\{1, \ldots, 2n\}$ is moved out of the information set I by at least one element of S, where $1 \leq s \leq t$ and t is the error-correcting capability of C_{bin}.

If $I = [i_1, \ldots, i_{\gamma+\delta}] \subseteq \{1, \ldots, n\}$ is an information set for C such that the code obtained by puncturing C at positions $\{1, \ldots, n\} \backslash \{i_{\gamma+1}, \ldots, i_{\gamma+\delta}\}$ is of type 4^δ, then $\Phi(I) = [2i_1 - 1, \ldots, 2i_\gamma - 1, 2i_{\gamma+1} - 1, 2i_{\gamma+1}, \ldots, 2i_{\gamma+\delta} - 1, 2i_{\gamma+\delta}]$ is an information set for C_{bin}. It is also easy to see that if S is a subset of the permutation automorphism group of C, that is, $S \subseteq \mathrm{PAut}(C) \subseteq \mathrm{Sym}(n)$, then $\Phi(S) = [\Phi(\tau) : \tau \in S] \subseteq \mathrm{PAut}(C_{bin}) \subseteq \mathrm{Sym}(2n)$, where

$$\Phi(\tau)(i) = \begin{cases} 2\tau(i/2), & \text{if } i \text{ is even,} \\ 2\tau((i+1)/2) - 1 & \text{if } i \text{ is odd.} \end{cases} \tag{6.1}$$

Given a subset of coordinate positions $I \subseteq \{1, \ldots, n\}$ and a subset $S \subseteq \mathrm{Sym}(n)$, in order to check that $\Phi(S)$ is an s-PD-set for C_{bin} with respect to $\Phi(I)$, it is enough to check that S is a subset of the permutation

automorphism group of C, I is an information set for C, and every s-set of coordinate positions in $\{1, \ldots, n\}$ is moved out of the information set I by at least one element of S [BV16a, BV16b].

`IsPermutationDecodeSet(C, I, S, s)`

Given a code C over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$, a sequence $I \subseteq \{1, \ldots, 2n\}$, a sequence S of elements in the symmetric group $\mathrm{Sym}(2n)$ of permutations on the set $\{1, \ldots, 2n\}$, and an integer $s \geq 1$, return `true` if and only if S is an s-PD-set for $C_{bin} = \Phi(C)$, where Φ is the Gray map, with respect to the information set I.

The arguments I and S can also be given as a sequence $I \subseteq \{1, \ldots, n\}$ and a sequence S of elements in the symmetric group $\mathrm{Sym}(n)$ of permutations on the set $\{1, \ldots, n\}$, respectively. In this case, the function returns `true` if and only if $\Phi(S)$ is an s-PD-set for $C_{bin} = \Phi(C)$ with respect to the information set $\Phi(I)$, where $\Phi(I)$ and $\Phi(S)$ are the sequences defined as above.
Depending on the length of the code C, its type, and the integer s, this function could take some time to compute whether S or $\Phi(S)$ is an s-PD-set for C_{bin} with respect to I or $\Phi(I)$, respectively. Specifically, if the function returns `true`, it is necessary to check $\sum_{i=1}^{s} \binom{|I|}{i} \cdot \binom{N-|I|}{s-i}$ s-sets, where $N = n$ and $|I| = \gamma + \delta$ when I is given as an information set for C, or $N = 2n$ and $|I| = \gamma + 2\delta$ when I is given as an information set for C_{bin}.

The verbose flag `IsPDSetFlag` is set to level 0 by default. If it is set to level 1, the total time used to check the condition is shown. Moreover, the reason why the function return `false` is also shown, that is, whether I is not an information set, S is not a subset of the permutation automorphism group or S is not an s-PD-set. If it is set to level 2, the percentage of the computation process performed is also printed.

`PermutationDecode(C, I, S, s, u)`

The arguments for the intrinsic are as follows:

- C is a code over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$;

- $I = [i_1, \ldots, i_{\gamma+\delta}] \subseteq \{1, \ldots, n\}$ is an information set for C given as a sequence of coordinate positions such that the code C obtained by puncturing C at coordinate positions $\{1, \ldots, n\} \backslash \{i_{\gamma+1}, \ldots, i_{\gamma+\delta}\}$ is of type 4^δ;

- S is a sequence S such that either S or $\Phi(S)$ is an s-PD-set for $C_{bin} = \Phi(C)$, where Φ is the Gray map, with respect to $\Phi(I)$;

- s is an integer such that $s \in \{1, \ldots, t\}$, where t is the error-correcting capability of C_{bin};

- u is a vector from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$,

Given the above assumptions, the function attempts to decode u with respect to C. If the decoding algorithm succeeds in computing a vector $u' \in C$ as the decoded version of $u \in V$, then the function returns the values `true`, u' and $\Phi(u')$. If the decoding algorithm does not succeed in decoding u, then the function returns the values `false`, the zero vector in V and the zero vector in V_2.

Assume that the received vector $\Phi(u) = c + e$, where $u \in V$, $c \in C_{bin}$ and $e \in V_2$ is the error vector with at most t errors. The permutation decoding algorithm proceeds by moving all errors in a received vector $\Phi(u)$ out of the information positions. That is, the nonzero coordinates of e are moved out of the information set $\Phi(I)$ for C_{bin}, by using an automorphism of C_{bin}.

Note that $\Phi(I)$ and $\Phi(S)$ are the sequences defined as above. Moreover, the function does not check whether I is an information set for C, nor whether S or $\Phi(S)$ is an s-PD-set for C_{bin} with respect to $\Phi(I)$, nor that $s \leq t$.

`PermutationDecode(C, I, S, s, Q)`

Given

- a code C over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$;

- an information set $I = [i_1, \ldots, i_{\gamma+\delta}] \subseteq \{1, \ldots, n\}$ for C as a sequence of coordinate positions, such that the code C punctured on the coordinates $\{1, \ldots, n\} \backslash \{i_{\gamma+1}, \ldots, i_{\gamma+\delta}\}$ is of type 4^δ;

- a sequence S such that either S or $\Phi(S)$ is an s-PD-set for $C_{bin} = \Phi(C)$, where Φ is the Gray map, with respect to $\Phi(I)$;

- an integer $s \in \{1, \ldots, t\}$, where t is the error-correcting capability of C_{bin};

- and a sequence Q of vectors from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$,

attempt to decode the vectors of Q with respect to C. This function is similar to the version of `PermutationDecode` that decodes a single vector except that it decodes a sequence of vectors and returns a sequence of booleans and two sequences of decoded vectors corresponding to the given sequence. The algorithm used is the same as that used by the single vector version of `PermutationDecode`.

Example 88. *First the Hadamard code* C *over* $\mathbb{Z}_4$ *of length* 32 *and type* $2^1 4^3$ *is constructed. It is known that* $I = [17, 1, 2, 5]$ *is an information set for* C *and* $S = \{\pi^i : 1 \leq i \leq 8\}$*, where* $\pi = (1, 24, 26, 15, 3, 22, 28, 13)$ $(2, 23, 27, 14, 4, 21, 25, 16)$ $(5, 11, 32, 20, 7, 9, 30, 18)$ $(6, 10, 29, 19, 8, 12, 31, 17)$*, is a subset of the permutation automorphism group of* C *such that* $\Phi(S)$ *is a* 7-PD*-set for* $C_{bin} = \Phi(C)$ *with respect to* $\Phi(I)$*. Then, choosing a codeword* c *of* C*,* c *is perturbed by the addition of an error vector to give a new vector* u*, and finally permutation decoding is applied to* u *to recover* c*.*

```
> C := HadamardCodeZ4(3, 6);
> C;
((32, 4^3 2^1)) Linear Code over IntegerRing(4)
Generator matrix:
[1 0 3 2 0 3 2 1 3 2 1 0 2 1 0 3 1 0 3 2 0 3 2 1 3 2 1 0 2 1 0 3]
[0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3]
[0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
> t := Floor((MinimumLeeDistance(C) - 1)/2);
> t;
15

> I := [17, 1, 2, 5];
> p := Sym(32)!(1, 24, 26, 15, 3, 22, 28, 13)(2, 23, 27, 14, 4, 21, 25, 16)
               (5, 11, 32, 20, 7, 9, 30, 18)(6, 10, 29, 19, 8, 12, 31,17);
> S := [ p^i : i in [1..8] ];
> SetVerbose("IsPDSetFlag", 2);
> IsPermutationDecodeSet(C, I, S, 7);
Checking whether I is an information set...
Checking whether S is in the permutation automorphism group...
Checking whether S is an s-PD-set...
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
Took 136.430 seconds (CPU time)
true
> SetVerbose("IsPDSetFlag", 0);

> c := C ! [1,2,3,0,0,1,2,3,3,0,1,2,2,3,0,1,3,0,1,2,2,3,0,1,1,2,3,0,0,1,2,3];
> c in C;
true
```

```
> u := c;
> u[1] := c[1] + 2;
> u[2] := c[2] + 2;
> u[3] := c[3] + 1;
> u[16] := c[16] + 3;
> u[27] := c[27] + 1;
> u in C;
false
> LeeDistance(u, c);
7

> grayMap := GrayMap(UniverseCode(Integers(4), Length(C)));
> cbin := grayMap(c);
> ubin := grayMap(u);
> Distance(ubin, cbin);
7

> isDecoded, uDecoded, ubinDecoded := PermutationDecode(C, I, S, 7, u);
> isDecoded;
true
> uDecoded eq c;
true
> ubinDecoded eq cbin;
true

> isDecoded, uDecoded, ubinDecoded := PermutationDecode(C, I, S, 7, ubin);
> isDecoded;
true
> uDecoded eq c;
true
> ubinDecoded eq cbin;
true
```

`PDSetHadamardCodeZ4(`δ`, m)`

`AlgMethod` MONSTGELT *Default* : "Deterministic"

Given an integer $m \geq 5$, and an integer δ such that $3 \leq \delta \leq \lfloor (m+1)/2 \rfloor$, the Hadamard code C over $\mathbb{Z}_4$ of length $n = 2^{m-1}$ and type $2^\gamma 4^\delta$, where $\gamma = m+1-2\delta$, given by the function `HadamardCodeZ4(`δ`, m)`, is considered. The function returns an information set $I = [i_1, \ldots, i_{\gamma+\delta}] \subseteq \{1, \ldots, n\}$ for C together with a subset S of the permutation automorphism group of C such that $\Phi(S)$ is an s-PD-set for $C_{bin} = \Phi(C)$ with respect to $\Phi(I)$, where Φ is the Gray map and $\Phi(I)$ and $\Phi(S)$ are defined above. The function also returns the information set $\Phi(I)$ and the s-PD-set $\Phi(S)$. For $m \geq 1$ and

$1 \leq \delta \leq 2$, the Gray map image of C is linear and it is possible to find an s-PD-set for $C_{bin} = \Phi(C)$, for any $s \leq \lfloor 2^m/(m+1) \rfloor - 1$, by using the function `PDSetHadamardCode(m)`.

The information sets I and $\Phi(I)$ are returned as sequences of $\gamma+\delta$ and $\gamma+2\delta$ integers, giving the coordinate positions that correspond to the information sets for C and C_{bin}, respectively. The sets S and $\Phi(S)$ are also returned as sequences of elements in the symmetric groups $\mathrm{Sym}(n)$ and $\mathrm{Sym}(2n)$ of permutations on the set $\{1, \ldots, n\}$ and $\{1, \ldots, 2n\}$, respectively.

A deterministic algorithm is used by default. In this case, the function returns the s-PD-set of size $s+1$ with $s = \lfloor (2^{2\delta-2} - \delta)/\delta \rfloor$, which is the maximum value of s when $\gamma = 0$, as described in [BV16a]. If the parameter `AlgMethod` is assigned the value `"Nondeterministic"`, the function tries to improve the previous result by finding an s-PD-set of size $s+1$ such that $\lfloor (2^{2\delta-2} - \delta)/\delta \rfloor \leq s \leq \lfloor (2^{m-1} + \delta - m - 1)/(m + 1 - \delta) \rfloor$. In this case, the function starts from the maximum value of s and decreases it if the s-PD-set is not found after a specified time.

`PDSetKerdockCodeZ4(m)`

Given an integer $m \geq 4$ such that $2^m - 1$ is not a prime number, the Kerdock code C over $\mathbb{Z}_4$ of length $n = 2^m$ and type 4^{m+1}, given by the function `KerdockCodeZ4(m)` is considered. The function returns the information set $I = [1, \ldots, m+1]$ for C together with a subset S of the permutation automorphism group of C such that $\Phi(S)$ is an s-PD-set for $C_{bin} = \Phi(C)$ with respect to $\Phi(I)$, where Φ is the Gray map and $\Phi(I)$ and $\Phi(S)$ are defined above. The function also returns the information set $\Phi(I) = [1, \ldots, 2m+2]$ and the s-PD-set $\Phi(S)$. The size of the s-PD-set S is always $\lambda = s+1$, where λ is the greatest divisor of $2^m - 1$ such that $\lambda \leq 2^m/(m+1)$.
The information sets I and $\Phi(I)$ are returned as sequences of $m+1$ and $2m+2$ integers, giving the coordinate positions that correspond to the information sets for C and C_{bin}, respectively. The sets S and $\Phi(S)$ are also returned as sequences of elements in the symmetric groups $\mathrm{Sym}(n)$ and $\mathrm{Sym}(2n)$ of permutations on the sets $\{1, \ldots, n\}$ and $\{1, \ldots, 2n\}$, respectively. The s-PD-set S contains the $s+1$ permutations described in [BV16b].

Example 89. *A* 4-PD-*set* S *of size* 5 *for the Hadamard code* C *over* $\mathbb{Z}_4$ *of length* 16 *and type* $2^0 4^3$ *is constructed. A check that it really is a* 4-PD-*set for* C *is then made. Note that* $\lfloor (2^{2\delta-2} - \delta)/\delta \rfloor = 4$. *Finally, a codeword* c *of* C *is selected, perturbed by an error vector* e *to give a vector* u, *to which permutation decoding is applied to recover* c.

```
> C := HadamardCodeZ4(3, 5);

> I, S, Ibin, Sbin := PDSetHadamardCodeZ4(3, 5);
> s := #Sbin-1; s;
4
> s eq Floor((2^(2*3-2)-3)/3);
true
> IsPermutationDecodeSet(C, I, S, s);
true
> IsPermutationDecodeSet(C, Ibin, Sbin, s);
true

> c := C ! [3,2,1,0,1,0,3,2,3,2,1,0,1,0,3,2];
> R := UniverseCode(Integers(4), Length(C));
> u := R ! [2,3,2,0,1,0,3,2,3,2,1,0,1,0,3,3];
> u in C;
false
> LeeDistance(u, c);
4
> grayMap := GrayMap(R);
> cbin := grayMap(c);

> isDecoded, uDecoded, ubinDecoded := PermutationDecode(C, I, S, 4, u);
> isDecoded;
true
> uDecoded eq c;
true
> ubinDecoded eq cbin;
true
```

For the Hadamard code C over $\mathbb{Z}_4$ of length 32 and type $2^1 4^3$, a 4-PD-set of size 5 can be constructed either by using the deterministic method (by default), or by using a nondeterministic method to obtain an s-PD-set of size $s+1$ with $4 \leq s \leq 7$. In both cases, the given sets are checked for really being s-PD-sets for C.

```
> C := HadamardCodeZ4(3, 6);

> I, S, Ibin, Sbin := PDSetHadamardCodeZ4(3, 6);
> s := #Sbin-1; s;
4
> IsPermutationDecodeSet(C, I, S, s);
true

> I, S, Ibin, Sbin := PDSetHadamardCodeZ4(3, 6 : AlgMethod := "Nondeterministic");
> s := #Sbin-1; s;
```

```
6
> IsPermutationDecodeSet(C, I, S, s);
true
```

Finally, a 2-PD*-set of size* 3 *is constructed for the Kerdock code of length* 16 *and type* $2^0 4^5$*, and formally checked for being a* 2-PD*-set for this code.*

```
> C := KerdockCode(4);

> I, S, Ibin, Sbin := PDSetKerdockCodeZ4(4);
> IsPermutationDecodeSet(C, I, S, 2);
true
> IsPermutationDecodeSet(C, Ibin, Sbin, 2);
true
```

6.4.5 Information space and information sets

`InformationSpace(C)`

Given a code C over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$, return the $\mathbb{Z}_4$-submodule of $\mathbb{Z}_4^{\gamma+\delta}$ isomorphic to $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$ such that the first γ coordinates are of order two, that is, the space of information vectors for C. The function also returns the $(\gamma + 2\delta)$-dimensional binary vector space, which is the space of information vectors for the corresponding binary code $C_{bin} = \Phi(C)$, where Φ is the Gray map. Finally, for the encoding process, it also returns the corresponding isomorphisms f and f_{bin} from these spaces of information vectors onto C and C_{bin}, respectively.

Example 90.

```
> C := LinearCode<Integers(4), 4 | [[2,0,0,2],[0,1,1,3]]>;
> R, V, f, fbin := InformationSpace(C);
> G := MinRowsGeneratorMatrix(C);

> (#R eq #C) and (#V eq #C);
true
> Set([f(i) : i in R]) eq Set(C);
true
> Set([i*G : i in R]) eq Set(C);
false

> i := R![2,3];
> c := f(i);
> c;
(2 3 3 3)
> c in C;
```

```
true
> i*G eq c;
false

> ibin := V![1,1,0];
> cbin := fbin(ibin);
> cbin;
(1 1 1 0 1 0 1 0)
> cbin in GrayMapImage(C);
true
> cbin eq GrayMap(C)(c);
true
```

`InformationSet(C)`

Given a code C over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$, return an information set $I = [i_1, \ldots, i_{\gamma+\delta}] \subseteq \{1, \ldots, n\}$ for C such that the code C punctured on $\{1, \ldots, n\} \backslash \{i_{\gamma+1}, \ldots, i_{\gamma+\delta}\}$ is of type 4^δ, and the corresponding information set $\Phi(I) = [2i_1 - 1, \ldots, 2i_\gamma - 1, 2i_{\gamma+1} - 1, 2i_{\gamma+1}, \ldots, 2i_{\gamma+\delta} - 1, 2i_{\gamma+\delta}] \subseteq \{1, \ldots, 2n\}$ for the binary code $C_{bin} = \Phi(C)$, where Φ is the Gray map. The information sets I and $\Phi(I)$ are returned as a sequence of $\gamma + \delta$ and $\gamma + 2\delta$ integers, giving the coordinate positions that correspond to the information set of C and C_{bin}, respectively.

An information set I for C is an ordered set of $\gamma + \delta$ coordinate positions such that $|C^I| = 2^\gamma 4^\delta$, where $C^I = \{v^I : v \in C\}$ and v^I is the vector v restricted to the I coordinates. An information set J for C_{bin} is an ordered set of $\gamma + 2\delta$ coordinate positions such that $|C^J_{bin}| = 2^{\gamma+2\delta}$.

`IsInformationSet(C, I)`

Given a code C over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$ and a sequence $I \subseteq \{1, \ldots, n\}$ or $I \subseteq \{1, \ldots, 2n\}$, return `true` if and only if $I \subseteq \{1, \ldots, n\}$ is an information set for C. This function also returns another boolean, which is `true` if an only if $I \subseteq \{1, \ldots, 2n\}$ is an information set for the corresponding binary code $C_{bin} = \Phi(C)$, where Φ is the Gray map.

An information set I for C is an ordered set of $\gamma + \delta$ coordinate positions such that $|C^I| = 2^\gamma 4^\delta$, where $C^I = \{v^I : v \in C\}$ and v^I is the vector v restricted to the I coordinates. An information set J for C_{bin} is an ordered set of $\gamma + 2\delta$ coordinate positions such that $|C^J_{bin}| = 2^{\gamma+2\delta}$.

Example 91.

```
> C := HadamardCodeZ4(3,6);
```

```
> C;
((32, 4^3 2^1)) Linear Code over IntegerRing(4)
Generator matrix:
[1 0 3 2 0 3 2 1 3 2 1 0 2 1 0 3 1 0 3 2 0 3 2 1 3 2 1 0 2 1 0 3]
[0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3]
[0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

> I, Ibin := InformationSet(C);
> I;
[ 16, 28, 31, 32 ]
> Ibin;
[ 31, 55, 56, 61, 62, 63, 64 ]
> #PunctureCode(C, {1..32} diff Set(I)) eq #C;
true
> Cbin := GrayMapImage(C);
> V := VectorSpace(GF(2), 7);
> #{V![c[i] : i in Ibin] : c in Cbin} eq #Cbin;
true

> IsInformationSet(C, I);
true false
> IsInformationSet(C, Ibin);
false true

> IsInformationSet(C, [1, 2, 5, 17]);
true false
> IsInformationSet(C, [1, 2, 3, 4, 9, 10, 33]);
false true

> D := LinearCode<Integers(4), 5 | [[2,0,0,2,0],[0,2,0,2,2],[0,0,2,2,0]]>;
> IsInformationSet(D, [1,3,5]);
true true
```

6.4.6 Syndrome space and coset leaders

`SyndromeSpace(C)`

Given a code C over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$, return the $\mathbb{Z}_4$-submodule of $\mathbb{Z}_4^{n-\delta}$ isomorphic to $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^{n-\gamma-\delta}$ such that the first γ coordinates are of order two, that is, the space of syndrome vectors for C. The function also returns the $(2n-2\delta-\gamma)$-dimensional binary vector space, which is the space of syndrome vectors for the corresponding binary code $C_{bin} = \Phi(C)$, where Φ is the Gray map. Note that these spaces are computed by using the function `InformationSpace(C)` applied to the dual code of C, produced by function

`Dual(C)`.

`Syndrome(u, C)`

Given a code C over $\mathbb{Z}_4$ of length n and type $2^\gamma 4^\delta$, and a vector u from the ambient space $V = \mathbb{Z}_4^n$ or $V_2 = \mathbb{Z}_2^{2n}$, construct the syndrome of u relative to the code C. This will be an element of the syndrome space of C, considered as the $\mathbb{Z}_4$-submodule of $\mathbb{Z}_4^{n-\delta}$ isomorphic to $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^{n-\gamma-\delta}$ such that the first γ coordinates are of order two.

`CosetLeaders(C)`

Given a code C over $\mathbb{Z}_4$ of length n, with ambient space $V = \mathbb{Z}_4^n$, return a set of coset leaders (vectors of minimal Lee weight in their cosets) for C in V as an indexed set of vectors from V. This function also returns a map from the syndrome space of C onto the coset leaders (mapping a syndrome into its corresponding coset leader). Note that this function is only applicable when V and C are small.

Example 92.

```
> C := LinearCode<Integers(4), 4 | [[2,0,0,2],[0,1,1,3]]>;
> R, V, f, fbin := InformationSpace(C);
> Rs, Vs := SyndromeSpace(C);

> #R * #Rs eq 4^Length(C);
true
> #V * #Vs eq 4^Length(C);
true

> i := R![2,3];
> c := f(i);
> c;
(2 3 3 3)
> u := c;
> u[2] := u[2]+3;
> u;
(2 2 3 3)

> s := Syndrome(u, C);
> s in Rs;
true
> H := Transpose(MinRowsGeneratorMatrix(Dual(C)));
> s eq u*H;
true

> L, mapCosetLeaders := CosetLeaders(C);
```

```
> errorVector := mapCosetLeaders(s);
> errorVector;
(0 3 0 0)
> errorVector in L;
true
> u-errorVector eq c;
true
```

6.5 MAGMA functions for codes over finite fields

This section provides functions for decoding vectors from the ambient space of a linear code C over a finite field by using the permutation decoding method. If the parameter s is less than the error-correcting capability of the code, then a partial permutation decoding is performed. A function to know whether or not a set of permutations S is an s-PD-set for a code C with respect to an information set I is included. Moreover, functions for constructing the s-PD-sets of size $s+1$ described in Chapter 3 for the family of binary linear Hadamard codes, and the ones already known for simplex codes [FKM12] can also be found. The section also provides detailed examples for all the implemented functions, in order to see better how they work.

6.5.1 Permutation decoding

`IsPermutationDecodeSet(C, I, S, s)`

Given

- an $[n, k]$ linear code C over a finite field K;
- a sequence $I \subseteq \{1, \ldots, n\}$;
- a sequence S of elements in the group of monomial matrices of degree n over K, OR if C is a binary code, a sequence of elements in the symmetric group $\mathrm{Sym}(n)$ acting on the set $\{1, \ldots, n\}$;
- and an integer $s \in \{1, \ldots, t\}$, where t is the error-correcting capability of C;

this intrinsic returns `true` if and only if S is an s-PD-set for C with respect to the information set I.

Depending on the length of the code C, its dimension k, and the integer s, this function could take some time to compute whether S is an s-PD-set for

C with respect to I. Specifically, if the function returns `true`, it is necessary to check $\sum_{i=1}^{s} \binom{k}{i} \cdot \binom{n-k}{s-i}$ s-sets.

The verbose flag `IsPDSetFlag` is set to level 0 by default. If it is set to level 1, the total time used to check the condition is shown. Moreover, the reason the function returns `false` is also shown, that is, whether I is not an information set, S is not a subset of the monomial automorphism group of C or S is not an s-PD-set. If it is set to level 2, the percentage of the computation process performed is also printed.

`PermutationDecode(C, I, S, s, u)`

Given

- an $[n, k]$ linear code C over a finite field K;
- an information set $I \subseteq \{1, \ldots, n\}$ for C as a sequence of coordinate positions;
- a sequence S of elements in the group of monomial matrices of degree n over K, OR if C is a binary code, a sequence of elements in the symmetric group $\mathrm{Sym}(n)$ acting on the set $\{1, \ldots, n\}$. In either case S must be an s-PD-set for C with respect to I;
- an integer $s \in \{1, \ldots, t\}$, where t is the error-correcting capability of C;
- and a vector u from the ambient space V of C,

the intrinsic attempts to decode u with respect to C. If the decoding algorithm succeeds in computing a vector $u' \in C$ as the decoded version of $u \in V$, then the function returns `true` and the codeword u'. If the decoding algorithm does not succeed in decoding u, then the function returns `false` and the zero vector in V.

The permutation decoding algorithm works by moving all errors in the received vector $u = c + e$, where $c \in C$ and $e \in V$ is the error vector with at most t errors, out of the information positions, that is, moving the nonzero coordinates of e out of the information set I for C, by using an automorphism of C. Note that the function does not check any of the conditions that I is an information set for C, S is an s-PD-set for C with respect to I, or $s \leq t$.

`PermutationDecode(C, I, S, s, Q)`

Given

- an $[n, k]$ linear code C over a finite field K;

- an information set $I \subseteq \{1, \ldots, n\}$ for C as a sequence of coordinate positions;

- a sequence S of elements in the group of monomial matrices of degree n over K, OR if C is a binary code, a sequence of elements in the symmetric group $\mathrm{Sym}(n)$ acting on the set $\{1, \ldots, n\}$. In either case S must be an s-PD-set for C with respect to I;

- an integer $s \in \{1, \ldots, t\}$, where t is the error-correcting capability of C;

- and a sequence Q of vectors from the ambient space V of C,

the intrinsic attempts to decode the vectors of Q with respect to C. This function is similar to the function `PermutationDecode(C, I, S, s, u)` except that rather than decoding a single vector, it decodes a sequence of vectors and returns a sequence of booleans and a sequence of decoded vectors corresponding to the given sequence. The algorithm used is as for the function `PermutationDecode(C, I, S, s, u)`.

`PDSetSimplexCode(K, m)`

Given a finite field K of cardinality q, and a positive integer m, the intrinsic constructs the $[n = (q^m - 1)/(q - 1), m, q^{m-1}]$ linear simplex code C over K, as `Dual(HammingCode(K, m))`, and then searches for an s-PD-set for C. The function returns an information set I for C together with a subset S of the monomial automorphism group of C such that S is an s-PD-set for C with respect to I, where $s = \lfloor (q^m - 1)/(m(q - 1)) \rfloor - 1$.

The information set I is returned as a sequence of m integers, giving the coordinate positions that correspond to the information set for C. The set S is also returned as a sequence, which contains the $s+1$ elements in the group of monomial matrices of degree n over K described in [FKM12]. When K is $GF(2)$, the function also returns the elements of S represented as elements in the symmetric group $\mathrm{Sym}(n)$ of permutations on the set $\{1, \ldots, n\}$.

`PDSetHadamardCode(m)`

Given a positive integer m, the intrinsic constructs the $[2^m, m + 1, 2^{m-1}]$ binary linear Hadamard code C, as `Dual(ExtendCode(HammingCode(GF(2), m)))`, and then searches for an s-PD-set for C. The function returns an

information set $I \subseteq \{1, \ldots, 2^m\}$ for C together with a subset S of the permutation automorphism group of C such that S is an s-PD-set for C with respect to I, where $s = \lfloor 2^m/(m+1) \rfloor - 1$.

The information set I is returned as a sequence of $m+1$ integers, giving the coordinate positions that correspond to the information set for C. The set S is also returned as a sequence, which contains the $s+1$ elements in the group of permutation matrices of degree 2^m over $GF(2)$ described in [BV16a]. The function also returns the elements of S represented as elements in the symmetric group $\mathrm{Sym}(2^m)$ of permutations on the set $\{1, \ldots, 2^m\}$.

Example 93.

```
> C := Dual(ExtendCode(HammingCode(GF(2), 5)));
> C;
[32, 6, 16] Linear Code over GF(2)
Generator matrix:
[1 0 0 0 0 0 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1]
[0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0]
[0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 1 1 1 1]
[0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0 0]
[0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0]
[0 0 0 0 0 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1]

> I, SMAut, SPAut := PDSetHadamardCode(5);
> I in AllInformationSets(C);
true
> s := #SMAut-1;  s;
4
> [ LinearCode(GeneratorMatrix(C)*SMAut[i]) eq C : i in [1..s+1] ];
[true, true, true, true, true];
> [ LinearCode(GeneratorMatrix(C)^SPAut[i]) eq C : i in [1..s+1] ];
[true, true, true, true, true];
> IsPermutationDecodeSet(C, I, SMAut, s);
true
> IsPermutationDecodeSet(C, I, SPAut, s);
true

> c := C ! [1^^32];
> c in C;
true
> u := c;
> u[1] := c[1] + 1;
> u[2] := c[2] + 1;
> u[4] := c[4] + 1;
> u[32] := c[32] + 1;
> u in C;
```

```
false
> isDecoded, uDecoded := PermutationDecode(C, I, SMAut, s, u);
> isDecoded;
true
> uDecoded eq c;
true
> isDecoded, uDecoded := PermutationDecode(C, I, SPAut, s, u);
> isDecoded;
true
> uDecoded eq c;
true
```

Example 94.

```
> K<a> := GF(4);
> C := Dual(HammingCode(K, 3));
> C;
[21, 3, 16] Linear Code over GF(2^2)
Generator matrix:
[1 0 a^2 a 1 0 a^2 a 1 a^2 0 1 a a 1 0 a^2 1 a a^2 0]
[0 1 1 1 1 0 0 0 0 a^2 a^2 a^2 a^2 a a a a 1 1 1 1]
[0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

> I, SMAut := PDSetSimplexCode(K, 3);
> I in AllInformationSets(C);
true
> s := #SMAut-1;  s;
6
> [ LinearCode(GeneratorMatrix(C)*SMAut[i]) eq C : i in [1..s+1] ];
[true, true, true, true, true, true, true];
> IsPermutationDecodeSet(C, I, SMAut, s);
true

> c := C ! [0,1,1,1,1,0,0,0,0,a^2,a^2,a^2,a^2,a,a,a,a,1,1,1,1];
> c in C;
true
> u := c;
> u[1] := c[1] + a;
> u[2] := c[2] + a^2;
> u[3] := c[3] + a;
> u[4] := c[4] + a^2;
> u[5] := c[5] + a;
> u[6] := c[6] + a^2;
> u in C;
false
> isDecoded, uDecoded := PermutationDecode(C, I, SMAut, s, u);
> isDecoded;
true
```

```
> uDecoded eq c;
true
```

Chapter 7

Conclusions

7.1 Summary

Finding efficient decoding methods is of great interest in coding theory due to their practical applications. Permutation decoding is a decoding method, developed by MacWilliams [Mac64] and Prange [Pra62], which employs a fixed set of automorphisms of a linear code to assist in decoding received vectors. This technique is described in [MS77, Chapter 16], [HP03, Chapter 10], and [Huf98, Section 8]. Roughly speaking, this method aims to move all the errors in a received vector out of the information positions, using an automorphism from that special fixed set of automorphisms of the code called PD-set, in such a way that the information positions in the permuted vector are correct. Since the components of the permuted vector in the information positions are correct, one can easily obtain a vector (for example, by encoding the correct information) that must be a codeword, only permuted. By applying the inverse of the automorphism (that must be a permutation, since we are working with binary codes) to this new codeword, we find the codeword that was originally transmitted.

An open problem is to determine appropriate PD-sets for the code where we would like to perform this decoding method. Since small PD-sets are more efficient, there has been some interest in finding PD-sets of minimum size (i.e. where the Gordon-Schonheim bound is attained) for different families of linear codes. This is the case, for example, of binary Golay codes, where two PD-sets of (minimum) size 14 have been found independently in [Gor82, Wol83]. In [KMM05], the definition of PD-sets is extended to that of s-PD-sets to correct s errors, where s is lower than or equal to the error-correcting capability of the code. The question of determining s-PD-sets is addressed

for simplex codes in [KV08, FKM12], MacDonald codes in [KS16], binary Reed-Muller codes in [Sen09, KMM10], being the size of these sets minimal in some cases.

An alternative permutation decoding method that can be applied to certain binary nonlinear codes as $\mathbb{Z}_2\mathbb{Z}_4$-linear codes [BFP$^+$10], which include $\mathbb{Z}_4$-linear codes, was presented in [BBFV15]. However, the determination of s-PD-sets remained an open question. Since then, as far as we know, no work in this sense has been done.

In this dissertation, we first provide an upper bound f_m that represents the maximum s for which s-PD-sets of size $s+1$ for systematic Hadamard codes and $\mathbb{Z}_4$-linear Kerdock codes of length 2^m can be found. We also compute another bound, denoted by $f_{\gamma,\delta}$, for the $\mathbb{Z}_4$-linear Hadamard code $H_{\gamma,\delta} = \Phi(\mathcal{H}_{\gamma,\delta})$ of length $2^m = 2^{\gamma+2\delta-1}$, that becomes essential when s-PD-sets of size $s+1$ are searched for in the subgroup $\Phi(\mathrm{PAut}(\mathcal{H}_{\gamma,\delta}))$ of $\mathrm{PAut}(H_{\gamma,\delta})$. The bound $f_{\gamma,\delta}$ is generally smaller than f_m, since f_m is valid when we examine the whole permutation group $\mathrm{PAut}(H_{\gamma,\delta})$ to find these sets. Surprisingly, it turns out that $f_{0,\delta} = f_m$.

Then we focus on constructing s-PD-sets of minimum size $s+1$ for binary linear and $\mathbb{Z}_4$-linear Hadamard codes. The idea used here is similar to the one introduced for simplex codes in [FKM12]. For binary linear Hadamard codes H_m of length 2^m with $m \geq 4$ and (nonlinear) $\mathbb{Z}_4$-linear Hadamard codes $H_{0,\delta}$ of length $2^{2\delta-1}$ with $\delta \geq 3$, s-PD-sets of size $s+1$ with s up to the upper bound $f_m = f_{0,\delta}$ have been constructed. Moreover, for $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ with $\gamma > 0$ and $\delta \geq 3$, s-PD-sets of size $s+1$ up to $f_{0,\delta}$ are given. The knowledge of the permutation automorphism group of the $\mathbb{Z}_4$-linear Hadamard codes [PPV14, KV15] was helpful due to its crucial role in this decoding method.

In this work, we also present a theorem that states sufficient conditions for a permutation $\sigma \in \mathrm{PAut}(C)$ to generate an s-PD-set $S = \{\sigma^i : 1 \leq i \leq s+1\}$ of size $s+1$ for a $\mathbb{Z}_4$-linear code C. This new second approach may be applied to any $\mathbb{Z}_4$-linear code under certain conditions, unlike the first approach, which can only be applied to $\mathbb{Z}_4$-linear Hadamard codes. In particular, this new method also applies to $\mathbb{Z}_4$-linear Hadamard codes, so s-PD-sets have been constructed for these codes by using this new approach. Specifically, we obtain new s-PD-sets of size $s+1$ for the $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ for all $\delta \geq 4$ and $1 < s \leq 2^\delta - 3$. These new sets are generated by a single permutation, unlike the previously presented s-PD-sets of size $s+1$ for $\mathbb{Z}_4$-linear Hadamard codes. Howeover, $2^\delta - 3 \leq f_{\gamma,\delta}$ for all $\delta \geq 3$ and $\gamma \geq 0$, and hence the first approach becomes better to correct the greater number

of errors for $H_{\gamma,\delta}$. We also obtain s-PD-sets of size $s+1$, for all $m \geq 5$ and $1 < s \leq \lambda - 1$, for the binary Kerdock code of length 2^m such that $2^{m-1}-1$ is not prime, where λ is the greatest divisor of $2^{m-1}-1$ satisfying $\lambda \leq 2^{m-1}/m$. Unlike for $\mathbb{Z}_4$-linear Kerdock codes, the upper bound f_m for $\mathbb{Z}_4$-linear Hadamard codes K_m of length 2^m is achieved for some values of m by using the second approach.

A new package that expands the current functionality for quaternary linear codes in MAGMA, including functions to decode these codes and also their binary images under the Gray map (that is, the $\mathbb{Z}_4$-linear codes), has been developed. Functions related to the information space, information sets, syndrome space and coset leaders for quaternary linear codes have been implemented, as well as functions to decode these codes by using four different methods: coset, syndrome, lifted and permutation decoding. The explicit construction of s-PD-sets of size $s+1$ for $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ up to $s = f_{0,\delta}$ given in Chapter 4 has been implemented. Likewise, the function concerning s-PD-sets of size $s+1$ for $\mathbb{Z}_4$-linear Kerdock codes K_m is based on the construction presented for this codes in Chapter 5. The performance of the permutation decoding algorithm when these s-PD-sets are employed has been evaluated by real tests and have been compared with the outcomes of the rest of decoding methods implemented. When correcting up to s errors, permutation decoding is the method with the best behaviour among the methods implemented. The permutation decoding algorithm for linear codes over finite fields has also been implemented. The latest version of this package for codes over $\mathbb{Z}_4$ and this manual with the description of all developed functions can be downloaded from the web page `http://ccsg.uab.cat` [BCFS16, BPPV16].

7.2 Future research

In this section, we indicate some open problems that derive from this dissertation which may be considered for future research on this topic:

- Finding an explicit construction of s-PD-sets of size $s+1$ for the $\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta}$ of length $2^{\gamma+2\delta-1}$ with $\gamma > 0$ and $\delta \geq 3$ for all $f_{0,\delta} < s \leq f_{\gamma,\delta}$ by using the first approach, where $f_{\gamma,\delta} = \lfloor \frac{2^{\gamma+2\delta-2}-\gamma-\delta}{\gamma+\delta} \rfloor$. For $\gamma = 1$ and $\delta \in \{3,4\}$, these s-PD-sets of size $s+1$ have already been found and are exposed in Examples 58 and 59.

- Providing an explicit construction of s-PD-sets of size $s+1$ for the

$\mathbb{Z}_4$-linear Hadamard codes $H_{\gamma,\delta} = \Phi(\mathcal{H}_{\gamma,\delta})$ of length $2^{\gamma+2\delta-1}$ with $\gamma > 0$ and $\delta \geq 3$ for $f_{\gamma,\delta} < s \leq f_m$, where $f_m = \lfloor \frac{2^m}{m+1} \rfloor - 1$. The automorphism groups $\mathrm{MAut}(\mathcal{H}_{\gamma,\delta})$ and $\mathrm{PAut}(H_{\gamma,\delta})$ may be investigated to achieve this goal.

- Giving an explicit construction of s-PD-sets of size $s+1$ for the $\mathbb{Z}_4$-linear Kerdock codes of length 2^m where Corollary 78 cannot be applied (that is, those where $2^{m-1}-1$ is a prime number). Improving the s-PD-sets of size $s+1$ for the $\mathbb{Z}_4$-linear Kerdock codes of length 2^m where Corollary 78 applies but the bound $f_m = \lfloor \frac{2^{m-1}}{m} \rfloor - 1$ is not achieved.

- Finding PD-sets, that is, s-PD-sets where s equals the error-correcting capability of the code, for the codes considered throughout this work: binary linear Hadamard codes, $\mathbb{Z}_4$-linear Hadamard codes and $\mathbb{Z}_4$-linear Kerdock codes. Study whether or not the Gordon-Schonheim bound is sharp for these codes. For values of $s \geq f_m$, one may ask whether it is possible to obtain s-PD-sets of size $s+i$ for $i \geq 2$.

- In [PRS09], new families of quaternary linear codes, the quaternary linear Reed-Muller codes, denoted by $\mathcal{RM}_\ell(r,m)$, are constructed in such a way that, after applying the Gray map, the corresponding $\mathbb{Z}_4$-linear codes have the same parameters and properties as the binary linear Reed-Muller codes. In particular, the codes $\mathcal{RM}_\ell(1,m)$, after applying the Gray map, are $\mathbb{Z}_4$-linear Hadamard codes. A natural generalization of the results of this work is to find s-PD-sets for the $\mathbb{Z}_4$-linear codes obtained as the Gray map image of $\mathcal{RM}_\ell(r,m)$.

- In [KV15], it is shown that each $\mathbb{Z}_4$-linear Hadamard code is equivalent to a $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard code with a nontrivial binary part. Thus, for some of the $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes, the existence of s-PD-sets of size $s+1$ is already covered by the results given in this dissertation. Actually, this means that it only remains to determine s-PD-sets of minimum size $s+1$ for all values of s up to the upper bound f_m for $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes of length 2^m, m even, and type $2^1 4^{\frac{m}{2}}$.

- Implementing new functions in MAGMA for coset, lifted, syndrome and permutation decoding of $\mathbb{Z}_2\mathbb{Z}_4$-linear codes, both extending the current package for $\mathbb{Z}_2\mathbb{Z}_4$-linear codes and generalizing the package for $\mathbb{Z}_4$-linear codes done during this dissertation. Studying other known decoding methods for $\mathbb{Z}_4$-linear Hadamard and Kerdock codes and comparing their performance with the decoding methods already implemented.

- Studying other families of $\mathbb{Z}_4$-linear codes where Theorem 60 (i.e. the second approach) allows us to obtain s-PD-sets of size $s+1$. In this case, providing explicit constructions to obtain such sets for these codes.

- In [KZ13], a family of $\mathbb{Z}_4$-linear codes, called *extended dualized Kerdock codes*, is constructed as the Gray map image of quaternary linear codes with a high minimum Lee distance. In [KZ13, KWZ16], it is shown that in some cases these codes are better-than-linear and a table consisting of all known better-than-linear $\mathbb{Z}_4$-linear codes is provided. Among these $\mathbb{Z}_4$-linear codes, it would be interesting to provide PD-sets for those with high error-correcting capability. The study of their permutation automorphism groups could be a first step towards achieving this goal.

- Defining the permutation decoding algorithm for systematic binary nonlinear codes with subjacent ring structures different from $\mathbb{Z}_4$. For example, for the family of $\mathbb{Z}_{2^k}$-linear Hadamard codes introduced in [Kro07].

Bibliography

[AK92] E. F. Assmus and J. D. Key, *Designs and Their Codes*, Cambridge University Press, Great Britain, 1992.

[AA09] N. Aydin and T. Asamov, "A database of $\mathbb{Z}_4$ codes," *J. of Combinatorics, Information and System Sciences*, vol. 34, nos. 1–4, pp. 1–12, 2009. `http://Z4Codes.info/`.

[AS13] I. Aydogdu and I. Siap, "The structure of $\mathbb{Z}_2\mathbb{Z}_{2^s}$-additive codes: bounds on the minimum distance," *Appl. Math. Inf. Sci.*, vol. 7, no. 6, pp. 2271–2278, 2013.

[AS14] I. Aydogdu and I. Siap, "On $\mathbb{Z}_{p^r}\mathbb{Z}_{p^s}$-additive codes," *Linear and Multilinear Algebra*, vol. 63, pp. 2089–2102, 2014.

[BZ01] N. S. Babu and K.-H. Zimmermann, "Decoding of linear codes over Galois rings," *IEEE Trans. Inf. Theory*, vol. 47, pp. 1599–1603, 2001.

[BPV16] R. D. Barrolleta, J. Pujol, and M. Villanueva, "Comparing decoding methods for quaternary linear codes," to appear in *Electron. Note Discr. Math.*, 2016.

[BBS+15] R. D. Barrolleta, M. De Boeck, L. Storme, E. Suárez-Canedo, and P. Vandendriessche, "A geometrical bound for the sunflower property," in Proc. of *Design and Application of Random Network Codes (DARNEC '15)*, Istanbul, Turkey, pp. 39, 4–6 November 2015.

[BBS+16] R. D. Barrolleta, M. De Boeck, L. Storme, E. Suárez-Canedo, and P. Vandendriessche, "On constant distance random network codes," in Proc. of *Network Coding and Designs*, Dubrovnik, Croatia, pp. 48–49, 4–8 April 2016.

[BSSV16] R. D. Barrolleta, L. Storme, E. Suárez-Canedo, and P. Vandendriessche, "On primitive constant dimension codes and a geometrical sunflower bound," submitted to *Adv. in Math. of Commun.*, 2016.

[BV14] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for binary linear Hadamard codes," *Electron. Note Discr. Math.*, vol. 46, pp. 35–42, 2014.

[BV15] R. D. Barrolleta and M. Villanueva, "PD-sets for (nonlinear) Hadamard $\mathbb{Z}_4$-linear codes," in Proc. of the *21st Conference on Aplications of Computer Algebra (ACA 2015)*, Kalamata, Greece, pp. 135–139, 20–23 July 2015.

[BV16a] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for binary linear and $\mathbb{Z}_4$-linear Hadamard codes," submitted to *Designs, Codes and Cryptography*, 2016. arXiv:1512.01839

[BV16b] R. D. Barrolleta and M. Villanueva, "PD-sets for $\mathbb{Z}_4$-linear codes: Hadamard and Kerdock codes," in Proc. of the *IEEE International Symposium on Information Theory (ISIT 2016)*, Barcelona, pp. 1317–1321, 10–15 July 2016.

[BV16c] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for some families of $\mathbb{Z}_4$-linear codes," 2016 (in preparation).

[BPPV16] R. D. Barrolleta, J. Pernas, J. Pujol, and M.Villanueva, "Codes over $\mathbb{Z}_4$. A MAGMA package," version 2.0, Universitat Autònoma de Barcelona, 2016. `http://ccsg/uab.cat`.

[BBFV15] J. J. Bernal, J. Borges, C. Fernández-Córboda, and M. Villanueva, "Permutation decoding of $\mathbb{Z}_2\mathbb{Z}_4$-linear codes," *Des. Codes and Cryptogr.*, vol. 76, no. 2, pp. 269–277, 2015.

[BS11] J. J. Bernal and J. J. Simón, "Information sets from defining sets in abelian codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7990–7999, 2011.

[BS13] J. J. Bernal and J. J. Simón, "Partial permutation decoding for abelian codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 5152–5170, 2013.

[BEM99] A. Beutelspacher, J. Eisfeld, and J. Müller, "On sets of planes in projective space intersecting mutually in one point," *Geometriae Dedicata*, vol. 78, pp. 143–159, 1999.

[BFP+10] J. Borges, C. Fernández-Córdoba, J. Pujol, J. Rifà, and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-linear codes: generator matrices and duality," *Des. Codes and Cryptogr.*, vol. 54, no. 2, pp. 167–179, 2010.

[BFP+12] J. Borges, C. Fernández, J. Pujol, J. Rifà, and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-linear codes. A MAGMA package," version 3.5, Universitat Autònoma de Barcelona, 2012. `http://ccsg/uab.cat`.

[BFP+14] J. Borges, C. Fernández-Córdoba, J. Pujol, J. Rifà, and M. Villanueva, "Survey on $\mathbb{Z}_2\mathbb{Z}_4$-additive codes," in Proc. of the *contact forum Galois geometries and applications*. Royal Flemish Academy of Belgium for Science and the Arts (October 5, 2012), pp. 19–67, 2014.

[BPR03] J. Borges, K. T. Phelps, and J. Rifà, "The rank and kernel of extended 1-perfect $\mathbb{Z}_4$-linear and additive non-$\mathbb{Z}_4$-linear codes," *IEEE Trans. Inf. Theory*, vol, 49, no. 8, pp. 2028–2034, 2003.

[BPRZ03] J. Borges, K. T. Phelps, J. Rifà, and V. Zinoviev, "On $\mathbb{Z}_4$-linear Preparata-like and Kerdock-like codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2834–2843, 2003.

[BCFS16] W. Bosma, J. J. Cannon, C. Fieker, and A. Steel (eds.), *Handbook of Magma functions*, Edition 2.22 (2016) 5669 pages. `http://magma.maths.usyd.edu.au/magma/`.

[BHK92] S. Botzas, R. Hammons, and P. V. Kumar, "4-phase sequences with near-optimum correlations propierties," *IEEE Trans. Inf. Theory*, vol. 38, no. 3, pp. 1101–1113, 1992.

[Car91] C. Carlet, "The automorphism group of Kerdock codes," *J. Inf. Optimization Sci.*, vol. 12, pp. 378–400, 1991.

[DF11] S. T. Dougherty and C. Fernández-Córdoba, "Codes over $\mathbb{Z}_{2^k}$, Gray map and self-dual codes," *Adv. in Math. of Commun.*, vol. 5, no. 4, pp. 571–588, 2011.

[Eis02] J. Eisfeld, "On sets of n-dimensional subspaces of projective spaces intersecting mutually in an $(n-2)$-dimensional subspace," *Discrete Mathematics*, vol. 255, pp. 81–85, 2002.

[ER14] T. Etzion and N. Raviv, "Equidistant codes in the Grassmannian," 2015. arXiv:1308.6231v4

[FPV08] C. Fernández-Córdoba, J. Pujol, and M. Villanueva, "On rank and kernel of $\mathbb{Z}_4$-linear codes," *Lect. Notes Comput. Sc.*, vol. 5228, pp. 46–55, 2008.

[FPV10] C. Fernández-Córdoba, J. Pujol, and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-linear codes: rank and kernel," *Des. Codes and Cryptogr.*, vol. 56, no. 1, pp. 43–59, 2010.

[FKM12] W. Fish, J. D. Key, and E. Mwambene, "Partial permutation decoding for simplex codes," *Adv. Math. Commun.*, vol. 6, no. 4, pp. 505–516, 2012.

[Gor82] D. M. Gordon, "Minimal permutation sets for decoding the binary Golay codes," *IEEE Trans. Inf. Theory*, vol. 28, no. 3, pp. 541–543, 1982.

[Gra09] M. Grassl, "Code Tables: Bounds on the parameters of various types of codes," online available at `http://www.codetables.de`. Accessed on 2016-09-18.

[GV98] M. Greferath and U. Velbinger, "Efficient decoding of $\mathbb{Z}_{p^k}$-linear codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1288–1291, 1998.

[HKC+94] A. R. Hammons, Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, "The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 301–319, 1994.

[Hor07] K. J. Horadam, *Hadamard Matrices and Their Applications*, Princeton University Press, 2007.

[Huf98] W. C. Huffman, *Codes and Groups,* Handbook of Coding Theory, (V. S. Pless and W. C. Huffman, eds.), Elsevier, 1998.

[HP03] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, Cambridge, 2003.

[Kan95] W. H. Kantor, "Codes, quadratic forms and finite geometries. Different aspects of coding theory," in Proc. of *Symp. Appl. Math.*, San Francisco, pp. 153–177, 1995.

[Ker72] A. M. Kerdock, "A class of low-rate nonlinear binary codes," *Inf. and Control*, vol. 20, pp. 182–187, 1972.

[KMM05] J. D. Key, T. P. McDonough, and V. C. Mavron, "Partial permutation decoding for codes from finite planes," *European J. Combin.*, vol. 25, no. 22, pp. 665–682, 2005.

[KMM10] J. D. Key, T. P. McDonough, and V. C. Mavron, "Reed-Muller codes and permutation decoding," *Discrete Math.*, vol. 310, no. 22, pp. 3114–3119, 2010.

[KMM16] J. D. Key, T. P. McDonough, and V. C. Mavron, "Improved partial permutation decoding for Reed-Muller codes," submitted to *Discrete Math.*, 2016.

[KS16] J. D. Key and P. Seneviratne, "Partial Permutation decoding for MacDonald codes," *Applicable Algebra in Engineering, Communication and Computing*, pp. 1–14, 2016. doi: 10.1007/s00200-016-0286-7

[KWZ16] M. Kiermaier, A. Wassermann, and J. Zwanzger, "New upper bounds on binary linear codes and a $\mathbb{Z}_4$-code with a better-than-linear Gray image," 2016. arXiv:1503.03394.

[KZ13] M. Kiermaier and J. Zwanzger, "New ring-linear codes from dualization in projective Hjelmslev geometries," *Des. Codes Cryptogr.*, vol. 66, nos. 1–3, pp. 39–55, 2013.

[Kro16] D. S. Krotov, "On the automorphism group of the Z_2Z_4-linear 1-perfect and Preparata-like codes," *Des. Codes Cryptogr.*, 2016. doi:10.1007/s10623-016-0218-3

[Kro07] D. S. Krotov, "On $\mathbb{Z}_{2^k}$-dual binary codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1532–1537, 2007.

[KV08] H.-J. Kroll and R. Vincenti, "PD-sets for binary RM-codes and the codes related to the Klein quadric and to the Schubert variety of PG(5,2)," *Discrete Math.*, vol. 308, nos. 2–3, pp. 408–414, 2008.

[KV15] D. S. Krotov and M. Villanueva, "Classification of the $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard codes and their automorphism groups," *IEEE Trans. Inf. Theory*, vol. 61, no. 2, pp. 887–894, 2015.

[Kro01] D. S. Krotov, "$\mathbb{Z}_4$-linear Hadamard and extended perfect codes," *Electron. Note Discr. Math.*, vol. 6, pp. 107–112, 2001.

[Lam13] L. Lambert, *Random Network Coding and Designs over* $\mathbb{F}_q$, Master dissertation, Ghent University, 2013. `http://www.network-coding.eu/pubs/Thesis-Lien.pdf`.

[LRS99] S. Litsyn, R. M. Rains, and N. J. A. Sloane, "Table of nonlinear binary codes," online available at `http://www.eng.tau.ac.il/ litsyn/tableand/`. Accessed on 2016-09-18.

[Mac60] J. E. MacDonald, "Design methods for maximum minimum-distance error-correcting codes," *IBM J. Res. Dev.*, vol. 4, pp. 43–57, 1960.

[Mac64] F. J. MacWilliams, "Permutation decoding of systematics codes," *Bell System Tech. J.*, vol. 43, pp. 485–505, 1964.

[MS77] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing Company, 1977.

[Max51] M. W. Maxfield, "The order of a matrix under multiplication (modulo m)," *Duke Math. J.*, vol. 18, no. 3, pp. 619–621, 1951.

[MR15] P. Montolio and J. Rifà, "Construction of Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes for each allowable value of the rank and dimension of the kernel," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1948–1958, 2015.

[Nec89] A. A. Nechaev, "The Kerdock code in a cyclic form," *Diskret. Mat.*, vol. 1, pp. 123–139, 1989. (English translation in *Discrete Math. Appl.*, vol. 1, pp. 365–384, 1991.

[PPV11] J. Pernas, J. Pujol, and M. Villanueva, "Classification of some families of quaternary Reed-Muller codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6043–6051, 2011.

[PPV12] J. Pernas, J. Pujol, and M. Villanueva, "Codes over $\mathbb{Z}_4$. A MAGMA package," version 1.4, Universitat Autònoma de Barcelona, 2012. `http://ccsg/uab.cat`.

[PPV14] J. Pernas, J. Pujol, and M. Villanueva, "Characterization of the automorphism group of quaternary linear Hadamard codes," *Des. Codes Cryptogr.*, vol. 70, nos. 1–2, pp. 105–115, 2014.

[Phe15] K. T. Phelps, "Enumeration of Kerdock codes of length 64," *Des. Codes Cryptogr.*, vol. 77, pp. 357–363, 2015.

[PR02] K. T. Phelps and J. Rifà, "On binary 1-perfect additive codes: some structural properties," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2587–2592, 2002.

[PRV06] K. T. Phelps, J. Rifà, and M. Villanueva, "On the additive ($\mathbb{Z}_4$-linear and non-$\mathbb{Z}_4$-linear) Hadamard codes: rank and kernel," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 316–319, 2006.

[Pra62] E. Prange, "The use of information sets in decoding cyclic codes," *IRE Transaction on Information Theory*, vol. 8, no. 5, pp. 5–9, 1962.

[PRS09] J. Pujol, J. Rifà, and F. I. Solov'eva, "Construction of $\mathbb{Z}_4$-linear Reed–Muller codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 1, pp. 99–104, 2009.

[PV14] J. Pujol and M. Villanueva, "Binary codes. A MAGMA package," version 2.0, Universitat Autònoma de Barcelona, 2014. `http://ccsg/uab.cat`.

[RSV08] J. Rifà, F. I. Solov'eva, and M. Villanueva, "On the intersection of $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1346–1356, 2008.

[RSV09] J. Rifà, F. I. Solov'eva, and M. Villanueva, "On the intersection of $\mathbb{Z}_2\mathbb{Z}_4$-additive Hadamard codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 4, pp. 1766–1774, 2009.

[RR13] A. del Rio and J. Rifà, "Families of Hadamard $\mathbb{Z}_2\mathbb{Z}_4Q_8$-codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 5140–5151, 2013.

[Sen09] P. Seneviratne, "Partial permutation decoding for the first-order Reed-Muller codes," *Discrete Math.*, vol. 309, no. 8, pp. 1967–1970, 2009.

[Var97] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1757–1773, 1997.

[VZP15] M. Villanueva, F. Zeng, and J. Pujol, "Efficient representation of binary nonlinear codes: constructions and minimum distance computation," *Des. Codes and Cryptogr.*, vol. 76, pp. 3–21, 2015.

[Wan97] Z.-X. Wan, *Quaternary Codes*, World Scientific, 1997.

[Wan03] Z.-X. Wan, *Lectures on Finite Fields and Galois Rings*, World Scientific, 2003.

[Whi06] G. White, "Enumeration-based Algorithms in Coding Theory," PhD Thesis, University of Sydney, 2006.

[Wol83] J. Wolfmann, "A permutation decoding of the (24,12,8) Golay code," *IEEE Trans. Inf. Theory*, vol. 29, no. 5, pp. 748–750, 1983.

[ZF15] H.-J. Zepernick and A. Finger, *Pseudo Random Signal Processing: Theory and Application*, John Wiley & Sons Ltd, 2015.

[Zim96] K.-H. Zimmermann, "Integral Hecke modules, integral generalized Reed-Muller codes, and linear codes," Tech. Rep. 3–96, Technische Universität Hamburg-Harburg, 1996.

Roland D. Barrolleta
Cerdanyola del Vallès, September 2016