UAB
Universitat Autònoma de Barcelona

Zps-Linear Codes.
Generalizations and permutation decoding

Adrián Torres Martín

UAB
Universitat Autònoma
de Barcelona

Departament d'Enginyeria de la Informació i de les Comunicacions

$\mathbb{Z}_{p^s}$-Linear Codes. Generalizations and permutation decoding

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

by Adrián Torres Martín

Cerdanyola del Vallès, September 2024

Advisor: Dr. Mercè Villanueva Gay
Professor at Universitat Autònoma de Barcelona

I certify that I have read this thesis entitled "$\mathbb{Z}_{p^s}$-linear codes. Generalizations and permutation decoding" 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 2024

Dr. Mercè Villanueva Gay
(Advisor)

Abstract

Linear codes over rings have gained much attention over the last 30 years following some results that connected linear codes over $\mathbb{Z}_4$, also called quaternary codes, with some important families of nonlinear binary codes. The main concepts describing quaternary codes can be generalized to linear codes over $\mathbb{Z}_{p^s}$. A linear code over $\mathbb{Z}_{p^s}$ of length n is a subgroup of $\mathbb{Z}_{p^s}^n$, and is also called a $\mathbb{Z}_{p^s}$-additive code. A $\mathbb{Z}_{p^s}$-linear code is a code over $\mathbb{Z}_p$, not necessarily linear, which is the generalized Gray map image of a $\mathbb{Z}_{p^s}$-additive code.

The study of $\mathbb{Z}_{p^s}$-linear codes and $\mathbb{Z}_{p^s}$-additive codes constitutes the central topic of this PhD thesis. In particular, we explore the permutation decoding method for $\mathbb{Z}_{p^s}$-linear codes, along with some necessary properties such that it becomes feasible and efficient. That is, we prove the existence of a systematic encoding and give a construction of suitable PD-sets. Obtaining good PD-sets requires a good understanding of the permutation automorphism group of a code, which is generally a difficult task. However, certain families of $\mathbb{Z}_{p^s}$-linear codes offer a more manageable representation of this group. We have been able to describe the permutation automorphism group of $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes, allowing us to find r-PD-sets for these codes.

At the same time, a MAGMA package has been developed, providing new functionality for $\mathbb{Z}_{p^s}$-linear and $\mathbb{Z}_{p^s}$-additive codes. The results obtained in this thesis are also implemented as functions in this package, namely: systematic encoding and permutation decoding for any $\mathbb{Z}_{p^s}$-linear code, a more efficient computation of a parity-check matrix for any $\mathbb{Z}_{p^s}$-additive code, and constructions of r-PD-sets for any $\mathbb{Z}_{p^s}$-linear generalized Hadamard code.

Moreover, special attention is given to the computation of the minimum homogeneous distance for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, which are subgroups of $\mathbb{Z}_2^{\alpha} \times \mathbb{Z}_4^{\beta}$.

Resum

Els codis lineals sobre anells han despertat un gran interès en els darrers 30 anys, després de la publicació d'uns resultats que connectaven els codis lineals sobre $\mathbb{Z}_4$, també anomenats codis quaternaris, amb algunes famílies importants de codis binaris no lineals. Els conceptes principals que descriuen els codis quaternaris es poden generalitzar als codis lineals sobre l'anell $\mathbb{Z}_{p^s}$. Un codi lineal sobre l'anell $\mathbb{Z}_{p^s}$ de longitud n és un subgrup de $\mathbb{Z}_p^n$, i s'anomena codi $\mathbb{Z}_{p^s}$-additiu. Per altra banda, un codi $\mathbb{Z}_{p^s}$-lineal és un codi sobre $\mathbb{Z}_p$, no necessàriament lineal, que és alhora la imatge d'un codi $\mathbb{Z}_{p^s}$-additiu a través d'una generalització del Gray map.

L'estudi dels codis $\mathbb{Z}_{p^s}$-lineals i $\mathbb{Z}_{p^s}$-additius representa el tema central d'aquesta tesi doctoral. En particular, explorem el mètode de descodificació per permutacions aplicat als codis $\mathbb{Z}_{p^s}$-lineals, juntament amb algunes propietats necessàries per aconseguir que el mètode sigui viable i eficient. Per exemple, demostrem l'existència d'una codificació sistemàtica i donem construccions per obtenir PD-sets adients. Per obtenir bons PD-set cal conèixer l'estructura del grup d'automorfismes per permutació d'un codi, cosa que no és fàcil en general. No obstant això, algunes famílies de codis $\mathbb{Z}_{p^s}$-lineals presenten un grup d'automorfismes amb una estructura més simple. Hem estat capaços de descriure el grup d'automorfismes per permutació dels codis $\mathbb{Z}_{p^s}$-lineals que pertanyen a la família de Hadamard generalitzats, permetent-nos trobar r-PD-sets per a aquests codis.

Simultàniament, hem estat desenvolupant un paquet de Magma que proporciona noves funcionalitats per als codis $\mathbb{Z}_{p^s}$-lineals i $\mathbb{Z}_{p^s}$-additius. Els resultats obtinguts en aquesta tesi també s'han implementat com a funcions

en aquest paquet, és a dir: codificació sistemática i descodificació per permutations per a qualsevol codi $\mathbb{Z}_{p^s}$-lineal, una computació més eficient de la matriu de control per a qualsevol codi $\mathbb{Z}_{p^s}$-additiu, i diverses construccions de r-PD-sets per a qualsevol codi $\mathbb{Z}_{p^s}$-lineal que pertanyi a la família de Hadamard generalitzats. També s'ha dedicat una atenció especial al càlcul de la distància mínima homogènia dels codis $\mathbb{Z}_2\mathbb{Z}_4$-additius, que són subgrups de $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$.

Resumen

Los códigos lineales sobre anillos han recibido mucha atención en los últimos 30 años, después de la publicación de unos resultados que relacionaban los códigos lineales sobre $\mathbb{Z}_4$, también llamados códigos cuaternarios, con algunas familias importantes de códigos binarios no lineales. Los conceptos principales que describen a los códigos cuaternarios se pueden generalizar a los códigos lineales sobre el anillo $\mathbb{Z}_{p^s}$. Un código lineal sobre $\mathbb{Z}_{p^s}$ de longitud n es un subgrupo de $\mathbb{Z}_p^n$, y también se le puede llamar código $\mathbb{Z}_{p^s}$-aditivo. Por otra parte, un código $\mathbb{Z}_{p^s}$-lineal es un código sobre $\mathbb{Z}_p$, no necesariamente lineal, que es la imagen de un código $\mathbb{Z}_{p^s}$-aditivo a través de una generalización del Gray map.

El estudio de los códigos $\mathbb{Z}_{p^s}$-lineales y $\mathbb{Z}_{p^s}$-aditivos constituye el tema central de esta tesis doctoral. En particular, exploramos el método de decodificación por permutaciones aplicado a los códigos $\mathbb{Z}_{p^s}$-lineales, junto a algunas propiedades necesarias para conseguir que el método sea factible y eficiente. Por ejemplo, demostramos la existencia de una codificación sistemática y mostramos cómo se pueden construir unos PD-sets adecuados. Para obtener buenos PD-sets se necesita conocer la estructura del grupo de automorfismos por permutación de un código, lo que suele ser difícil en general. Sin embargo, ciertas familias de códigos $\mathbb{Z}_{p^s}$-lineales ofrecen una representación más manejable de este grupo. Hemos sido capaces de describir el grupo de automorfismos por permutación de los códigos $\mathbb{Z}_{p^s}$-lineales pertenecientes a la familia de Hadamard generalizados, lo que nos ha permitido encontrar r-PD-sets para estos códigos.

Al mismo tiempo, hemos estado desarrollando un paquete de Magma,

proporcionando nuevas funcionalidades para los códigos $\mathbb{Z}_{p^s}$-lineales y $\mathbb{Z}_{p^s}$-aditivos. Los resultados obtenidos en esta tesis también han sido implementados como funciones en este paquete, es decir: codificación sistemática y decodificación por permutaciones para cualquier código $\mathbb{Z}_{p^s}$-lineal, una computación más eficiente de la matriz de control para cualquier código $\mathbb{Z}_{p^s}$-aditivo, y varias construcciones de r-PD-sets para cualquier código $\mathbb{Z}_{p^s}$-lineal perteneciente a la familia de Hadamard generalizados. También se ha prestado especial atención al cálculo de la distancia mínima homogénea de los códigos $\mathbb{Z}_2\mathbb{Z}_4$-aditivos, que son subgrupos de $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$.

Acknowledgements

First, I would like to thank my thesis supervisor, Mercè Villanueva, for the support and guidance she has provided me over these years. Her dedication and attention to detail have served as a model of how research should be conducted. She encouraged me to embark on this journey, and I am now truly glad that I did.

I am also deeply grateful to Cristina Fernández-Córdoba, Josep Rifà and Carlos Vela, with whom I have had the privilege of collaborating on various projects, for the invaluable knowledge I have learned from our discussions.

I would also like to extend a special gratitude to my fellow labmate, Dipak, for all the help he has provided. We have shared a common goal during these years, and his own work has been essential in guiding me through the different stages of this journey.

Lastly, I am truly thankful to my family and friends for their constant support throughout these years. In particular, I want to acknowledge my parents, without whom none of this would have been possible.

Contents

Chapter 1

Introduction

Any kind of communication through noisy channels can be represented by the schematic diagram shown in Figure 1.1, as proposed by Shannon in [Sha48]. A *channel* is the medium by which the information is transmitted. Then, any phenomenon that modifies this information, by altering or erasing some of its values, is called *noise*. In this context, *error-correcting codes* [Ham50, MS77] deal with the problem of detecting and correcting those possible errors introduced by the noise.

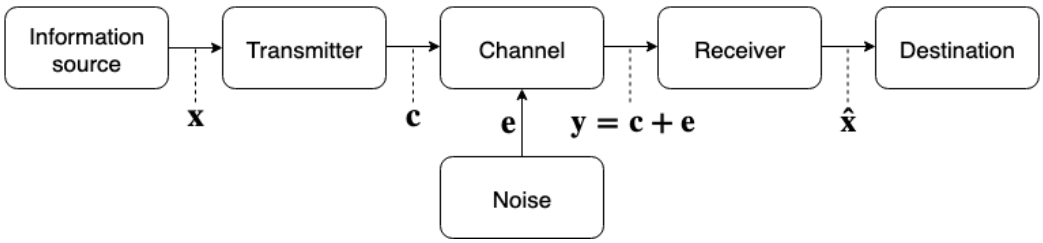

Figure 1.1: Schematic diagram describing a general transmission of information

In its most general sense, a *code* is a set of tuples, called *codewords*, defined over some set of symbols. In order to send some information, a *message*, through the channel, it must be transformed into a suitable signal. This process is called *encoding* and constitutes a bijection between all possible messages of a certain length and the codewords of a code. Let us assume that the message $\mathbf{x}$ is of the appropriate length, otherwise we can divide it into

several parts and treat them separately. The encoding assigns a codeword $\mathbf{c}$ to each message $\mathbf{x}$. After going through the channel, a different tuple of symbols $\mathbf{y} = \mathbf{c} + \mathbf{e}$ is received, where $\mathbf{e}$ represents the alteration induced by the noise. This tuple $\mathbf{y}$ may not coincide with any of the codewords, so an estimation $\hat{\mathbf{c}}$ is made hoping that $\hat{\mathbf{c}} = \mathbf{c}$. Since there is a bijection with the space of messages of a certain length, we obtain an estimation of the original message, $\hat{\mathbf{x}}$. The process of estimating a codeword from the received tuple, and obtaining the corresponding message, is called *decoding*.

Initially, error-correcting codes were defined over the binary field $\mathbb{Z}_2$ and later they were generalized to every finite field $\mathbb{Z}_p$, where p is a prime. Moreover, the most commonly used codes were linear codes, which provide an easier approach to encoding, decoding and storing. However, there are nonlinear codes which present interesting properties with respect to linear ones. For example, for the same length and minimum distance, there are nonlinear codes with more codewords than any linear code. This is the case of Kerdock [Ker72] and Preparata [Pre68] codes. In [HKC$^+$94], it was shown that these families of nonlinear binary codes are, in fact, the Gray map image of linear codes over the ring $\mathbb{Z}_4$. This finding prompted the research on linear codes over finite rings.

We say that a *linear code over* $\mathbb{Z}_4$ of length n is a subgroup of $\mathbb{Z}_4^n$. They are also called $\mathbb{Z}_4$-*additive codes*. Their image through the Gray map is a code over the finite field $\mathbb{Z}_2$, which may not be linear in the binary sense. We say that the Gray map image of a $\mathbb{Z}_4$-additive code is a $\mathbb{Z}_4$-*linear code*. The main concepts describing linear codes over finite fields can be generalized to linear codes over $\mathbb{Z}_4$ and even more general finite rings. For example, $\mathbb{Z}_{2^s}$-*additive* and $\mathbb{Z}_{p^s}$-*additive codes* of length n, with p prime, which are subgroups of $\mathbb{Z}_{2^s}^n$ and $\mathbb{Z}_{p^s}^n$, respectively, have been studied [Car98, GBL05, SSA18, TV03, SHQ$^+$21]. The image of these codes by appropriate generalizations of the Gray map produce codes over $\mathbb{Z}_2$ and $\mathbb{Z}_p$, called $\mathbb{Z}_{2^s}$-*linear* and $\mathbb{Z}_{p^s}$-*linear codes*, respectively. Furthermore, linear codes over mixed alphabets, such as $\mathbb{Z}_2\mathbb{Z}_4$-*additive codes* have been thoroughly studied [BFP$^+$10, BFP$^+$22b] and generalized to $\mathbb{Z}_2\mathbb{Z}_{2^s}$-*additive codes* [AS13], $\mathbb{Z}_p\mathbb{Z}_{p^2}$-*additive codes* [BFV22b, BFV23, LSW$^+$24], $\mathbb{Z}_p\mathbb{Z}_{p^s}$-*additive codes* [SWK19] and $\mathbb{Z}_{p^r}\mathbb{Z}_{p^s}$-additive codes

[AS15].

A *Hadamard code* is a binary code of length n, minimum distance $n/2$ and $2n$ codewords. If a Hadamard code is the image by the Gray map of a $\mathbb{Z}_4$-additive code then it is called a *$\mathbb{Z}_4$-linear Hadamard code (GH)*. Let H be a *generalized Hadamard matrix*, then a *generalized Hadamard code (GH)* is $C_H = \bigcup_{\alpha\in\mathbb{Z}_p}(F_H + \alpha\mathbf{1})$, where $F_H + \alpha\mathbf{1} = \{\mathbf{h} + \alpha\mathbf{1} : \mathbf{h} \in F_H\}$, F_H is the code consisting of the rows of H, and $\mathbf{1}$ denotes the all-one vector [DRV16]. If a GH code is also a $\mathbb{Z}_{p^s}$-linear code then it is called a *$\mathbb{Z}_{p^s}$-linear GH code*.

Most decoding methods are defined and developed for linear codes. However, the permutation decoding method, introduced by Prange [Pra62] and developed by MacWilliams [Mac64], can be easily generalized to nonlinear codes [BBFV15], as long as a systematic encoding is provided. The efficiency of this method strongly depends on the structure of the permutation automorphism group of the code. In particular, it relies on finding certain subsets of automorphisms, called *PD-sets*, or *r-PD-sets* if it corrects up to r errors. This is difficult for a general code, but it has been well studied for certain families, such as $\mathbb{Z}_4$-linear Hadamard codes [BV18].

Our efforts are focused on providing a systematic encoding and an alternative method of decoding for $\mathbb{Z}_{p^s}$-linear codes. First, we prove that $\mathbb{Z}_{p^s}$-linear codes are indeed systematic and we give a systematic encoding. This result enables the use of the permutation decoding method for any $\mathbb{Z}_{p^s}$-linear code. Then, the permutation automorphism group of $\mathbb{Z}_{p^s}$-linear GH codes is studied, showing its equivalence with a certain group of invertible matrices. We use this equivalence to construct r-PD-sets, with r up to an upper bound, for $\mathbb{Z}_{p^s}$-linear GH codes, which is vital in order to apply the method efficiently. Two constructions are presented. The first one generalizes the construction given in [BV18] for $\mathbb{Z}_4$-linear Hadamard codes, which is primarily designed to work for free codes (those that have a basis). The second construction improves on the first one by allowing larger values of r for non-free codes. This second construction is not only a new result for $\mathbb{Z}_{p^s}$-linear GH codes, but it also improves the previous results for $\mathbb{Z}_4$-linear Hadamard codes.

During the research leading to this thesis, we have developed a new MAGMA package that extends the functionality of the official distribution,

providing new features for $\mathbb{Z}_{p^s}$-additive and $\mathbb{Z}_{p^s}$-linear codes [FTV23]. In particular, many functions only available for $\mathbb{Z}_4$-linear codes have been generalized to $\mathbb{Z}_{p^s}$-linear codes.

This thesis is structured as follows:

- Chapter 2 contains a brief introduction to coding theory and describes the necessary concepts to understand the results and content that follows. We start by recalling the most basic concepts for linear codes, first over finite fields and then over $\mathbb{Z}_{p^s}$. Then, the permutation decoding method is described and the family of generalized Hadamard codes is introduced. Finally, the problem of computing the minimum weight of a linear code is presented, exploring some of the known computational methods.

- Chapter 3 proves that $\mathbb{Z}_{p^s}$-linear codes are systematic and a specific systematic encoding is given. Then, we show that the alternative permutation decoding method described in [BBFV15] for $\mathbb{Z}_4$-linear codes can also be applied to any $\mathbb{Z}_{p^s}$-linear code. An earlier version of these results, for $p = 2$, was presented at the *2020 Algebraic and Combinatorial Coding Theory* (ACCT 2020) conference and published in its proceedings [TV20]. Then, the general version, for any p prime, was published in the journal *IEEE Trans. Inform. Theory* [TV22a].

- Chapter 4 explores the permutation automorphism group of $\mathbb{Z}_{p^s}$-linear GH codes and describes two explicit constructions that produce r-PD-sets of size $r + 1$ for $\mathbb{Z}_{p^s}$-linear GH codes of type $(n; t_1, 0, \ldots, 0)$ and $(n; 1, 0, \ldots, 0, t_i, 0, \ldots, 0)$, with $t_1 \geq 2$ and $t_i \geq 1$. These constructions give r-PD-sets for any r up to the upper bounds $f_p^{t_1,0,\ldots,0}$ or $f_p^{1,0,\ldots,0,t_i,0,\ldots,0}$, respectively. A recursive construction is also given in order to obtain r-PD-sets for $\mathbb{Z}_{p^s}$-linear GH codes of any type. However, the values of r only reach up to $\tilde{f}_p^{t_1,\ldots,t_s} \leq f_p^{t_1,\ldots,t_s}$, that is, lower than the theoretical upper bound. Moreover, we give r-PD-sets obtained by a computational non-deterministic method with an r such that $\tilde{f}_p^{t_1,\ldots,t_s} \leq r \leq f_p^{t_1,\ldots,t_s}$, showing that we can get values of r closer

to the upper bound. An earlier version of these results, for $p = 2$ and $s = 3$, was presented at the *2022 IEEE Information Theory Workshop* (ITW 2022) and published in its proceedings [TV22b]. Then, the complete version was published in the journal *Finite Fields and Their Applications* [TV24a].

- Chapter 5 presents a new construction that improves the results of the previous chapter, by producing r-PD-sets of size $r + 1$ for $\mathbb{Z}_{p^s}$-linear GH codes of any type, for values of r larger than $\tilde{f}_p^{t_1,\ldots,t_s}$ and closer to the theoretical upper bound $f_p^{t_1,\ldots,t_s}$. An earlier version of these results, for $p = 2$ and $s = 3$, was introduced as part of a conference talk at *Rijeka Conference on Combinatorial Objects and their Applications* (RICCOTA 2023). The final version, for any p prime and any $s \geq 2$, has been accepted in the journal *IEEE Trans. Inform. Theory* [RTV24].

- Chapter 6 describes an efficient method for computing a parity-check matrix for any $\mathbb{Z}_{p^s}$-additive code. Two similar methods are introduced, one of which performs significantly better. The computational complexity of both is studied and compared. Finally, we also show experimental evidence that both methods improve the computational times of the current algorithm included in MAGMA for any linear code over a finite ring. The faster method was included in a new MAGMA package for $\mathbb{Z}_{p^s}$-additive codes [FTV23] (see Chapter 7). This work was presented as a conference paper at *2024 IEEE International Symposium on Information Theory* (ISIT 2024) [FTVV24a] and was published as a journal paper in *IEEE Trans. Inf. Theory* [FTVV24b].

- Chapter 7 describes the computational and software output produced in parallel during the course of this thesis. A new MAGMA package named *"Linear codes over the integer residue ring $\mathbb{Z}_{p^s}$. A MAGMA package"* [FTV23] has been developed. This package provides new functionality for $\mathbb{Z}_{p^s}$-linear codes and implements most of the results described in this thesis. We have also explored the computation of the minimum homogeneous weight of $\mathbb{Z}_{p^s}$-additive codes, generalizing

and adapting previous algorithms [Whi06, Zim96]. Our research on the minimum weight was also extended to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, which are subgroups of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$, that is, linear codes with some coordinates over $\mathbb{Z}_2$ and some over $\mathbb{Z}_4$. Specifically, we worked on the version 5.0 of the MAGMA package named *"$\mathbb{Z}_2\mathbb{Z}_4$-additive codes. A MAGMA package"* [BFG+22a], implementing new methods to obtain the minimum homogeneous weight of $\mathbb{Z}_2\mathbb{Z}_4$-additive codes and functions to construct a family of $\mathbb{Z}_2\mathbb{Z}_4$-additive Reed-Muller codes.

- Chapter 8 presents the conclusions of this thesis, summarizing all the results and proposing possible lines of further research.

In the following lines we give a summary of the research published during the PhD studies, which led to the content included in this thesis. In particular, we presented part of the research at the following conferences [TV20, TV22b, FTVV24a], appearing in their proceedings:

[TV20] A. Torres-Martín and M. Villanueva, "Systematic encoding for $\mathbb{Z}_{2^s}$-linear codes," in Proc. of *2020 Algebraic and Combinatorial Coding Theory (ACCT)*, Albena, Bulgaria (Virtual), 11–17 October, 2020, pp. 140–144, doi: 10.1109/ACCT51235.2020.9383384.

[TV22b] A. Torres-Martín and M. Villanueva, "Partial permutation decoding for $\mathbb{Z}_8$-linear Hadamard codes," in Proc. of *2022 IEEE Information Theory Workshop (ITW)*, Mumbai, India (Virtual), 1–2 November, 2022, pp. 113–118, doi: 10.1109/ITW54588.2022.9965899.

[FTVV24a] C. Fernández-Córdoba, A. Torres-Martín, C. Vela, and M. Villanueva, "Parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes: efficient computation," in Proc. of *2024 IEEE International Symposium on Information Theory (ISIT)*, Athens, Greece, 7-12 July, 2024, pp. 127–132, doi: 10.1109/ISIT57864.2024.10619267.

We also gave a talk at the following conference, which had no proceedings:

J. Rifà, A. Torres-Martín, and M. Villanueva, "Partial permutation decoding for $\mathbb{Z}_{p^s}$-linear Hadamard codes" (Conference session), *Rijeka Conference on Combinatorial Objects and their Applications* (RICCOTA), Rijeka, Croatia, 3–7 July, 2023. Abstract available in: `https://riccota2023.math.uniri.hr/`.

The results presented in the conferences were further expanded and each led to a published journal paper [TV22a, TV24a, RTV24, FTVV24b]:

[TV22a] A. Torres-Martín and M. Villanueva, "Systematic encoding and permutation decoding for $\mathbb{Z}_{p^s}$-linear codes," *IEEE Trans. Inf. Theory*, vol. 68(7), pp. 4435–4443, 2022, doi: 10.1109/TIT.2022.3157192.

[TV24a] A. Torres-Martín and M. Villanueva, "Partial permutation decoding and PD-sets for $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes," *Finite Fields Their Appl.*, vol. 93, 102316, 2024, doi: 10.1016/J.FFA.2023.102316.

[RTV24] J. Rifà, A. Torres-Martín, and M. Villanueva, "Improving explicit constructions of PD-sets for $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes," *IEEE Trans. Inf. Theory*, early access, 2024, doi: 10.1109/TIT.2024.3448230.

[FTVV24b] C. Fernández-Córdoba, A. Torres-Martín, C. Vela, and M. Villanueva, "Computing efficiently a parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes," *IEEE Trans. Inf. Theory*, early access, 2024, doi: 10.1109/TIT.2024.3370410.

Moreover, a new MAGMA package [FTV23] which provides new functionality for $\mathbb{Z}_{p^s}$-linear and $\mathbb{Z}_{p^s}$-additive codes has been developed. The results given in this thesis are implemented as functions in this new MAGMA package, namely: systematic encoding and permutation decoding for any $\mathbb{Z}_{p^s}$-linear code, a more efficient computation of the parity-check matrix for any $\mathbb{Z}_{p^s}$-additive code, and constructions of r-PD-sets for any $\mathbb{Z}_{p^s}$-linear GH code.

[FTV23] C. Fernández-Córdoba, A. Torres-Martín, and M. Villanueva, "Linear codes over the integer residue ring $\mathbb{Z}_{p^s}$. A MAGMA package", version 1.0, Universitat Autònoma de Barcelona, 2023. `https://ccsg.uab.cat`

Finally, we also contributed to the version 5.0 of a MAGMA package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, implementing new methods to compute the minimum homogeneous weight of these codes:

[BFG+22a] J. Borges, C. Fernández-Córdoba, B. Gastón, J. Pujol, J. Rifà, and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-additive codes," version 5.0, Universitat Autònoma de Barcelona, 2022. `https://ccsg.uab.cat`.

The work carried out during the preparation of this thesis was partially supported by the Spanish Ministerio de Ciencia e Innovación under Grants PID2019-104664GB-I00, PID2022-137924NB-I00, and RED2022-134306-T (AEI / 10.13039/501100011033) and by the Agència de Gestió d'Ajuts Universitaris i de Recerca, Generalitat de Catalunya grant 2021SGR00643.

Chapter 2

State of the art

In this chapter, we present a self-contained review of several concepts of coding theory, from the most basic properties of linear codes to more specific topics related to our thesis. In Section 2.1, we describe linear codes over finite fields and recall some basic concepts. In Section 2.2, linear codes over finite rings are described and most of the concepts introduced for linear codes over finite fields are generalized for finite rings. In Section 2.3, the permutation decoding method is explained, first for binary linear codes and then for linear codes over a finite field, even if they are not linear. In Section 2.4, generalized Hadamard (GH) codes are introduced, and a construction is given for linear GH codes over $\mathbb{Z}_{p^s}$. In Section 2.5, we address the topic of computing the minimum weight of a linear code over a finite field, and the Brouwer-Zimmermann method is explained.

The reader is referred to the following references, which have been consulted in the preparation of this chapter, for a more detailed study of these concepts. For a general introduction to coding theory [HP03, MS77]; for linear codes over finite rings [HKC$^+$94, Wan03, BFP$^+$22b]; for permutation decoding [Mac64, Pra62, BBFV15]; for GH codes [Kro01, BFV22a]; for the computation of the minimum weight [Whi06].

2.1 Linear codes over finite fields

In this thesis we focus on linear codes over the ring $\mathbb{Z}_{p^s}$, with p prime, and their image by a generalized Gray map, which are codes over the finite field $\mathbb{Z}_p$. For the rest of this section, we will define concepts for linear codes over $\mathbb{Z}_p$, with p prime, but they are completely applicable to any linear code over a finite field with $q = p^k$ elements, for an integer $k \geq 1$.

A *code* over $\mathbb{Z}_p$, with p prime, of length n is a nonempty subset of $\mathbb{Z}_p^n$, and its elements are called *codewords*. A *linear code* of *length* n and *dimension* k over $\mathbb{Z}_p$ is a vector subspace of dimension k of $\mathbb{Z}_p^n$, and is denoted by $C[n,k]_p$.

Let $\mathcal{S}_n$ be the *symmetric group of permutations* acting on $\mathbb{Z}_p^n$ by permuting the coordinates of each vector. That is, if $\pi \in \mathcal{S}_n$ and $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$, then $\pi(\mathbf{v}) = (v_{\pi^{-1}(1)}, \ldots, v_{\pi^{-1}(n)})$. The *permutation automorphism group* of a code $C[n,k]_p$ is denoted by $\mathrm{PAut}(C)$, that is, $\mathrm{PAut}(C) = \{\pi \in \mathcal{S}_n : \pi(C) = C\}$. Two codes over $\mathbb{Z}_p$ of length n, C_1 and C_2, are said to be *equivalent* if there is a vector $\mathbf{a} \in \mathbb{Z}_p^n$ and a permutation of coordinates $\pi \in \mathcal{S}_n$ such that $C_2 = \{\mathbf{a} + \pi(\mathbf{c}) : \mathbf{c} \in C_1\}$. If $\mathbf{a} = \mathbf{0}$, where $\mathbf{0}$ is the all-zero vector, then C_1 and C_2 are said to be *permutation equivalent*.

Let C be a code over $\mathbb{Z}_p$ of length n with p^k codewords. For a vector $\mathbf{u} \in \mathbb{Z}_p^n$ and a set $I \subseteq \{1, \ldots, n\}$, we denote the projection of $\mathbf{u}$ to the coordinates of I by $\mathbf{u}|_I$. We say that C is a *systematic code* if there is a set $I \subseteq \{1, \ldots, n\}$ of k coordinate positions such that $|C_I| = p^k$, where $C_I = \{\mathbf{u}|_I : \mathbf{u} \in C\}$. This set I is called an *information set* for C and $\{1, \ldots, n\} \setminus I$ is called a *redundancy set*. If C is a systematic code with information set I, then a *systematic encoding* with respect to I is an injective map $f : \mathbb{Z}_p^k \longrightarrow \mathbb{Z}_p^n$, such that for any information vector $\mathbf{a} \in \mathbb{Z}_p^k$, the corresponding codeword $f(\mathbf{a}) \in C$ satisfies $f(\mathbf{a})|_I = \mathbf{a}$. Note that nonlinear codes are not always systematic, for example, $C = \{(0,0,0), (1,0,0), (0,1,0), (0,0,1)\} \subseteq \mathbb{Z}_2^3$ is not [MS77, p. 303].

A *generator matrix* of a linear code $C[n,k]_p$ is a $k \times n$ matrix over $\mathbb{Z}_p$ whose rows form a basis for $C[n,k]_p$. It is well known that linear codes over $\mathbb{Z}_p$ are always systematic. Indeed, any set of k linearly independent

columns of a generator matrix G of a linear code C of dimension k forms an information set for C. Moreover, for every information set I of $C[n,k]_p$, there exists a generator matrix G such that the columns of G given by I form an identity matrix. Such a matrix is called a *systematic matrix* of C with respect to the information set I.

Every linear code $C[n,k]_p$ is permutation equivalent to a linear code $C'[n,k]_p$ which has a generator matrix in standard form. A generator matrix G is said to be in *standard form* if it can be written as

$$G = (\mathrm{Id}_k \ A)\,, \tag{2.1}$$

where Id_k is the identity matrix of size k, and A is a matrix of size $k \times (n-k)$ over $\mathbb{Z}_p$. If $C[n,k]_p$ has a generator matrix G in standard form, then the first k coordinate positions form an information set, and $f(\mathbf{a}) = \mathbf{a}G$ gives a systematic encoding, where $\mathbf{a} \in \mathbb{Z}_p^k$ is an information vector.

Example 1. *Consider the linear code $C[7,3]_2$ over $\mathbb{Z}_3$ defined by the following generator matrix:*

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

This matrix cannot be transformed into standard form just by linear combinations of rows. A permutation of coordinates is needed. In this case, permuting the third and fourth columns results in a matrix in standard form:

$$G' = \begin{pmatrix} 1 & 0 & 0 & 1 & 2 & 2 & 0 \\ 0 & 1 & 0 & 2 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

The code C' generated by matrix G' is permutation equivalent to C. □

Since a linear code $C[n,k]_p$ is a subspace of the vector space $\mathbb{Z}_p^n$, then it is the kernel of some linear transformation. Its matrix representation is called the *parity-check matrix* and can be defined as the matrix H of size $(n-k) \times n$ that satisfies $H\mathbf{c}^T = \mathbf{0}$ for any $\mathbf{c} \in C$. If $G = (\mathrm{Id}_k \ A)$ is a generator matrix

in standard form for $C[n,k]_p$, then

$$H = \left(-A^T \ \mathrm{Id}_{n-k}\right) \tag{2.2}$$

is a parity-check matrix for $C[n,k]_p$.

A parity-check matrix of $C[n,k]_p$ can also be seen as a generator matrix for a different linear code $C'[n,n-k]_p$, which is directly related to $C[n,k]_p$. Let $\mathbf{x} = (x_1,\ldots,x_n)$ and $\mathbf{y} = (y_1,\ldots,y_n)$ be two vectors of $\mathbb{Z}_p^n$. Then its inner product is $\mathbf{x}\cdot\mathbf{y} = \sum_{i=1}^n x_i y_i$. The *dual code* of $C[n,k]_p$ is denoted by $C^\perp$ and is defined as

$$C^\perp = \{\mathbf{v} \in \mathbb{Z}_p^n \,:\, \mathbf{v}\cdot\mathbf{c} = 0,\ \forall \mathbf{c} \in C\}. \tag{2.3}$$

The dual code of C is generated by a parity-check matrix H of C. That is, a parity matrix of C is a generator matrix of $C^\perp$, and vice versa.

Example 2. *Consider the code C' given in Example 1, which has a generator matrix G' in standard form. The matrix*

$$H' = \begin{pmatrix} 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

is a parity-check matrix for C'. Undoing the column permutation, that is, swapping again the third and fourth columns of H', we obtain the following parity-check matrix for C:

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

The *Hamming weight* of a vector $\mathbf{u} \in \mathbb{Z}_p^n$, denoted by $\mathrm{wt}_H(\mathbf{u})$, is the number of non-zero coordinates of $\mathbf{u}$. The *Hamming distance* of two vectors

$\mathbf{u}, \mathbf{v} \in \mathbb{Z}_p^n$, denoted by $d_H(\mathbf{u}, \mathbf{v})$, is the number of coordinates in which they differ. Note that $d_H(\mathbf{u}, \mathbf{v}) = \mathrm{wt}_H(\mathbf{v} - \mathbf{u})$. The *minimum distance* of a code C over $\mathbb{Z}_p$ is $d(C) = \min\{d_H(\mathbf{u}, \mathbf{v}) : \mathbf{u}, \mathbf{v} \in C, \mathbf{u} \neq \mathbf{v}\}$.

Given a code C with minimum Hamming distance d, the *error correcting capability* is

$$t = \lfloor \frac{d-1}{2} \rfloor, \tag{2.4}$$

where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. Alternatively, C is said to be a *t-error-correcting code*. That is, if a vector $\mathbf{y}$ is received where less than t errors have occurred, then there is a unique closest codeword in C to $\mathbf{y}$.

Given a linear code $C[n, k]_q$, its *Hamming weight enumerator* is the polynomial

$$\mathrm{Ham}_C(x, y) = \sum_{i=0}^{n} A_i x^i y^{n-i} = \sum_{\mathbf{c} \in C} x^{\mathrm{wt}(c)} y^{n - \mathrm{wt}(c)}, \tag{2.5}$$

where A_i is the number of codewords of weight i for $0 \leq i \leq n$. The Hamming weight enumerators of $C[n, k]_q$ and $C^\perp$ are connected through the *MacWilliams Identity*:

$$\mathrm{Ham}_{C^\perp}(x, y) = \frac{1}{|C|} \mathrm{Ham}_C(y - x, y + (q-1)x). \tag{2.6}$$

Example 3. *Consider the linear code $C[4, 2]_3$ given by the generator matrix*

$$G = \begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 2 & 1 \end{pmatrix}. \tag{2.7}$$

The code C has 9 codewords: one of weight 0, two of weight 2, four of weight 3, and two of weight 4. Its Hamming weight enumerator is $\mathrm{Ham}_C(x, y) = y^4 + 2x^2y^2 + 4x^3y + 2x^4$. The dual code $C^\perp$ of C is generated by the matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{pmatrix}. \tag{2.8}$$

The code $C^\perp$ has 9 codewords: one of weight 0, two of weight 2, four of weight

3, *and two of weight* 4. *Its Hamming weight enumerator is* $\mathrm{Ham}_{C^\perp}(x,y) = y^4 + 2x^2y^2 + 4x^3y + 2x^4$. *Using the MacWilliams Identity, we can verify that both Hamming weight enumerators are equal*

$$\begin{aligned}\mathrm{Ham}_{C^\perp}(x,y) &= \frac{1}{9}\mathrm{Ham}_C(y-x, y+2x)\\ &= \frac{1}{9}\left((y+2x)^4 + 2(y-x)^2(y+2x)^2 + 4(y-x)^3(y+2x)\right.\\ &\qquad\qquad\left. +2(y-x)^4\right)\\ &= \frac{1}{9}\left(9y^4 + 18x^2y^2 + 36x^3y + 18x^4\right)\\ &= y^4 + 2x^2y^2 + 4x^3y + 2x^4.\end{aligned}$$

2.2 Linear codes over $\mathbb{Z}_{p^s}$

Let $\mathbb{Z}_{p^s}$ be the ring of integers modulo p^s with $s \geq 1$ and p prime, and $\mathbb{Z}_{p^s}^n$ be the set of n-tuples over $\mathbb{Z}_{p^s}$. In this thesis, the elements of $\mathbb{Z}_{p^s}^n$ are also called vectors over $\mathbb{Z}_{p^s}$ of length n. A *code* over $\mathbb{Z}_p$ of length n is a nonempty subset of $\mathbb{Z}_p^n$, and it is *linear* if it is a subspace of $\mathbb{Z}_p^n$. A nonempty subset of $\mathbb{Z}_{p^s}^n$ is a $\mathbb{Z}_{p^s}$*-additive code* if it is a subgroup of $\mathbb{Z}_{p^s}^n$. Note that, when $p = 2$ and $s = 1$, a $\mathbb{Z}_{p^s}$-additive code is a binary linear code and, when $p = 2$ and $s = 2$, it is a quaternary linear code or a linear code over $\mathbb{Z}_4$. The *order* of a vector $\mathbf{u}$ over $\mathbb{Z}_{p^s}$, denoted by $\mathrm{ord}(\mathbf{u})$, is the smallest positive integer m such that $m\mathbf{u} = \mathbf{0}$.

Recall that two codes over $\mathbb{Z}_p$ of length n, C_1 and C_2, are said to be permutation equivalent if there is a permutation of coordinates $\pi \in \mathcal{S}_n$ such that $C_2 = \pi(C_1)$. In a similar way, two $\mathbb{Z}_{p^s}$-additive codes of length n, $\mathcal{C}_1$ and $\mathcal{C}_2$, are said to be *permutation equivalent* if they differ only by a permutation of coordinates, that is, if there is a permutation of coordinates $\pi \in \mathcal{S}_n$ such that $\mathcal{C}_2 = \pi(\mathcal{C}_1)$.

Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of length n. Since $\mathcal{C}$ is a subgroup of $\mathbb{Z}_{p^s}^n$, it is isomorphic to an abelian group $\mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_{p^2}^{t_{s-1}} \times \mathbb{Z}_p^{t_s}$, and we say that $\mathcal{C}$ is of type $(p^s)^{t_1}(p^{s-1})^{t_2}\ldots p^{t_s}$ or $(t_1, \ldots, t_s)$ briefly. Sometimes, the length is also included in the type, that is, we can also say that $\mathcal{C}$ is of

type $(n;\, t_1, \ldots, t_s)$. It is clear that a $\mathbb{Z}_{p^s}$-additive code of type $(t_1, \ldots, t_s)$ has $p^{st_1+(s-1)t_2+\cdots+t_s}$ codewords.

In general, $\mathbb{Z}_{p^s}$-additive codes are not *free* as submodules of $\mathbb{Z}_{p^s}^n$ (they are if $t_2 = \cdots = t_s = 0$), which means that they usually do not have a basis in the module sense. However, any codeword of a $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$ can be expressed uniquely in the form

$$\sum_{j=1}^{s} \sum_{i=1}^{t_j} \lambda_i^{(j)} \mathbf{u}_i^{(j)}, \tag{2.9}$$

where $\lambda_i^{(j)} \in \mathbb{Z}_{p^{s-j+1}}$ and $\mathbf{u}_i^{(j)}$ are codewords of $\mathcal{C}$ of order p^{s-j+1}. These codewords $\mathbf{u}_i^{(j)}$ form a *generator matrix* of $\mathcal{C}$ having minimum number of rows, that is, $t_1 + \cdots + t_s$ rows. Moreover, $\mathcal{C}$ is permutation equivalent to a $\mathbb{Z}_{p^s}$-additive code with generator matrix of the form

$$\mathcal{G} = \begin{pmatrix} \mathrm{Id}_{t_1} & A_{0,1} & A_{0,2} & A_{0,3} & \cdots & \cdots & A_{0,s} \\ \mathbf{0} & p\mathrm{Id}_{t_2} & pA_{1,2} & pA_{1,3} & \cdots & \cdots & pA_{1,s} \\ \mathbf{0} & \mathbf{0} & p^2\mathrm{Id}_{t_3} & p^2A_{2,3} & \cdots & \cdots & p^2A_{2,s} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & p^{s-1}\mathrm{Id}_{t_s} & p^{s-1}A_{s-1,s} \end{pmatrix}, \tag{2.10}$$

where $A_{i,j}$ are matrices over $\mathbb{Z}_{p^s}$, for $0 \leq i \leq s-1$ and $1 \leq j \leq s$, and $\mathbf{0}$ is the all-zero matrix. If the generator matrix of a $\mathbb{Z}_{p^s}$-additive code has this form, it is said to be in *standard form*.

Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code with generator matrix $\mathcal{G}$ in standard form (2.10). Any information vector of $\mathcal{C}$ is of the form $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s) = (\iota_1(\mathbf{x}'_1), \ldots, \iota_s(\mathbf{x}'_s)) \in \mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$, where $\mathbf{x}'_k \in \mathbb{Z}_{p^{s-k+1}}^{t_k}$ and $\iota_k(a) = a$ is the *identity map* from $\mathbb{Z}_{p^{s-k+1}}$ to $\mathbb{Z}_{p^s}$, for $1 \leq k \leq s$. Then, an encoding from the

set of all possible information vectors to $\mathcal{C}$ can be carried out by using the product $\mathbf{x}\mathcal{G} = (\mathbf{x}_1, \ldots, \mathbf{x}_s)\mathcal{G}$. However, this encoding is not always systematic in the first coordinates, since the matrices $A_{i,j}$ in (2.10), with $1 \leq j \leq s-1$, may not be all zero matrices.

Example 4. *Consider the $\mathbb{Z}_{27}$-additive code $\mathcal{C}$ of length 5 and type $(1,1,1)$ generated by the matrix*

$$\mathcal{G} = \begin{pmatrix} 1 & 2 & 7 & 0 & 9 \\ 0 & 3 & 3 & 18 & 21 \\ 0 & 0 & 9 & 18 & 0 \end{pmatrix}. \tag{2.11}$$

Let $\mathbf{x} = (17, 8, 1)$ be an information vector, then the encoding of $\mathbf{x}$ is carried out as follows: $\mathbf{x}\mathcal{G} = (17, 11, 4)\mathcal{G} = (17, 25, 11, 18, 9)$.

The *inner product* of $\mathbf{u} = (u_1, \ldots, u_n)$ and $\mathbf{v} = (v_1, \ldots, v_n)$ in $\mathbb{Z}_{p^s}^n$ is defined as $\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i \in \mathbb{Z}_{p^s}$. Then, if $\mathcal{C}$ is a $\mathbb{Z}_{p^s}$-additive code of length n, its *dual code* is

$$\mathcal{C}^{\perp} = \{\mathbf{v} \in \mathbb{Z}_{p^s}^n : \mathbf{u} \cdot \mathbf{v} = 0 \text{ for all } \mathbf{u} \in \mathcal{C}\}.$$

In [CS95], it is proved that if $\mathcal{C}$ is a $\mathbb{Z}_{p^s}$-additive code of type $(n; t_1, \ldots, t_s)$, then $\mathcal{C}^{\perp}$ is a $\mathbb{Z}_{p^s}$-additive code of type $(n; n-t, t_s, t_{s-1}, \ldots, t_2)$, where $t = \sum_{i=1}^{s} t_i$.

Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code with generator matrix $\mathcal{G}$. A matrix $\mathcal{H}$ is a *parity-check matrix* of $\mathcal{C}$ if it is a generator matrix of its dual code $\mathcal{C}^{\perp}$. In this sense, the code $\mathcal{C}$ can be generated from $\mathcal{H}$ by computing all the orthogonal vectors to it, that is,

$$\mathcal{C} = \{\mathbf{u} \in \mathbb{Z}_{p^s}^n : \mathcal{H}\mathbf{u}^T = \mathbf{0}\}.$$

A parity-check matrix $\mathcal{H}$ holds that $\mathcal{G}\mathcal{H}^T = (\mathbf{0})$, which is a crucial property that plays the main role in syndrome decoding. It can be used to correct errors but also to correct erasures, since it provides a linear system of equations that can be solved in order to recover the sent information.

In this thesis, we consider the following metric, the *homogeneous weight*, defined in [CH97], and also used in [GS99, SWK19]:

$$\mathrm{wt}^*(x) = \begin{cases} 0, & \text{if } x = 0, \\ p^{s-1}, & \text{if } x \in p^{s-1}\mathbb{Z}_{p^s} \backslash \{0\}, \\ (p-1)p^{s-2}, & \text{otherwise.} \end{cases} \tag{2.12}$$

The *weight* of a vector $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{Z}_{p^s}^n$ is $\mathrm{wt}^*(\mathbf{u}) = \sum_{j=1}^n \mathrm{wt}^*(u_j) \in \mathbb{Z}_{p^s}$; and the *distance* between two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_{p^s}^n$ is $d^*(\mathbf{u}, \mathbf{v}) = \mathrm{wt}^*(\mathbf{u}-\mathbf{v})$. The *minimum distance* of a $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$, or in general a code over $\mathbb{Z}_{p^s}$, is $d^*(\mathcal{C}) = \min\{d^*(\mathbf{u}, \mathbf{v}) : \mathbf{u}, \mathbf{v} \in \mathcal{C}, \mathbf{u} \neq \mathbf{v}\}$. If C is a linear code over $\mathbb{Z}_p$, then $d_H(C)$ coincides with the minimum weight of C, $\mathrm{wt}_H(C) = \min\{\mathrm{wt}_H(\mathbf{u}) : \mathbf{u} \in C, \mathbf{u} \neq \mathbf{0}\}$ [MS77, p. 10]. Similarly, $d^*(\mathcal{C})$ coincides with $\mathrm{wt}^*(\mathcal{C}) = \min\{\mathrm{wt}^*(\mathbf{u}) : \mathbf{u} \in \mathcal{C}, \mathbf{u} \neq \mathbf{0}\}$ if $\mathcal{C}$ is a $\mathbb{Z}_{p^s}$-additive code.

In [HKC+94, Nec91], a *Gray map* from $\mathbb{Z}_4$ to $\mathbb{Z}_2^2$ is defined as $\phi(0) = (0, 0)$, $\phi(1) = (0, 1)$, $\phi(2) = (1, 1)$ and $\phi(3) = (1, 0)$. There exist different generalizations of this Gray map, which go from $\mathbb{Z}_{2^s}$ to $\mathbb{Z}_2^{2^{s-1}}$ [Car98, DF11, Kro07]. The one given by Krotov in [Kro07] is defined in terms of the codewords of a Hadamard code, and the one given by Carlet in [Car98] is a particular case of Krotov's one satisfying $\sum \lambda_i \phi_s(2^i) = \phi_s(\sum \lambda_i 2^i)$ [FVV19]. In this thesis, we consider a generalization of Carlet's Gray map, from $\mathbb{Z}_{p^s}$ to $\mathbb{Z}_p^{p^{s-1}}$, denoted by ϕ_s and defined as follows:

$$\phi_s(u) = (u_{s-1}, \ldots, u_{s-1}) + (u_0, \ldots, u_{s-2})Y_{s-1}, \tag{2.13}$$

where $u \in \mathbb{Z}_{p^s}$, $[u_0, u_1, \ldots, u_{s-1}]_p$ is the p-ary expansion of u, that is $u = \sum_{i=0}^{s-1} p^i u_i$ ($u_i \in \mathbb{Z}_p$), and Y_{s-1} is a matrix of size $(s-1) \times p^{s-1}$ whose columns are all the distinct elements from $\mathbb{Z}_p^{s-1}$. Note that the rows of Y_{s-1} form a basis of a first order Reed-Muller code after adding the all-one row. If $s = 1$, then ϕ_s is the identity map. In order to simplify the notation, we write ϕ instead of ϕ_s, when s is clear from the context. Then, we define $\Phi : \mathbb{Z}_{p^s}^n \to \mathbb{Z}_p^{np^{s-1}}$ as the component-wise extension of ϕ.

Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of length n and type $(t_1, t_2, \ldots, t_s)$. We

say that its Gray map image, $C = \Phi(\mathcal{C})$, is a $\mathbb{Z}_{p^s}$*-linear code* of length $p^{s-1}n$ of type $(t_1, t_2, \ldots, t_s)$. It is known that the Gray map ϕ is an isometric embedding from $(\mathbb{Z}_{p^s}, d^*)$ into $(\mathbb{Z}_p^{p^{s-1}}, d_H)$ [GS99, SWK19]. Moreover, the $\mathbb{Z}_{p^s}$-linear codes obtained from this Gray map are distance invariant, that is, the Hamming weight distribution is invariant under translation by a codeword. This is proven directly using the fact that $d_H(\phi(u), \phi(v)) = \mathrm{wt}_H(\phi(u - v))$ for any $u, v \in \mathbb{Z}_{p^s}$ [Car98, GS99].

A $\mathbb{Z}_{p^s}$-linear code with $s \geq 2$ may not be linear as a code over $\mathbb{Z}_p$. It is known that a $\mathbb{Z}_4$-linear code $C = \Phi_2(\mathcal{C})$ is a binary linear code if and only if $2\mathbf{u} * \mathbf{v} \in \mathcal{C}$ for all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$, where $*$ denotes the component-wise product of two vectors over $\mathbb{Z}_4$ [HKC$^+$94]. There is also a similar result for $\mathbb{Z}_{p^s}$-linear codes [TV03, BFV22a]. Let $[u_0, \ldots, u_{s-1}]_p$ and $[v_0, \ldots, v_{s-1}]_p$ be the p-ary expansions of u and v from $\mathbb{Z}_{p^s}$, respectively. We define the operation "$\odot_p$" between two elements $u, v \in \mathbb{Z}_{p^s}$ as $u \odot_p v = \sum_{i=0}^{s-1} t_i p^i$, where

$$t_i = \begin{cases} 1 & \text{if } u_i + v_i \geq p, \\ 0 & \text{otherwise.} \end{cases}$$

Note that if $p = 2$, then $u \odot_2 v = \sum_{i=0}^{s-1} u_i v_i 2^i$ as it is defined in [TV03]. The component-wise extension of this operation is also denoted by $\odot_p$. Note that if $p = 2$ and $s = 2$, then $2\mathbf{u} * \mathbf{v} = 2(\mathbf{u} \odot_2 \mathbf{v})$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_4^n$. Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code. Then, for $s \geq 2$, we have that the $\mathbb{Z}_{p^s}$-linear code $C = \Phi(\mathcal{C})$ is linear over $\mathbb{Z}_p$ if and only if $p(\mathbf{u} \odot_p \mathbf{v}) \in \mathcal{C}$ for all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ [TV03, BFV22a].

2.3 Permutation decoding

Permutation decoding is a technique, introduced by Prange [Pra62] and developed by MacWilliams [Mac64] for linear codes, that involves finding a subset of the permutation automorphism group of a code in order to assist in decoding. In [BBFV15], a new permutation decoding method for $\mathbb{Z}_4$-linear codes (not necessarily linear) was introduced, based on having a systematic

encoding for these codes. Actually, it is also proved that this method can be used for any nonlinear binary code, as long as it has a systematic encoding.

The idea behind the permutation decoding technique is to move all errors in a received vector out of the information positions by using a permutation that preserves the code. Let C be a t-error-correcting code over $\mathbb{Z}_p$. Then, it is necessary to find a subset $S \subseteq \mathrm{PAut}(C)$, with respect to an information set for C, such that every r-set of coordinate positions is moved out of the information coordinates by at least one element in S, where $1 \leq r \leq t$. The set S is called an *r-PD-set* and, if $r = t$, it is called a *PD-set*. The efficiency of the permutation decoding method depends on the size of the r-PD-set $S \subseteq \mathrm{PAut}(C)$, since it needs to find the suitable permutation in S, for each received vector.

An r-PD-set ensures that if an error of weight $r \leq t$ is produced, then we can move the error coordinates out of the information set via one of its elements. That is, the information positions of the received vector are correct. The following theorem gives a necessary and sufficient condition to verify that the information coordinates are correct if the code is linear.

Theorem 5 ([MS77])**.** *Let C be a t-error-correcting linear code over $\mathbb{Z}_p$ with information set I and parity-check matrix H in standard form. Let $\mathbf{y} = \mathbf{x}+\mathbf{e}$, where $\mathbf{x} \in C$ and $\mathbf{e}$ is a vector of weight $\mathrm{wt}_H(\mathbf{e}) \leq t$. Then the information coordinates of $\mathbf{y}$ are correct if and only if $\mathrm{wt}_H(H\mathbf{y}^T) \leq t$.*

Let $C[n,k]_q$ be a linear code of error-correcting capability t, parity-check matrix H and information set I. Assume we have a PD-set, $S \subseteq \mathrm{PAut}(C)$, with respect to the information set I. Theorem 5 induces the following algorithm:

1. For a received vector $\mathbf{y} = \mathbf{x}+\mathbf{e}$, if $\mathrm{wt}_H(H\mathbf{y}^T) \leq t$, then the information coordinates of $\mathbf{y}$ are correct and we can decode y by $\mathbf{y}_I = \mathbf{x}_I$.

2. Else, we search for a permutation $\pi \in S$ such that $\mathrm{wt}_H(H\pi(\mathbf{y})^T) \leq t$. If there is no such π, we conclude that more than t errors have occurred.

3. If we have found such π, then the information positions of $\pi(\mathbf{y})$ are

correct. We take $\mathbf{x}'$ as the unique codeword such that $\mathbf{x}'_I = \pi(\mathbf{y})_I$, and decode y by $\pi^{-1}(\mathbf{x}')_I$.

Theorem 5 is only valid when it is applied to linear codes, therefore, the standard permutation decoding algorithm cannot be applied to $\mathbb{Z}_4$-linear codes in general. In [BBFV15], an alternative permutation decoding algorithm is presented, which enables permutation decoding for any binary systematic code, as long as a systematic encoding is known. This alternative algorithm is based on the following theorem.

Theorem 6 ([BBFV15])**.** *Let C be a binary systematic t-error-correcting code of length n with information set I and let f be a systematic encoding with respect to I. Suppose that $\mathbf{y} = \mathbf{x} + \mathbf{e}$ is a received vector, where $\mathbf{x} \in C$ and $\mathrm{wt}_H(\mathbf{e}) \leq t$. Then the information coordinates of $\mathbf{y}$ are correct if and only if $\mathrm{wt}_H(\mathbf{y} + f(\mathbf{y}_I)) \leq t$.*

Consider a binary systematic code C with information set I and a systematic encoding f with respect to I. Let S be a PD-set with respect to I, then the alternative permutation decoding algorithm is carried out as follows:

1. For a received vector $\mathbf{y} = \mathbf{x} + \mathbf{e}$, if $\mathrm{wt}_H(\mathbf{y} + f(\mathbf{y}_I)) \leq t$, then the information coordinates of $\mathbf{y}$ are correct, and we can decode by $\mathbf{y}_I{=}\mathbf{x}_I$.

2. Else, we search for a permutation $\pi \in S$ such that

$$\mathrm{wt}_H(\pi(\mathbf{y}) + f(\pi(\mathbf{y})_I)) \leq t.$$

   If there is no such π, we conclude that more than t errors have occurred.

3. If we find such π, then the decoded vector is $\mathbf{x}'_I$, where

$$\mathbf{x}' = \pi^{-1}(f(\pi(\mathbf{y})_I)). \tag{2.14}$$

Example 7. *Consider the $\mathbb{Z}_4$-additive code $\mathcal{C}$ with generator matrix*

$$\mathcal{G} = \begin{pmatrix} 1 & 0 & 1 & 2 & 3 & 0 & 3 & 2 \\ 0 & 1 & 0 & 3 & 2 & 1 & 2 & 3 \\ 0 & 0 & 2 & 0 & 0 & 2 & 2 & 2 \end{pmatrix}. \tag{2.15}$$

Let $C = \Phi(\mathcal{C})$ be the corresponding $\mathbb{Z}_4$-linear code. The code C is of type $4^2 2^1$ and has length 16. It easy to check that $I = \{1, 2, 3, 4, 5\}$ is an information set for C [BBFV15]. Note that C is not linear as a binary code, but it is systematic (a systematic encoding for any $\mathbb{Z}_4$-linear code is given in [BBFV15]). Moreover, it has minimum distance $d = 8$, therefore its error-correcting capability is $t = 3$. Consider the following 7 permutations of S_{16}:

$$\begin{aligned}
\vartheta_1 &= (1,7)(2,8)(3,9)(4,10)(5,15)(6,16)(11,13)(12,14)\\
\vartheta_2 &= (1,11,5,3)(2,12,6,4)(9,15,13,7)(10,16,14,8)\\
\vartheta_3 &= (1,13)(2,14)(3,7)(4,8)(5,9)(6,10)(11,15)(12,16)\\
\vartheta_4 &= (1,3)(2,4)(9,7)(10,8)(5,11)(6,12)(13,15)(14,16)\\
\vartheta_5 &= (1,15,5,7)(2,16,6,8)(3,9,11,13)(4,10,12,14)\\
\vartheta_6 &= (1,5)(2,6)(3,11)(4,12)(9,13)(10,14)(7,15)(8,16)\\
\vartheta_7 &= (1,9)(2,10)(3,15)(4,16)(7,11)(8,12)(5,13)(6,14).
\end{aligned}$$

Using MAGMA*, it is easy to check that the set of permutations $S = \{Id, \vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5, \vartheta_6, \vartheta_7\}$, where Id is the identity permutation, is a PD-set with respect to the information set I.*

For example, let us consider an information vector $\mathbf{a} = (1, 0, 1, 1, 1) \in \mathbb{Z}_2^5$. Then, using the systematic encoding f for $I = \{1, 2, 3, 4, 5\}$ given in [BBFV15], the corresponding codeword is

$$\begin{aligned}
\mathbf{x} = f(\mathbf{a}) &= \Phi(\sigma(\Phi^{-1}(\mathbf{a}))) = \Phi((3, 2, 1+1)\mathcal{G})\\
&= \Phi((3, 2, 3, 0, 1, 2, 1, 0)) = (1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0),
\end{aligned}$$

where σ is as defined in [BBFV15]. That is, for a code of type (t_1, t_2), given a vector $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{Z}_4^{t_1} \times \mathbb{Z}_2^{t_2}$, then

$$\sigma(\mathbf{u}) = (\mathbf{u}_1, \mathbf{u}_2 + \Phi^{(1)}(\mathbf{u}_1 A_{0,1})),$$

where $A_{0,1}$ is as in (2.10) and $\Phi^{(1)}$ is the coordinate-wise extension of $\phi^{(1)}$, which denotes the first component of the Gray map.

Now, suppose that the received vector is $\mathbf{y} = \mathbf{x} + \mathbf{e}$*, where the error vector is* $\mathbf{e} = (1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0)$*. That is,*

$$\mathbf{y} = (0,0,1,0,0,0,0,0,0,1,1,1,0,1,0,0). \tag{2.16}$$

Projecting to the information set I*, we have* $\mathbf{y}_I = (0,0,1,0,0)$ *and*

$$f(\mathbf{y}_I) = (0,0,1,0,0,0,0,1,1,1,1,0,1,1,0,1). \tag{2.17}$$

Note that $\mathrm{wt}_H(\mathbf{y} + f(\mathbf{y}_I)) = 5 > 3 = t$*, therefore, by Theorem 6, the information coordinates of* $\mathbf{y}$ *have errors, as we already knew. However, let us consider the permutation* ϑ_1 *from the set* S*. We have*

$$\vartheta_1(\mathbf{y}) = (0,0,0,1,0,0,0,0,1,0,0,1,1,1,0,0) \tag{2.18}$$

and $\vartheta_1(\mathbf{y})_I = (0,0,0,1,0)$*. Encoding this vector we obtain the codeword*

$$f(\vartheta_1(\mathbf{y})_I) = (0,0,0,1,0,0,1,0,1,1,0,1,1,1,1,0). \tag{2.19}$$

We have that $\mathrm{wt}(\vartheta_1(\mathbf{y}) + f(\vartheta(\mathbf{y})_I) = 3 \leq t = 3$*. Therefore, the information coordinates of* $\vartheta_1(\mathbf{y})$ *are correct. Hence, we decode* $\mathbf{y}$ *as*

$$\vartheta_1^{-1}(f(\vartheta_1(\mathbf{y})_I)) = (1,0,1,1,1,0,0,0,0,1,1,1,0,1,0,0) = \mathbf{x} \tag{2.20}$$

and the information vector is $\mathbf{x}_I = (1,0,1,1,1)$*.* □

In Chapter 3, we show that Theorem 6 can be generalized to any systematic code over $\mathbb{Z}_p$, with p prime, linear or not, as long as a systematic encoding is provided. Moreover, we also prove that all $\mathbb{Z}_{p^s}$-linear codes are systematic, and a systematic encoding is given. Therefore, the alternative permutation decoding method can also be used for any $\mathbb{Z}_{p^s}$-linear code.

2.4 Generalized Hadamard codes

A *generalized Hadamard (GH) matrix* $H(p,\lambda)=(h_{ij})$ of order $N=p\lambda$ over $\mathbb{Z}_p$ is a $p\lambda\times p\lambda$ matrix with entries in $\mathbb{Z}_p$ with the property that, for every i,j, $1\le i<j\le p\lambda$, each of the multisets $\{h_{ik}-h_{jk}:1\le k\le p\lambda\}$ contains every element of $\mathbb{Z}_p$ exactly λ times [Jun79]. An ordinary Hadamard matrix of order 4μ corresponds to a GH matrix $H(2,\lambda)$ over $\mathbb{Z}_2$, where $\lambda=2\mu$.

Two GH matrices H_1 and H_2 of order N are said to be *equivalent* if one can be obtained from the other by a permutation of the rows and columns and adding the same element of $\mathbb{Z}_p$ to all the coordinates in a row or in a column. We can always change the first row and column of a GH matrix into zeros, obtaining an equivalent GH matrix which is called *normalized*. From a GH matrix H, the *generalized Hadamard (GH) code* is $C_H=\bigcup_{\alpha\in\mathbb{Z}_p}(F_H+\alpha\mathbf{1})$, where $F_H+\alpha\mathbf{1}=\{\mathbf{h}+\alpha\mathbf{1}:\mathbf{h}\in F_H\}$, F_H is the code consisting of the rows of H, and $\mathbf{1}$ denotes the all-one vector [DRV16]. Note that C_H is not necessarily linear as a code over $\mathbb{Z}_p$.

A $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$ such that $\Phi(\mathcal{C})$ is a GH code is called a *$\mathbb{Z}_{p^s}$-additive GH code* and $\Phi(\mathcal{C})$ is called a *$\mathbb{Z}_{p^s}$-linear GH code*. Note that a GH code over $\mathbb{Z}_p$ of length N has pN codewords and minimum distance $(p-1)N/p$. If $p=2$, these codes are referred to as $\mathbb{Z}_{2^s}$-additive Hadamard codes and $\mathbb{Z}_{2^s}$-linear Hadamard codes, respectively. The $\mathbb{Z}_4$-linear Hadamard codes of length 2^t have been studied and classified in [Kro01, PRV06], and their automorphism groups have been characterized in [KV15, PPV14]. For $s>2$, $\mathbb{Z}_{2^s}$-linear Hadamard codes where first introduced in [Kro07]. A full classification of $\mathbb{Z}_8$-linear Hadamard codes is provided in [FVV20b]. For $s>3$, a partial classification and bounds on the number of nonequivalent $\mathbb{Z}_{2^s}$-linear Hadamard codes of length 2^t can be found in [FVV19]. More generally, for any $s\ge 2$ and p prime, $\mathbb{Z}_{p^s}$-linear GH codes are studied and partially classified in [BFV22a, BFVV24]. Moreover, it is proved that, for $p\ge 3$, the $\mathbb{Z}_{p^s}$-linear GH codes of type $(n;1,0,\dots,0,t_s)$ are the only ones which are linear [BFV22a]. For $p=2$, they are only linear when their type is $(n;1,0,\dots,0,t_s)$ or $(n;1,0,\dots,0,1,t_s)$ [FVV19].

Generator matrices having minimum number of rows for $\mathbb{Z}_4$-additive Hadamard codes, as well as a recursive construction of these matrices, are given in [Kro01]. In [Kro07], $\mathbb{Z}_{2^s}$-additive Hadamard codes with $s > 2$ are introduced and generator matrices with minimum number of rows are also given. A recursive construction for these matrices is presented in [FVV19]. More recently, $\mathbb{Z}_{p^s}$-additive GH codes are considered in [BFV22a] generalizing these results to any $p \geq 3$ prime. In this section, we recall the construction of $\mathbb{Z}_{p^s}$-additive GH codes with $s \geq 2$ and p prime.

Let t_1, t_2,...,t_s be nonnegative integers with $t_1 \geq 1$. Consider the matrix $\mathcal{G}^{t_1,\ldots,t_s}$ whose columns are exactly all the vectors of the form $\mathbf{z}^T$, $\mathbf{z} \in \{1\} \times \mathbb{Z}_{p^s}^{t_1-1} \times (p\mathbb{Z}_{p^s})^{t_2} \times \cdots \times (p^{s-1}\mathbb{Z}_{p^s})^{t_s}$.

Example 8. *For $p = 3$ and $s = 3$, we have the following matrices over $\mathbb{Z}_{27}$:*

$$\mathcal{G}^{1,0,1} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 9 & 18 \end{pmatrix}, \quad \mathcal{G}^{1,1,0} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{pmatrix},$$

$$\mathcal{G}^{2,0,0} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 \end{pmatrix},$$

$$\mathcal{G}^{1,1,1} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \end{pmatrix},$$

$$\mathcal{G}^{2,0,1} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & \cdots & 26 & 0 & 1 & 2 & \cdots & 26 & 0 & 1 & 2 & \cdots & 26 \\ 0 & 0 & 0 & \cdots & 0 & 9 & 9 & 9 & \cdots & 9 & 18 & 18 & 18 & \cdots & 18 \end{pmatrix}.$$

Let $\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \mathbf{p^s - 1}$ be the vectors having the same element $0, 1, 2, \ldots, p^s - 1$ from $\mathbb{Z}_{p^s}$ in all its coordinates, respectively. The *order* of a vector $\mathbf{u}$ over $\mathbb{Z}_{p^s}$, denoted by $\mathrm{ord}(\mathbf{u})$, is the smallest positive integer m such that $m\mathbf{u} = \mathbf{0}$.

Any matrix $\mathcal{G}^{t_1,\ldots,t_s}$ can also be obtained by applying the following recursive construction. We start with $\mathcal{G}^{1,0,\ldots,0} = (1)$. Then, from a matrix $\mathcal{G} = \mathcal{G}^{t_1,\ldots,t_s}$, we can construct a new matrix $\mathcal{G}_i$, for any $i \in \{1, \ldots, s\}$, such

that

$$\mathcal{G}_i = \begin{pmatrix} \mathcal{G} & \mathcal{G} & \cdots & \mathcal{G} \\ 0 \cdot \mathbf{p^{i-1}} & 1 \cdot \mathbf{p^{i-1}} & \cdots & (p^{s-i+1}-1) \cdot \mathbf{p^{i-1}} \end{pmatrix}. \tag{2.21}$$

Finally, permuting the rows of $\mathcal{G}_i$, we obtain a matrix $\mathcal{G}^{t'_1,\ldots,t'_s}$, where $t'_j = t_j$ for $j \neq i$ and $t'_i = t_i + 1$. Note that any permutation of columns of $\mathcal{G}_i$ gives also a matrix $\mathcal{G}^{t'_1,\ldots,t'_s}$.

Example 9. *From the matrix $\mathcal{G}^{1,0,0} = (1)$, we obtain the matrix $\mathcal{G}^{2,0,0}$; and from $\mathcal{G}^{2,0,0}$ we can construct $\mathcal{G}^{2,0,1}$, where $\mathcal{G}^{2,0,0}$ and $\mathcal{G}^{2,0,1}$ are the matrices given in Example 8. Note that we can also generate another matrix $\mathcal{G}^{2,0,1}$ as follows: from $\mathcal{G}^{1,0,0} = (1)$ we obtain the matrix $\mathcal{G}^{1,0,1}$ given in Example 8, and from $\mathcal{G}^{1,0,1}$ we can construct the matrix*

$$\mathcal{G}_1 = \begin{pmatrix} 1\,1\,1 & 1\,1\,1 & 1\,1\,1 & \cdots & 1 & 1 & 1 & 1 & 1 & 1 \\ 0\,9\,18 & 0\,9\,18 & 0\,9\,18 & \cdots & 0 & 9 & 18 & 0 & 9 & 18 \\ 0\,0\,0 & 1\,1\,1 & 2\,2\,2 & \cdots & 25 & 25 & 25 & 26 & 26 & 26 \end{pmatrix}.$$

Then, after permuting the rows of $\mathcal{G}_1$, we have the matrix

$$\mathcal{G}^{2,0,1} = \begin{pmatrix} 1\,1\,1 & 1\,1\,1 & 1\,1\,1 & \cdots & 1 & 1 & 1 & 1 & 1 & 1 \\ 0\,0\,0 & 1\,1\,1 & 2\,2\,2 & \cdots & 25 & 25 & 25 & 26 & 26 & 26 \\ 0\,9\,18 & 0\,9\,18 & 0\,9\,18 & \cdots & 0 & 9 & 18 & 0 & 9 & 18 \end{pmatrix},$$

which is different to the matrix $\mathcal{G}^{2,0,1}$ given in Example 8. Note that these two matrices $\mathcal{G}^{2,0,1}$ generate permutation equivalent codes.

In this thesis, we assume that the matrices $\mathcal{G}^{t_1,\ldots,t_s}$ are constructed recursively starting from $\mathcal{G}^{1,0,\ldots,0}$ in the following way. First, we obtain $\mathcal{G}^{t_1,0,\ldots,0}$ by adding $t_1 - 1$ rows of order p^s; then $\mathcal{G}^{t_1,t_2,0,\ldots,0}$ is generated by adding t_2 rows of order p^{s-1}; and so on, until $\mathcal{G}^{t_1,\ldots,t_s}$ is reached by adding t_s rows of order p.

We denote by $\mathcal{H}^{t_1,\ldots,t_s}$ the $\mathbb{Z}_{p^s}$-additive code of type $(n; t_1, \ldots, t_s)$ generated by $\mathcal{G}^{t_1,\ldots,t_s}$, where $t_1, \ldots, t_s$ are nonnegative integers with $t_1 \geq 1$. Note that $n = p^{t-s+1}$, where $t = (\sum_{i=1}^{s}(s-i+1) \cdot t_i) - 1$. Let $H^{t_1,\ldots,t_s} = \Phi(\mathcal{H}^{t_1,\ldots,t_s})$ denote the corresponding $\mathbb{Z}_{p^s}$-linear code, which is a GH code of

length p^t [BFV22a]. Thus, we say that $\mathcal{H}^{t_1,\dots,t_s}$ is a $\mathbb{Z}_{p^s}$-additive GH code, and $H^{t_1,\dots,t_s}$ a $\mathbb{Z}_{p^s}$-linear GH code.

For the permutation decoding method described in Section 2.3 to be efficient, we need to find r-PD-sets for the corresponding code. This can be a difficult task for the vast majority of codes. However, there are some known families of codes which present an approachable permutation automorphism group, making the search for r-PD-sets an easier task. In particular, the permutation automorphism group of $\mathbb{Z}_4$-linear Hadamard codes have been studied and a construction of r-PD-sets was given [BV18]. In Chapter 4 we generalize some of these results, describing the structure of the permutation automorphism group for any $\mathbb{Z}_{p^s}$-additive GH code, and giving two constructions of r-PD-sets. In Chapter 5, these constructions are improved, achieving a larger value of r for non-free codes.

2.5 Computation of the minimum weight

The minimum distance is one of the most important invariants of a code, and it plays a crucial role in determining whether a code is considered "good" or not. As it is shown in Section 2.1, the minimum distance d of a code is directly connected to its error-correcting capability t, as shown in (2.4). Therefore, for codes with the same length and number of codewords (or dimension in the case of linear codes), a higher minimum distance is preferred. If the code is linear, then the minimum distance coincides with the minimum weight of the code.

The computation of the minimum distance of a code is not a trivial task. In fact, we only deal with the minimum weight, which is enough for linear codes. Even in the case of binary linear codes, it has been shown that it is an NP-hard problem [Var97]. In general, only lower bounds are known for the minimum weight of most linear codes. Therefore, it is important to have a computational method which is able to obtain the minimum weight in an efficient way.

A survey on the different methods of calculating the minimum weight

of a linear code can be found in [Whi06]. We focus on the best-performing method for the more general cases: the Brouwer-Zimmermann algorithm. This method was invented originally by Brouwer, but was not published, and later it was improved by Zimmermann [Zim96]. A description of this method, along with some specifics about its implementation for linear codes over finite fields, can be found in [Gra06]. Then, in [Whi06], White proposes an adaptation of this algorithm for linear codes over $\mathbb{Z}_4$, which computes the minimum *Lee weight* of the code. Note that the Lee weight coincides with the homogeneous weight, defined in (2.12), for $p = s = 2$. Therefore, the computation of the minimum Lee weight of a linear code over $\mathbb{Z}_4$ also gives the minimum Hamming weight for the corresponding $\mathbb{Z}_4$-linear code.

In the following lines we describe the fundamental concepts of Brouwer's and Zimmermann's methods. First for linear codes over finite fields and then White's adaptation for linear codes over $\mathbb{Z}_4$.

A brute force method to compute the minimum weight of a code C can be achieved by computing the weight of all codewords in C and taking the lowest nonzero weight. For linear codes, Brouwer's algorithm is able to reduce the number of codewords that need to be enumerated. During this method, a lower bound on the weight of all codewords yet to be enumerated is tracked. This lower bound is updated using the codewords that have already been enumerated, that is, whose weight has been determined. Once the lower bound is equal or greater than the minimum weight found up to this point, then this weight is acknowledged as the minimum weight of the code.

The enumeration of codewords is based on the enumeration of information vectors of constant weight r. Starting from the lowest possible weight, every information vector $\mathbf{m}$ of weight r is considered and the weight of the corresponding codeword $\mathbf{c} = \mathbf{m}G$ is computed, where G is a generator matrix of C. The following lemma can be easily proven and establishes the main point of the method.

Lemma 10. *Let $C[n,k]_p$ be a linear code over $\mathbb{Z}_p$ with an information set I and a generator matrix G such that $G_I = \mathrm{Id}$, where G_I denotes the matrix formed by the columns of G given by I. If $\mathbf{m} \in \mathbb{Z}_p^k$ is an information vector*

and $\mathbf{c} = \mathbf{m}G$ *is a codeword of* C*. Then* $\mathrm{wt}(\mathbf{c}) \geq \mathrm{wt}(\mathbf{c}_I) = \mathrm{wt}(\mathbf{m})$.

Suppose that all information vectors of weight less than or equal to r have been enumerated. We know that any codeword $\mathbf{c} = \mathbf{m}G$ that has not been enumerated yet satisfies $\mathrm{wt}(\mathbf{c}) \geq \mathrm{wt}(\mathbf{m}) > r$. That is, $r + 1$ is a lower bound on the weight of all codewords yet to be enumerated. The enumeration ends when a codeword is found with weight equal to or lower than the lower bound.

However, using this method, the lower bound increases at an excessively slow pace. Note that the bound given by Lemma 10 only takes into account the weight coming from the information coordinates. For longer codes, this bound becomes insignificant. In order to take benefit of the other coordinates we consider a sequence of disjoint information sets $I_1, \ldots, I_h$ and a sequence of generator matrices $G_1, \ldots, G_h$ such that G_j is a systematic generator matrix for I_j. Then Lemma 10 leads to the following theorem.

Theorem 11 ([Whi06])**.** *Let* $C[n, k]_p$ *be a linear code over* $\mathbb{Z}_p$*. Let* $I_1, \ldots, I_h$ *be a sequence of disjoint information sets for* C *and* $G_1, \ldots, G_h$ *a sequence of systematic generator matrices of* C *such that* G_j *is a systematic generator matrix for* I_j*. Consider the set*

$$S_i = \{\mathbf{m}G_i : \mathbf{m} \in \mathbb{Z}_p^k, \mathrm{wt}(\mathbf{m}) \leq r\},$$

for each matrix G_i*, then all* $\mathbf{c} \in C \setminus \bigcup_{i=1}^{h} S_i$ *satisfy*

$$\mathrm{wt}(\mathbf{c}) \geq h(r + 1). \tag{2.22}$$

Using Theorem 11, we can improve the method by considering all the coordinates included in the information sets. For each weight r, in an increasing order, every information vector $\mathbf{m}$ of weight r is considered and the weight of the codewords $\mathbf{c}_1 = \mathbf{m}G_1, \ldots, \mathbf{c}_h = \mathbf{m}G_h$ is computed. After enumerating all information vectors of weight less than or equal to r, the lower bound is updated to $h(r + 1)$. This process is known as Brouwer's method.

An improvement was proposed by Zimmermann [Zim96] by considering non-disjoint information sets in order to obtain more generator matrices and,

in turn, a faster growth of the lower bound. The overlap between different information sets comes at a cost of efficiency, since some coordinates are considered more than once for the lower bound. However, there are many cases where the benefits in a faster growing lower bound overcome the inefficiency of duplicated information coordinates.

The overlap between information sets is defined in terms of previously defined information sets. We use the concept of *relative rank*, defined as $k_i = |I_i| - |I_i \cap \bigcup_{j<i} I_j|$. Note that for the first generator matrix $k_1 = |I_1| = k$, where k is the dimension of the code. Using this notation, Theorem 11 can be adapted to non-disjoint information sets.

Theorem 12 ([Whi06])**.** *Let $C[n,k]_p$ be a linear code over $\mathbb{Z}_p$. Let $I_1, \ldots, I_h$ be a sequence of disjoint information sets for C, with relative ranks $k_1, \ldots, k_h$, and let $G_1, \ldots, G_h$ be a sequence of systematic generator matrices of C such that G_j is a systematic generator matrix for I_j. Consider the set*

$$S_i = \{\mathbf{m}G_i : \mathbf{m} \in \mathbb{Z}_p^k, \, \mathrm{wt}(\mathbf{m}) \leq r\},$$

for each matrix G_i, then all $\mathbf{c} \in C \setminus \bigcup_{i=1}^{h} S_i$ satisfy

$$\mathrm{wt}(\mathbf{c}) \geq \sum_{i=1}^{h} \max\left(0, r + 1 - (k - k_i)\right). \tag{2.23}$$

Zimmermann's adaptation works especially well when it is able to provide more generator matrices than Brouwer's method with little overlap between information sets. Even when the maximum number of disjoint information sets, i.e. $\lceil n/k \rceil$, is reached by Brouwer's method, Zimmermann's variation may be able to produce one extra information set with a relative rank of $n \pmod k$. A hybrid method can be implemented, which selects the best variation for each case. This method is known as the Brouwer-Zimmermann method. In [Whi06], this method is generalized to linear codes over $\mathbb{Z}_4$ and, in Section 7.3.3, we generalize it to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, which are subgroups of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$, that is, linear codes with some coordinates over $\mathbb{Z}_2$ and some over $\mathbb{Z}_4$.

Chapter 3

Systematic encoding for $\mathbb{Z}_{p^s}$-linear codes

It is well known that linear codes are always systematic. Indeed, any set of k linearly independent columns of a generator matrix G of a linear code of dimension k forms an information set for C. This is not the case for nonlinear codes, as pointed out in Section 2.1. Let C be a code over $\mathbb{Z}_p$ of length n and with p^k codewords, there may not be a set of k coordinates I such that $|C_I| = |C|$. However, in [BBFV15], it is shown that any (nonlinear) $\mathbb{Z}_4$-linear code is systematic and a systematic encoding is given for a specific information set. Also in [BBFV15], a new permutation decoding method is introduced, which can be used for any nonlinear binary code as long as it has a systematic encoding. In particular, this new method can be applied to $\mathbb{Z}_4$-linear codes, using the given systematic encoding.

In this chapter, we generalize these results to $\mathbb{Z}_{p^s}$-linear codes with p prime and $s \geq 2$. That is, we give a systematic encoding for $\mathbb{Z}_{p^s}$-linear codes and show that the new permutation decoding method can be easily generalized to any systematic nonlinear code over $\mathbb{Z}_p$. The generalization of the systematic encoding is not as straightforward as one might expect, since the generalized Gray map ϕ_s is not bijective for $p > 2$ or $s > 2$. An earlier version of these results, for $p = 2$, was presented in [TV20]. The general version, for any p prime, was published in [TV22a].

The chapter is organized as follows. In Section 3.1, some preliminary remarks on $\mathbb{Z}_{p^s}$-linear codes and the generalization of Carlet's Gray map Φ_s are given. In Section 3.2, a systematic encoding for $\mathbb{Z}_{p^s}$-linear codes is given. In Section 3.3, we recall the permutation decoding algorithm described in [BBFV15] for binary codes (not necessarily linear), and see that it also applies to codes over $\mathbb{Z}_p$, as long as a systematic encoding is provided. We show some examples of how to use the described systematic encoding for $\mathbb{Z}_{p^s}$-linear codes in this permutation decoding algorithm.

3.1 Generalization of Carlet's Gray map

Consider the generalization of Carlet's Gray map ϕ_s given by (2.13). Recall that Y_{s-1} is a matrix of size $(s-1) \times p^{s-1}$ whose columns are all the distinct elements from $\mathbb{Z}_p^{s-1}$. The elements of $\mathbb{Z}_p^{s-1}$ can also be seen as the p-ary expansions of the elements of $\mathbb{Z}_{p^{s-1}}$. Therefore, we can arrange the columns of Y_{s-1} in ascending order, as elements of $\mathbb{Z}_{p^{s-1}}$. This definition of the matrix Y_{s-1} enables a recursive construction. Starting from $Y_1 = \begin{pmatrix} 0 & 1 & \dots & p-1 \end{pmatrix}$, we can define Y_s recursively as follows:

$$Y_s = \begin{pmatrix} Y_{s-1} & Y_{s-1} & \dots & Y_{s-1} \\ \mathbf{0} & \mathbf{1} & \dots & \mathbf{p-1} \end{pmatrix}, \tag{3.1}$$

where $\mathbf{0}$ and $\mathbf{1}$ denote the all-zero and all-one vectors, respectively; and $\mathbf{p-1}$ denotes the vector with all coordinates equal to $p-1$.

The generalization of Carlet's Gray map ϕ_s can also be defined as follows: $\phi_s(u) = (u_0, \dots, u_{s-2}, u_{s-1})M_{s-1}$, where $[u_0, u_1, \dots, u_{s-1}]_p$ is the p-ary expansion of $u \in \mathbb{Z}_{p^s}$ and

$$M_{s-1} = \begin{pmatrix} Y_{s-1} \\ \mathbf{1} \end{pmatrix}. \tag{3.2}$$

This definition is equivalent to the one given in [GS99]. Note that the image of ϕ_s can be seen as a linear code over $\mathbb{Z}_p$ generated by the matrix M_{s-1}. It is easy to check that it is a linear two-weight code of size p^s with nonzero

weights $(p-1)p^{s-2}$ and p^{s-1}. Indeed, it is a linear generalized Hadamard code of length p^{s-1} over $\mathbb{Z}_p$, which comes from a generalized Hadamard matrix $H(p, p^{s-2})$, known as the Sylvester Hadamard matrix. It can also be seen as the p-ary first order Reed-Muller code as mentioned in [GS99].

Let C_H be a GH code of length p^{s-1} over $\mathbb{Z}_p$. We can arrange the codewords in $C_H = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{p^s-1}\}$ in such a way that $\mathbf{c}_0 = \mathbf{0}$ and for all u, $0 \leq u \leq p^{s-1}-1$, and j, $0 \leq j \leq p-1$, we have $\mathbf{c}_{u+jp^{s-1}} - \mathbf{c}_u = (j, j, \ldots, j)$. Then, more generally, a Gray map ϕ_s from $\mathbb{Z}_{p^s}$ to $\mathbb{Z}_p^{p^{s-1}}$ can be defined as $\phi_s(u) = \mathbf{c}_u$ [Kro07, SWK19]. It is easy to see that the generalization of Carlet's Gray map (2.13), or equivalently the one defined from (3.2), is a particular case of this one. From now on, we only consider the generalization of Carlet's Gray map.

3.2 Systematic encoding for $\mathbb{Z}_{p^s}$-linear codes

In [BBFV15], a systematic encoding for any $\mathbb{Z}_4$-linear code (not necessarily linear) is given. In this section, we generalize this result to $\mathbb{Z}_{p^s}$-linear codes. Specifically, an information set I for these codes is found, and a systematic encoding with respect to I is defined. Therefore, we prove that they are always systematic.

Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of length n and type $(t_1, \ldots, t_s)$, and $C = \Phi_s(\mathcal{C})$ its corresponding $\mathbb{Z}_{p^s}$-linear code. Recall that C has length $p^{s-1}n$. Moreover, since C has $p^{st_1+(s-1)t_2+\cdots+t_s}$ codewords, an information set I of C consists of $st_1 + (s-1)t_2 + \cdots + t_s$ coordinate positions out of a total of $p^{s-1}n$ and it requires that $|C_I| = p^{st_1+(s-1)t_2+\cdots+t_s}$. This last condition can be proved directly if we find a systematic encoding with respect to I, that is, an injective map f from the information space $\mathbb{Z}_p^{st_1+(s-1)t_2+\cdots+t_s}$ to C such that $f(\mathbf{a})|_I = \mathbf{a}$ for any information vector $\mathbf{a} \in \mathbb{Z}_p^{st_1+(s-1)t_2+\cdots+t_s}$, since there are exactly $p^{st_1+(s-1)t_2+\cdots+t_s}$ different vectors in the information space.

If $C = \Phi_s(\mathcal{C})$ is linear, we know that it is systematic. Actually, in this case, we can always find a permutation equivalent code C' with a generator matrix G' in standard form. Then, the first $st_1+(s-1)t_2+\cdots+t_s$ coordinates form an

information set for C' and the multiplication by G' constitutes a systematic encoding. In general, the information set and systematic encoding described in this section for any $\mathbb{Z}_{p^s}$-linear code C do not coincide with the one obtained from G' when C is linear.

On the other hand, if C is nonlinear, we need to use a different approach to obtain a systematic encoding. First, note that it is not possible to generalize directly the procedure followed in [BBFV15] for $\mathbb{Z}_4$-linear codes, since there are some important differences. For example, for $\mathbb{Z}_4$-linear codes, it is easy to consider the inverse of the Gray map Φ_2, since it is bijective. However, for $\mathbb{Z}_{p^s}$-linear codes with $p^s \neq 4$, the generalized Gray map Φ_s is not bijective. In order to solve this problem, we find a set of s coordinate positions L_s such that $\phi_s(\mathbb{Z}_{p^s})|_{L_s}$ has p^s different vectors. Since $\phi_s(\mathbb{Z}_{p^s})$ is a linear code over $\mathbb{Z}_p$ generated by the matrix M_{s-1}, we can determine L_s by finding s linearly independent columns in M_{s-1}. This is equivalent to finding an information set for the linear code $\phi_s(\mathbb{Z}_{p^s})$.

Proposition 13. *Let M_{s-1} be a matrix of the form of (3.2) with $s \geq 2$. Then, the column vectors of M_{s-1} corresponding to the set of coordinate positions $L_s = \{1, p^0 + 1, p^1 + 1, \ldots, p^{s-2} + 1\}$ form a set of s linearly independent vectors. In other words, L_s is an information set for $\phi_s(\mathbb{Z}_{p^s})$.*

Proof. We prove the result by induction on s. For $s = 2$, we have $L_2 = \{1, 2\}$ and clearly both columns of M_1 are independent, since

$$M_1 = \begin{pmatrix} 0 & 1 & \ldots & p-1 \\ 1 & 1 & \ldots & 1 \end{pmatrix}.$$

Let $\mathbf{m}'_i$ be the ith column vector of the matrix M_{s-2}. For $s > 2$, by induction hypothesis, we have that the column vectors $\mathbf{m}'_1, \mathbf{m}'_{p^0+1}, \mathbf{m}'_{p^1+1}, \ldots, \mathbf{m}'_{p^{s-3}+1}$ are independent. By (3.1) and (3.2), we have

$$M_{s-1} = \begin{pmatrix} Y_{s-2} & Y_{s-2} & \ldots & Y_{s-2} \\ \mathbf{0} & \mathbf{1} & \ldots & \mathbf{p-1} \\ \mathbf{1} & \mathbf{1} & \ldots & \mathbf{1} \end{pmatrix}.$$

A permutation of rows does not affect the linear dependency between columns. Therefore, we can equivalently consider

$$\tilde{M}_{s-1} = \begin{pmatrix} Y_{s-2} & Y_{s-2} & \dots & Y_{s-2} \\ \mathbf{1} & \mathbf{1} & \dots & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \dots & \mathbf{p-1} \end{pmatrix} = \begin{pmatrix} M_{s-2} & M_{s-2} & \dots & M_{s-2} \\ \mathbf{0} & \mathbf{1} & \dots & \mathbf{p-1} \end{pmatrix}.$$

Let $\mathbf{m}_i$ be the ith column vector of $\tilde{M}_{s-1}$. By construction, $\mathbf{m}_i = (\mathbf{m}'_i, 0)$ for all $1 \leq i \leq p^{s-2}$. By induction hypothesis, $K = \{\mathbf{m}_1, \mathbf{m}_{p^0+1}, \mathbf{m}_{p^1+1}, \dots, \mathbf{m}_{p^{s-3}+1}\}$ is a set of linearly independent column vectors. Moreover, since the vectors in K have a zero in the last coordinate and $\mathbf{m}_{p^{s-2}+1}$ has a one in the last coordinate, the column vector $\mathbf{m}_{p^{s-2}+1}$ cannot be expressed as a linear combination of the vectors in K. Therefore, $\{\mathbf{m}_1, \mathbf{m}_{p^0+1}, \mathbf{m}_{p^1+1}, \dots, \mathbf{m}_{p^{s-3}+1}, \mathbf{m}_{p^{s-2}+1}\}$ is a set of linearly independent vectors and the result follows. □

An information vector has the form $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s)$, where $\mathbf{a}_k \in \mathbb{Z}_p^{(s-k+1)t_k}$. Now, we define a map that sends this vector $\mathbf{a}$ to an element of $\mathbb{Z}_{p^s}^{t_1+\dots+t_s}$. First, we define an alternative component-wise Gray map $\Phi : \mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \dots \times \mathbb{Z}_p^{t^s} \longrightarrow \mathbb{Z}_p^{p^{s-1}t_1+p^{s-2}t_2+\dots+t_s}$, denoted by $\Phi = (\Phi_s, \Phi_{s-1}, \dots, \Phi_1)$, such that $\Phi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s) = (\Phi_s(\mathbf{x}_1), \Phi_{s-1}(\mathbf{x}_2), \dots, \Phi_1(\mathbf{x}_s))$, where $\mathbf{x}_k \in \mathbb{Z}_{p^{s-k+1}}^{t_k}$ and

$$\Phi_k : \mathbb{Z}_{p^k}^{t_{s-k+1}} \longrightarrow \mathbb{Z}_p^{p^{k-1}t_{s-k+1}} \tag{3.3}$$

is the Gray map extended coordinate-wise, for $2 \leq k \leq s$, and Φ_1 is the identity map.

Next, we define an "inverse" function of Φ, denoted by $\Phi^{-1} = (\Phi_s^{-1}, \Phi_{s-1}^{-1}, \dots, \Phi_1^{-1})$. Recall that Φ_k, where $1 \leq k \leq s$, is bijective only for $k = 1$ and $p = k = 2$. However, from the set L_k of coordinate positions, defined in Proposition 13, it is possible to generate the image of ϕ_k. This means that, given a vector $\mathbf{a} \in \mathbb{Z}_p^k$, there exists a unique element $\mathbf{v} \in \phi_k(\mathbb{Z}_{p^k}) \subseteq \mathbb{Z}_p^{p^{k-1}}$ such that $\mathbf{v}|_{L_k} = \mathbf{a}$. Then, we define $\phi_k^{-1} : \mathbb{Z}_p^k \longrightarrow \mathbb{Z}_{p^k}$, where $\phi_k^{-1}(\mathbf{a})$ is the inverse image of the vector $\mathbf{v}$ such that $\mathbf{v}|_{L_k} = \mathbf{a}$. We define the coordinate-wise

extension of this map as $\Phi_k^{-1} : \mathbb{Z}_p^{kt_{s-k+1}} \longrightarrow \mathbb{Z}_{p^k}^{t_{s-k+1}}$.

Example 14. *Let us consider the Gray map $\phi_3 : \mathbb{Z}_8 \longrightarrow \mathbb{Z}_2^4$, which is defined as*

$$\begin{aligned}
\phi_3(0) &= (0,0,0,0), \quad \phi_3(4) = (1,1,1,1),\\
\phi_3(1) &= (0,1,0,1), \quad \phi_3(5) = (1,0,1,0),\\
\phi_3(2) &= (0,0,1,1), \quad \phi_3(6) = (1,1,0,0),\\
\phi_3(3) &= (0,1,1,0), \quad \phi_3(7) = (1,0,0,1).
\end{aligned}$$

Note that in this case

$$Y_2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{3.4}$$

Since $p = 2$ and $k = 3$, we consider a vector $\mathbf{a} = (1,0,1) \in \mathbb{Z}_2^3$. Note that $\mathbf{v}|_{L_3} = \mathbf{a}$, where $\mathbf{v} = \phi_3(5) = (1,0,1,0)$ and $L_3 = \{1,2,3\}$. Then, $\phi_3^{-1}(\mathbf{a}) = 5 \in \mathbb{Z}_8$. We can extend this process coordinate-wisely. Consider now another vector $\mathbf{a} = (1,0,1,1,0,0) \in \mathbb{Z}_2^6$. First, we split $\mathbf{a}$ into vectors of $k = 3$ components: $\mathbf{a_1} = (1,0,1)$ and $\mathbf{a_2} = (1,0,0)$. Then, we need to find the Gray map images $\mathbf{v}_1$ and $\mathbf{v}_2$ such that $\mathbf{v}_1|_{L_3} = \mathbf{a}_1$ and $\mathbf{v}_2|_{L_3} = \mathbf{a}_2$. Since $\phi_3(5) = (1,0,1,0) = \mathbf{v}_1$ and $\phi_3(7) = (1,0,0,1) = \mathbf{v}_2$, we have $\Phi_3^{-1}(\mathbf{a}) = (5,7) \in \mathbb{Z}_8^2$. □

An additional pair of complementary maps is defined. Let ξ_k be the natural modulo p^{s-k+1} map from $\mathbb{Z}_{p^s}$ to $\mathbb{Z}_{p^{s-k+1}}$, that is, $\xi_k(a) = a \pmod{p^{s-k+1}}$, for $1 \le k \le s$. We denote by ξ the map from $\mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$ to $\mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$ defined as $\xi(\mathbf{u}) = (\xi_1(\mathbf{u}_1), \xi_2(\mathbf{u}_2), \ldots, \xi_s(\mathbf{u}_s))$, where $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s) \in \mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$ and $\mathbf{u}_k \in \mathbb{Z}_{p^s}^{t_k}$. Similarly, let $\iota_k(a) = a$ be the *identity map* from $\mathbb{Z}_{p^{s-k+1}}$ to $\mathbb{Z}_{p^s}$, for $1 \le k \le s$. We denote by ι the map from $\mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$ to $\mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$ defined as $\iota(\mathbf{u}) = (\iota_1(\mathbf{u}_1), \iota_2(\mathbf{u}_2), \ldots, \iota_s(\mathbf{u}_s))$, where $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s) \in \mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$ and $\mathbf{u}_k \in \mathbb{Z}_{p^{s-k+1}}^{t_k}$. Note that $\xi(\iota(\mathbf{u})) = \mathbf{u}$ for any $\mathbf{u} \in \mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$.

From any information vector $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s)$, where $\mathbf{a}_k \in \mathbb{Z}_p^{(s-k+1)t_k}$ for $1 \le k \le s$, we define $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s) \in \mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$ as

follows:

$$\begin{aligned}\mathbf{u} &= (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s) \\ &= \Phi^{-1}(\mathbf{a}) \\ &= \left(\Phi_s^{-1}(\mathbf{a}_1), \Phi_{s-1}^{-1}(\mathbf{a}_2), \ldots, \Phi_1^{-1}(\mathbf{a}_s)\right),\end{aligned}$$

and $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s) \in \mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$ as

$$\begin{aligned}\mathbf{b} &= (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s) \\ &= \iota(\mathbf{u}) \\ &= (\iota_1(\mathbf{u}_1), \iota_2(\mathbf{u}_2), \ldots, \iota_s(\mathbf{u}_s)).\end{aligned}$$

The vector $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s)$ could be encoded by using the $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$, via product by a generator matrix $\mathcal{G}$, which we may assume to be in standard form (2.10). However, if we just multiply it by $\mathcal{G}$, the encoding is not always systematic. Instead, we define an alternative vector $(\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_s) \in \mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$ such that, after multiplying by $\mathcal{G}$ and applying the Gray map Φ_s, we obtain exactly the information vector $\mathbf{a}$ in a certain set of coordinates.

Let us start by defining a family of functions. Let s, t be two integers such that $s \geq t \geq 1$. We define the function $\psi_{s,t} : \mathbb{Z}_{p^s} \longrightarrow \mathbb{Z}_{p^t}$ as follows. Let $[u_0, \ldots, u_{s-1}]_p$ be the p-ary expansion of $u \in \mathbb{Z}_{p^s}$, that is, $u = \sum_{i=0}^{s-1} u_i p^i$. Then, $\psi_{s,t}(u) = \sum_{j=0}^{t-1} u_{s-t+j} p^j \in \mathbb{Z}_{p^t}$, that is, the element of $\mathbb{Z}_{p^t}$ such that its p-ary expansion is $[u_{s-t}, \ldots, u_{s-1}]_p$. Note that if $t = s$, then $\psi_{s,s}(u) = u$. We denote by $\Psi_{s,t}$ its extension coordinate-wise.

Proposition 15. *Let s, t be two integers such that $s \geq t \geq 1$, $u, v \in \mathbb{Z}_{p^s}$. Then, we have*

$$\psi_{s,t}(u + p^{s-t}v) = \psi_{s,t}(u) + \xi_{s-t+1}(v). \tag{3.5}$$

Proof. If $s = t$, we have $\psi_{s,s}(u+v) = u+v = \psi_{s,s}(u)+\xi_1(v)$, since $\psi_{s,s}$ is the identity function and $\xi_1(v) = v$. Otherwise, let $[u_0, \ldots, u_{s-1}]_p$ be the p-ary expansion of $u \in \mathbb{Z}_{p^s}$, that is, $u = \sum_{i=0}^{s-1} u_i p^i$. Let $[v_0, \ldots, v_{s-1}]_p$ be the p-ary

expansion of $v \in \mathbb{Z}_{p^s}$, that is, $v = \sum_{j=0}^{s-1} v_j p^j$. Then, $p^{s-t}v = \sum_{j=0}^{s-1} v_j p^{s-t+j} = \sum_{i=s-t}^{s-1} v_{i-s+t} p^i$. Finally,

$$\begin{aligned}\psi_{s,t}(\sum_{i=0}^{s-1} u_i p^i + \sum_{i=s-t}^{s-1} v_{i-s+t} p^i) &= \psi_{s,t}(\sum_{i=0}^{s-t-1} u_i p^i + \sum_{i=s-t}^{s-1} u_i p^i + \sum_{i=s-t}^{s-1} v_{i-s+t} p^i) \\ &= \sum_{i=0}^{t-1} u_{s-t+j} p^i + \sum_{i=0}^{t-1} v_i p^i \\ &= \psi_{s,t}(u) + \xi_{s-t+1}(v),\end{aligned}$$

and the result follows. □

Now, let us define the function $\sigma : \mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s} \longrightarrow \mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$ as $\sigma(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s) = (\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_s)$, where the vectors $\mathbf{b}'_k \in \iota_k(\mathbb{Z}_{p^{s-k+1}}^{t_k})$, for $1 \leq k \leq s$, are given by

$$\begin{aligned}
\mathbf{b}'_1 &= \iota_1(\mathbf{u}_1), \\
\mathbf{b}'_2 &= \iota_2(\mathbf{u}_2 - \Psi_{s,s-1}(\mathbf{b}'_1 A_{0,1})), \\
\mathbf{b}'_3 &= \iota_3(\mathbf{u}_3 - \Psi_{s,s-2}(\mathbf{b}'_1 A_{0,2} + p\mathbf{b}'_2 A_{1,2})), \\
&\vdots \\
\mathbf{b}'_k &= \iota_k(\mathbf{u}_k - \Psi_{s,s-k+1}(\mathbf{b}'_1 A_{0,k-1} + p\mathbf{b}'_2 A_{1,k-1} + \\
&\quad \cdots + p^{k-2}\mathbf{b}'_{k-1} A_{k-2,k-1})), \\
&\vdots \\
\mathbf{b}'_s &= \iota_s(\mathbf{u}_s - \Psi_{s,1}(\mathbf{b}'_1 A_{0,s-1} + p\mathbf{b}'_2 A_{1,s-1} + \\
&\quad \cdots + p^{s-2}\mathbf{b}'_{s-1} A_{s-2,s-1})),
\end{aligned}$$

and $A_{i,j}$ are the submatrices of the generator matrix of $\mathcal{C}$ in standard form (2.10).

The idea of the systematic encoding f is to obtain the original information vector $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s)$, restricting the codeword $f(\mathbf{a}) \in C = \Phi_s(\mathcal{C})$ to some information set I. After obtaining a codeword of $\mathcal{C}$ by multiplying $(\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_s)$ by $\mathcal{G}$, the last step of the encoding method consists in applying the Gray map Φ_s to a codeword of $\mathcal{C} \subseteq \mathbb{Z}_{p^s}^n$. However, the first step was

to perform an "inverse" of $\mathbf{a}_k$ for each $1 \leq k \leq s$. Therefore, we would like to obtain the images of Φ_k by restricting the images of Φ_s to some coordinates. The following proposition enables us to do that.

Proposition 16. *Let* $J_{s,t} = \{1, p^{s-t}+1, 2\cdot p^{s-t}+1, 3\cdot p^{s-t}+1, \ldots, (p^{t-1}-1)\cdot p^{s-t}+1\}$ *be a set of* p^{t-1} *coordinate positions. Then, for* $s \geq t \geq 1$,

$$\phi_s(u)|_{J_{s,t}} = \phi_t(\psi_{s,t}(u)) \tag{3.6}$$

for any $u \in \mathbb{Z}_{p^s}$. *Note that* $J_{s,s} = \{1, \ldots, p^{s-1}\}$.

Proof. Let $u \in \mathbb{Z}_{p^s}$ and $[u_0, \ldots, u_{s-1}]_p$ its p-ary expansion. If $t = 1$, then $J_{s,1} = \{1\}$. Thus, $\phi_s(u)|_{J_{s,1}} = u_{s-1}$. We also have that $\phi_1(\psi_{s,1}(u)) = u_{s-1}$, since ϕ_1 is the identity map and $\psi_{s,1}(u) = u_{s-1} \in \mathbb{Z}_p$. Now, we can assume that $s \geq t \geq 2$.

Recall that the matrix Y_{s-1} is formed by the p-ary expansions of the elements of $\mathbb{Z}_{p^{s-1}}$ as columns in ascending order. Let $\mathbf{y}_i$, for $1 \leq i \leq p^{s-1}$, be the ith column of Y_{s-1}. The columns given by the set $J_{s,t}$ are, in fact, the ones associated to the elements of $p^{s-t}\mathbb{Z}_{p^{s-1}}$, that is, the elements such that the first $s-t$ components in their p-ary expansions are zeros. Therefore, $\mathbf{y}_i = (0, \ldots, 0, \mathbf{y}'_i)^T$ for $i \in J_{s,t}$. Note that $\mathbf{y}'_i$ are columns of $s-1-(s-t) = t-1$ components, which are in fact the p-ary expansions of the elements of $\mathbb{Z}_{p^{t-1}}$. Since they are arranged in ascending order, we have that the columns $\mathbf{y}'_i$, for $i \in J_{s,t}$, form the matrix Y_{t-1}.

We have $\phi_s(u) = (u_0, \ldots, u_{s-1})M_{s-1}$, where M_{s-1} has the form given in (3.2). Let $\mathbf{m}_i$ be the ith column of M_{s-1}, for $1 \leq i \leq p^{s-1}$. Note that $\mathbf{m}_i = (\mathbf{y}_i, 1)^T = (0, \ldots, 0, \mathbf{y}'_i, 1)^T = (0, \ldots, 0, \mathbf{m}'_i)^T$, for $i \in J_{s,t}$, and these columns $\mathbf{m}'_i$ form the matrix M_{t-1}. Restricting $\phi_s(u)$ to the set of coordinate positions $J_{s,t}$, we have

$$\begin{aligned}
&\phi_s(u)|_{J_{s,t}} = \\
&= ((u_0, \ldots, u_{s-1})\mathbf{m}_1, \quad (u_0, \ldots, u_{s-1})\mathbf{m}_{p^{s-t}+1}, \ldots, \\
&\qquad\qquad (u_0, \ldots, u_{s-1})\mathbf{m}_{p^{s-1}-p^{s-t}+1}) \\
&= ((u_{s-t}, \ldots, u_{s-1})\mathbf{m}'_1, (u_{s-t}, \ldots, u_{s-1})\mathbf{m}'_{p^{s-t}+1}, \ldots,
\end{aligned}$$

$$(u_{s-t},\ldots,u_{s-1})\mathbf{m}'_{p^{s-1}-p^{s-t}+1}).$$

Since the p-ary expansion of $u \in \mathbb{Z}_{p^s}$ is $[u_0,\ldots,u_{s-1}]_p$, we have that $\psi_{s,t}(u) \in \mathbb{Z}_{p^t}$ has p-ary expansion $[u_{s-t},\ldots,u_{s-1}]_p$. Moreover, the columns $\mathbf{m}'_1, \mathbf{m}'_{p^{s-t}+1}$, $\ldots, \mathbf{m}'_{p^{s-1}-p^{s-t}+1}$ form the matrix M_{t-1}. Thus,

$$\begin{aligned}
&\phi_t(\psi_{s,t}(u))\\
&= (u_{s-t},\ldots,u_{s-1})M_{t-1}\\
&= ((u_{s-t},\ldots,u_{s-1})\mathbf{m}'_1, (u_{s-t},\ldots,u_{s-1})\mathbf{m}'_{p^{s-t}+1},\ldots,\\
&\qquad\qquad (u_{s-t},\ldots,u_{s-1})\mathbf{m}'_{p^{s-1}+p^{s-t}+1}).
\end{aligned}$$

Therefore, we have $\phi_s(u)|_{J_{s,t}} = \phi_t(\psi_{s,t}(u))$. □

From an information vector $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s)$, we have seen how to obtain $\Phi^{-1}(\mathbf{a}) = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s)$. Then, by applying σ to this vector, we obtain $(\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_s) \in \mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$. Now, by multiplying this last vector by a generator matrix $\mathcal{G}$ in standard form, we have

$$\begin{aligned}
(\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_s)\mathcal{G} = (&\mathbf{b}'_1,\\
&\mathbf{b}'_1 A_{0,1} + p\mathbf{b}'_2,\\
&\mathbf{b}'_1 A_{0,2} + p\mathbf{b}'_2 A_{1,2} + p^2\mathbf{b}'_3,\\
&\vdots\\
&\mathbf{b}'_1 A_{0,s-1} + p\mathbf{b}'_2 A_{1,s-1} + \cdots + p^{s-1}\mathbf{b}'_s,\\
&\mathbf{b}'_1 A_{0,s} + p\mathbf{b}'_2 A_{1,s} + \cdots + p^{s-1}\mathbf{b}'_s A_{s-1,s})\\
= (&\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1}).
\end{aligned}$$

The resulting vector $(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1})$ is a codeword of $\mathcal{C}$, where $\mathbf{c}_i \in \mathbb{Z}_{p^s}^{t_i}$ for $1 \le i \le s$ and $\mathbf{c}_{s+1} \in \mathbb{Z}_{p^s}^{n-\sum_{i=1}^{s} t_i}$. Then, this codeword (vector of $\mathbb{Z}_{p^s}^n$) is mapped to a vector of $\mathbb{Z}_p^{p^{s-1}n}$ by using the Gray map Φ_s. Note that we obtain a codeword of the $\mathbb{Z}_{p^s}$-linear code $C = \Phi_s(\mathcal{C})$. Now, we just need to prove that there exists a set I of $st_1 + (s-1)t_2 + \cdots + t_s$ coordinate positions such that $|C_I| = p^{st_1+(s-1)t_2+\cdots+t_s}$, in order to show that this process defines

a systematic encoding for C with respect to this set I. We use only the first $t_1 + t_2 + \cdots + t_s$ coordinates over $\mathbb{Z}_{p^s}$.

Let $J_{s,t}$ be the set of p^{t-1} coordinate positions given in Proposition 16 for $1 \le t \le s$. Let $J_s \subseteq \{1, \ldots, p^{s-1}\}$ be the set of coordinate positions such that $\Phi_s((\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1}))|_{J_s} = (\Phi_s(\mathbf{c}_1)|_{J_{s,s}}, \Phi_s(\mathbf{c}_2)|_{J_{s,s-1}}, \ldots, \Phi_s(\mathbf{c}_s)|_{J_{s,1}})$, extending each $J_{s,t}$ coordinate-wisely. Note that none of the coordinate positions corresponding to $\mathbf{c}_{s+1}$ is selected by J_s. Then, by Proposition 16, we have

$$
\begin{aligned}
&\Phi_s((\mathbf{c}_1,\mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1}))|_{J_s} \\
&\quad = \big(\Phi_s(\mathbf{b}'_1), \\
&\qquad \Phi_{s-1}(\Psi_{s,s-1}(\mathbf{b}'_1 A_{0,1} + p\mathbf{b}'_2)), \\
&\qquad \vdots \\
&\qquad \Phi_1(\Psi_{s,1}(\mathbf{b}'_1 A_{0,s-1} + p\mathbf{b}'_2 A_{1,s-1} + \cdots + p^{s-1}\mathbf{b}'_s))\big).
\end{aligned}
$$

Recall that $\xi_k(\iota_k(\mathbf{v}_k)) = \mathbf{v}_k$ for any $\mathbf{v}_k \in \mathbb{Z}_{p^{s-k+1}}^{t_k}$. Then, by Proposition 15, we have

$$
\begin{aligned}
&\Phi_s((\mathbf{c}_1,\mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1}))|_{J_s} \\
&\quad = \big(\Phi_s(\mathbf{b}'_1), \\
&\qquad \Phi_{s-1}(\Psi_{s,s-1}(\mathbf{b}'_1 A_{0,1}) + \xi_2(\mathbf{b}'_2)), \\
&\qquad \Phi_{s-2}(\Psi_{s,s-2}(\mathbf{b}'_1 A_{0,2} + p\mathbf{b}'_2 A_{1,2}) + \xi_3(\mathbf{b}'_3)), \\
&\qquad \vdots \\
&\qquad \Phi_1(\Psi_{s,1}(\mathbf{b}'_1 A_{0,s-1} + p\mathbf{b}'_2 A_{1,s-1} + \cdots + \\
&\qquad\qquad p^{s-2}\mathbf{b}'_{s-1} A_{s-2,s-1}) + \xi_s(\mathbf{b}'_s)\big) \\
&\quad = \big(\Phi_s(\mathbf{u}_1), \\
&\qquad \Phi_{s-1}(\Psi_{s,s-1}(\mathbf{b}'_1 A_{0,1}) + \mathbf{u}_2 - \Psi_{s,s-1}(\mathbf{b}'_1 A_{0,1})), \\
&\qquad \vdots \\
&\qquad \Phi_1(\Psi_{s,1}(\mathbf{b}'_1 A_{0,s-1} + \cdots + p^{s-2}\mathbf{b}'_{s-1} A_{s-2,s-1}) + \mathbf{u}_s \\
&\qquad - \Psi_{s,1}(\mathbf{b}'_1 A_{0,s-1} + \cdots + p^{s-2}\mathbf{b}'_{s-1} A_{s-2,s-1})\big)
\end{aligned}
$$

$$= \big(\Phi_s(\mathbf{u}_1), \Phi_{s-1}(\mathbf{u}_2), \ldots, \Phi_1(\mathbf{u}_s)\big).$$

Finally, to find the information coordinates, we restrict these Gray map images to the corresponding set of coordinates L_t given in Proposition 13 for $1 \leq t \leq s$. Again, we consider the coordinate-wise extension of these sets, and we denote them by the same name L_t. Then,

$$\begin{aligned}
&\Phi_s(\mathbf{u}_1)|_{L_s} = \mathbf{a}_1,\\
&\Phi_{s-1}(\mathbf{u}_2)|_{L_{s-1}} = \mathbf{a}_2,\\
&\vdots\\
&\Phi_1(\mathbf{u}_s)|_{L_1} = \mathbf{a}_s.
\end{aligned}$$

Let $L \subseteq \{1, \ldots, p^{s-1}t_1 + p^{s-2}t_2 + \cdots + t_s\}$ be a set of coordinate positions such that

$$(\Phi_s(\mathbf{u}_1), \Phi_{s-1}(\mathbf{u}_2), \ldots, \Phi_1(\mathbf{u}_s))\,|_L = (\Phi_s(\mathbf{u}_1)|_{L_s}, \Phi_{s-1}(\mathbf{u}_2)|_{L_{s-1}}, \ldots, \Phi_1(\mathbf{u}_s)|_{L_1}).$$

Then,

$$\begin{aligned}
(\Phi_s((\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1}))|_{J_s})\,|_L &= \big((\Phi_s(\mathbf{c}_1)|_{J_{s,s}})|_{L_s}, (\Phi_s(\mathbf{c}_2)|_{J_{s,s-1}})|_{L_{s-1}}, \ldots,\\
&\qquad (\Phi_s(\mathbf{c}_s)|_{J_{s,1}})|_{L_1}\big)\\
&= (\Phi_s(\mathbf{u}_1)|_{L_s}, \Phi_{s-1}(\mathbf{u}_2)|_{L_{s-1}}, \ldots, \Phi_1(\mathbf{u}_s)|_{L_1})\\
&= (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s).
\end{aligned}$$

The information sets J_s and L can be condensed into a single information set of $st_1 + (s-1)t_2 + \cdots + t_s$ coordinate positions. The following proposition and its corollary give an explicit description of this information set.

Proposition 17. *Let* $\mathbf{v} = (v_1, \ldots, v_s) \in \mathbb{Z}_p^{p^{s-1}}$ *and consider the set of* $s-k+1$ *coordinate positions* $I_k = \{1, p^{k-1}+1, p^{k-1+1}+1, p^{k-1+2}+1, \ldots, p^{s-2}+1\}$. *Then,*

$$\mathbf{v}|_{I_k} = (\mathbf{v}|_{J_{s,s-k+1}})|_{L_{s-k+1}} \tag{3.7}$$

Proof. For any s and t, such that $s \geq t$, recall the set of p^{t-1} coordinate

positions $J_{s,t} \subseteq \{1, \ldots, p^{s-1}\}$, which was defined in Proposition 16. Note that

$$J_{s,s-k+1} = \{1, p^{k-1}+1, 2p^{k-1}+1, 3p^{k-1}+1, \ldots, (p^{s-k}-1)p^{k-1}+1\}$$

is a set of p^{s-k} coordinate positions and a subset of $\{1, \ldots, p^{s-1}\}$. Also,

$$L_{s-k+1} = \{1, p^0+1, p^1+1, \ldots, p^{s-k-1}+1\}$$

is a set of $s-k+1$ coordinate positions and a subset of $\{1, \ldots, p^{s-k}\}$. Let $\mathbf{u} = (u_1, \ldots, u_{p^{s-k}}) = \mathbf{v}|_{J_{s,s-k+1}} \in \mathbb{Z}_p^{p^{s-k}}$, we have $u_1 = v_1$ and $u_i = v_{(i-1)p^{k-1}+1}$ for $i \in \{2, \ldots, p^{s-k}\}$. Then,

$$\begin{aligned}
\mathbf{u}|_{L_{s-k+1}} &= (u_1, u_{p^0+1}, u_{p+1}, u_{p^2+1} \ldots, u_{p^{s-k-1}+1}) \\
&= (v_1, v_{p^{k-1}+1}, v_{pp^{k-1}+1}, v_{p^2p^{k-1}+1}, \ldots, v_{p^{s-k-1}p^{k-1}+1}) \\
&= \mathbf{v}|_{I_k}.
\end{aligned}$$

□

Let us define the map $\Phi^{(k)}$ on a set of coordinate positions $\mathcal{I}$, given by

$$\begin{aligned}
\Phi^{(k)}(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} \{ & p^{s-1}(i-1)+1, \\
& p^{s-1}(i-1)+p^{k-1}+1, \\
& p^{s-1}(i-1)+p^{k-1+1}+1, \\
& p^{s-1}(i-1)+p^{k-1+2}+1, \\
& \ldots, \\
& p^{s-1}(i-1)+p^{s-2}+1\},
\end{aligned}$$

and define the set of coordinate positions

$$\begin{aligned}
I = \Phi^{(1)}(\{1, \ldots, t_1\}) \cup \Phi^{(2)}(\{t_1+1, \ldots, t_1+t_2\}) \cup \\
\cdots \cup \Phi^{(s)}(\{t_1+\cdots+t_{s-1}+1, \ldots, t_1+\cdots+t_s\}).
\end{aligned} \tag{3.8}$$

Corollary 18. *Let* $\mathbf{v} \in \mathbb{Z}_p^{np^{s-1}}$. *Then,*

$$\mathbf{v}|_I = (\mathbf{v}|_{J_s})|_L. \tag{3.9}$$

Proof. Consider the following representation of $\mathbf{v}$: $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_s)$, where $\mathbf{v}_k \in \mathbb{Z}_p^{p^{s-1}t_k}$. We have

$$\mathbf{v}|_I = (\mathbf{v}_1|_{I_1}, \ldots, \mathbf{v}_s|_{I_s}).$$

Using Proposition 17 on $\mathbf{v}_k|_{I_k}$ for each $1 \leq k \leq s$, we obtain the result. □

Since $|I| = st_1 + (s-1)t_2 + \cdots + t_s$, and

$$\begin{aligned}\Phi_s((\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1}))|_I &= (\Phi_s((\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s, \mathbf{c}_{s+1}))|_{J_s})\,|_L \\ &= (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s).\end{aligned}$$

then the set I of coordinate positions is an information set. Therefore, we have proved the following theorem.

Theorem 19. *Let* $C = \Phi_s(\mathcal{C})$ *be a* $\mathbb{Z}_{p^s}$*-linear code of type* $(t_1, \ldots, t_s)$, *and* $\mathcal{G}$ *a generator matrix of* $\mathcal{C}$ *in standard form. Then,* C *is systematic and* $f(\mathbf{a}) = \Phi_s(\sigma(\Phi^{-1}(\mathbf{a}))\mathcal{G})$ *is a systematic encoding for any* $\mathbf{a} \in \mathbb{Z}_p^{st_1+(s-1)t_2+\cdots+t_s}$ *with respect to the information set* I.

The whole process of encoding an information vector $(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s)$ and the restriction to the information coordinates given by I is represented schematically in Figure 3.1. To further illustrate the procedure, we show a particular case in Example 20.

Example 20. *Consider the* $\mathbb{Z}_{27}$*-linear code* $C = \Phi_3(\mathcal{C})$, *where* $\mathcal{C}$ *is the* $\mathbb{Z}_{27}$*-additive code of type* $(1, 2, 2)$ *with the following generator matrix in standard form:*

$$\mathcal{G} = \begin{pmatrix} 1 & 0 & 1 & 4 & 2 & 8 \\ 0 & 3 & 0 & 0 & 6 & 3 \\ 0 & 0 & 3 & 3 & 0 & 24 \\ 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 19 \end{pmatrix}. \tag{3.10}$$

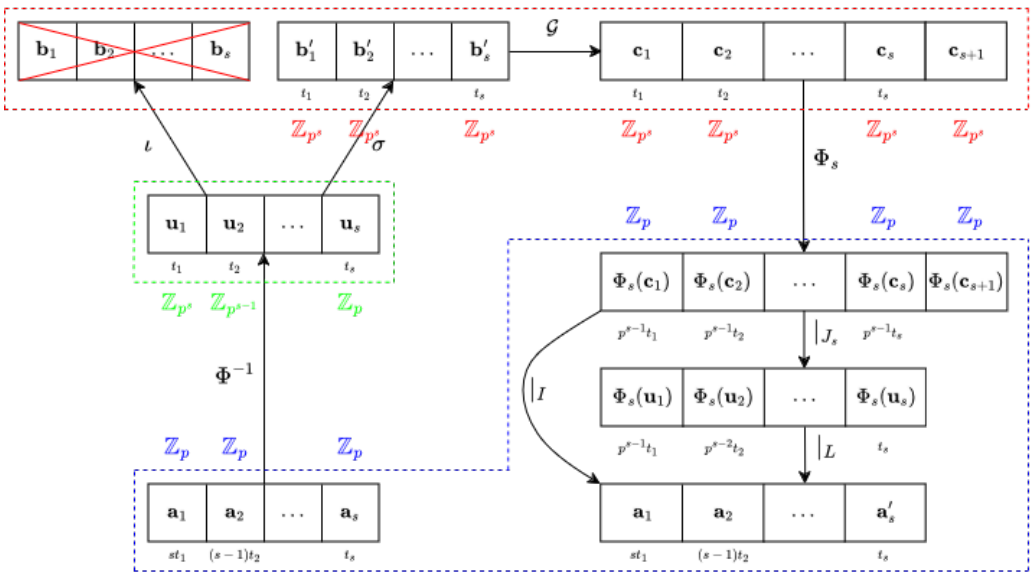

Figure 3.1: Schematic diagram of the encoding process. Under each cell, there is a term representing the length of the corresponding vector. The dashed lines define three regions: the vectors defined over $\mathbb{Z}_{p^s}$, the vectors defined over $\mathbb{Z}_p$, and the vectors in $\mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$.

Let $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3) = (1,2,2,\ 2,1,\ 1,1,\ 0,\ 2)$ *be an information vector over* $\mathbb{Z}_3$. *Note that the length of the vectors* $\mathbf{a}_k$, *where* $k \in \{1,2,3\}$, *is determined by the type of the code:* $\mathbf{a}_1 \in \mathbb{Z}_3^{3\cdot 1}$, $\mathbf{a}_2 \in \mathbb{Z}_3^{2\cdot 2}$ *and* $\mathbf{a}_3 \in \mathbb{Z}_3^{1\cdot 2}$. *First, we identify these vectors with an element of the images of the Gray maps* Φ_3, Φ_2 *and* Φ_1, *respectively. That is, for* $\mathbf{a}_1 = (1,2,2)$, *we take the vector* $(1,2,0,2,0,1,0,1,2) = (1,1,1)M_2 = \Phi_3(13) \in \Phi_3(\mathbb{Z}_{27})$, *since* $L_3 = \{1,2,4\}$. *Similarly, for* $\mathbf{a}_2 = (2,1,\ 1,1)$, *we take the vectors* $(2,1,0) = (2,2)M_1 = \Phi_2(8) \in \Phi_2(\mathbb{Z}_9)$ *and* $(1,1,1) = (0,1)M_1 = \Phi_2(3) \in \Phi_2(\mathbb{Z}_9)$, *since* $L_2 = \{1,2\}$. *Finally, for* $\mathbf{a}_3 = (0,\ 2)$, *we take* (0) *and* (2) *directly from* $\Phi_1(\mathbb{Z}_3) = \mathbb{Z}_3$. *Then,* $\Phi^{-1}(\mathbf{a}) = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) = \mathbf{u}$, *where* $\mathbf{u}_1 = (13) \in \mathbb{Z}_{27}$, $\mathbf{u}_2 = (8,3) \in \mathbb{Z}_9^2$ *and* $\mathbf{u}_3 = (0,2) \in \mathbb{Z}_3^2$. *By applying* σ *to* $\mathbf{u} = (13,8,3,0,2)$, *we have* $\sigma(\mathbf{u}) = (\mathbf{b}'_1, \mathbf{b}'_2, \mathbf{b}'_3) = \mathbf{b}'$, *where* $\mathbf{b}'_1 = \iota_1(\mathbf{u}_1) = (13)$,

$$\begin{aligned}\mathbf{b}'_2 &= \iota_2(\mathbf{u}_2 - \Psi_{3,2}(\mathbf{b}'_1 A_{0,1})\\ &= \iota_2((8,3) - \Psi_{3,2}((0,13))) = \iota_2((8,3) - (0,4)) = \iota_2((8,8)) = (8,8),\end{aligned}$$

$$\begin{aligned}\mathbf{b}'_3 &= \iota_3(\mathbf{u}_3 - \Psi_{3,1}(\mathbf{b}'_1 A_{0,2} + 3\mathbf{b}'_2 A_{1,2})) \\ &= \iota_3((0,2) - \Psi_{3,1}((22,20))) = \iota_3((0,2) - (2,2)) = \iota_3((1,0)) = (1,0).\end{aligned}$$

Now, we multiply $\mathbf{b}' = (13,8,8,1,0)$ *by* $\mathcal{G}$*, and obtain the codeword* $\mathbf{c} = (13,8,8,1,0)\mathcal{G} = (13,24,10,4,20,23) \in \mathcal{C} \subseteq \mathbb{Z}_{27}^6$*. Finally, the encoding of* $\mathbf{a} \in \mathbb{Z}_3^9$ *is* $f(\mathbf{a}) = \Phi_3(13,24,10,4,20,23) \in C \subseteq \mathbb{Z}_3^{54}$*.*

We can check that this encoding is systematic with respect to the information set defined in this section. First, we take the Gray map images of the first five coordinates of $\mathbf{c}$ *restricted to the corresponding sets* $J_{3,3} = \{1,\ldots,9\}$*,* $J_{3,2} = \{1,4,7\}$ *and* $J_{3,1} = \{1\}$*, given by Proposition 16:*

$$\begin{aligned}&\phi_3(13)|_{\{1,\ldots,9\}} = (1,2,0,2,0,1,0,1,2),\\ &\phi_3(24)|_{\{1,4,7\}} = (2,2,2,1,1,1,0,0,0)|_{\{1,4,7\}} = (2,1,0),\\ &\phi_3(10)|_{\{1,4,7\}} = (1,2,0,1,2,0,1,2,0)|_{\{1,4,7\}} = (1,1,1),\\ &\phi_3(4)|_{\{1\}} = (0,1,2,1,2,0,2,0,1)|_{\{1\}} = (0),\\ &\phi_3(20)|_{\{1\}} = (2,1,0,2,1,0,2,1,0)|_{\{1\}} = (2).\end{aligned}$$

Finally, restricting these vectors to the information sets L_t *of* $\phi_t(\mathbb{Z}_{p^t})$*, where* $t \in \{1,2,3\}$*, given by Proposition 13, we obtain the components of the information vector:*

$$\begin{aligned}&(1,2,0,2,0,1,0,1,2)|_{\{1,2,4\}} = (1,2,2),\\ &(2,1,0)|_{\{1,2\}} = (2,1),\\ &(1,1,1)|_{\{1,2\}} = (1,1),\\ &(0)|_{\{1\}} = (0),\\ &(2)|_{\{1\}} = (2).\end{aligned}$$

Alternatively, we can use the information set given in (3.8):

$$\begin{aligned}I &= \Phi^{(1)}(\{1\}) \cup \Phi^{(2)}(\{2,3\}) \cup \Phi^{(3)}(\{4,5\}) \\ &= \{1,2,4\} \cup \{10,13,19,22\} \cup \{28,37\}.\end{aligned}$$

Since

$$f(\mathbf{a}) = (1,2,0,2,0,1,0,1,2,2,2,2,1,1,1,0,0,0,1,2,0,1,2,0,1,2,0,\\ 0,1,2,1,2,0,2,0,1,2,1,0,2,1,0,2,1,0,2,1,0,0,2,1,1,0,2).$$

We obtain $f(\mathbf{a})|_I = (1,2,2,2,1,1,1,0,2) = \mathbf{a}$.

3.3 Permutation decoding for $\mathbb{Z}_{p^s}$-linear codes

In general, $\mathbb{Z}_{p^s}$-linear codes are not linear as codes over $\mathbb{Z}_p$, so it is not possible to perform the standard permutation decoding given in [Mac64, Pra62]. In [BBFV15], an alternative permutation decoding method for $\mathbb{Z}_4$-linear codes is described. In fact, in [BBFV15] it is shown that this method works for any binary systematic code (not necessarily linear). In this section we see that this is also valid for any code over $\mathbb{Z}_p$ (not necessarily linear), as long as a systematic encoding is known for a given information set. Then, the results presented in Section 3.2 enable us to apply the alternative permutation decoding method method to $\mathbb{Z}_{p^s}$-linear codes.

Theorem 6, given in [BBFV15] and described in Section 2.3, gives a necessary and sufficient condition to verify that the information coordinates of a received vector are correct, for $\mathbb{Z}_4$-linear codes. The generalization to $\mathbb{Z}_{p^s}$-linear codes is quite straightforward, as we see in the following theorem.

Theorem 21. *Let C be a systematic t-error-correcting code of length n over $\mathbb{Z}_p$. Let I be an information set of C and f be a systematic encoding with respect to I. Suppose that $\mathbf{y} = \mathbf{x} + \mathbf{e}$ is a received vector, where $\mathbf{x} \in C$ and $\mathrm{wt}_H(\mathbf{e}) \leq t$. Then, the information coordinates of $\mathbf{y}$ are correct if and only if $\mathrm{wt}_H(\mathbf{y} - f(\mathbf{y}|_I)) \leq t$.*

Proof. If $\mathrm{wt}_H(\mathbf{y} - f(\mathbf{y}|_I)) \leq t$, then $f(\mathbf{y}|_I)$ is the closest codeword to $\mathbf{y}$, that is, $f(\mathbf{y}|_I) = \mathbf{x}$. Hence, the systematic coordinates are the same, which means that $\mathbf{y}|_I = \mathbf{x}|_I$. On the other hand, if $\mathbf{x}|_I = \mathbf{y}|_I$, then $\mathrm{wt}_H(\mathbf{y} - f(\mathbf{y}|_I)) = \mathrm{wt}_H(\mathbf{y} - \mathbf{x}) = \mathrm{wt}_H(\mathbf{e}) \leq t$. □

Let C be a $\mathbb{Z}_{p^s}$-linear code with an information set I and a systematic encoding f with respect to I. For example, f and I can be the ones described in Section 3.2. Let $\mathbf{y}$ be a received vector and S a PD-set with respect to I. The alternative permutation decoding algorithm works as follows:

1. If $\mathrm{wt}_H(\mathbf{y}-f(\mathbf{y}|_I)) \leq t$, then the information coordinates of $\mathbf{y}$ are correct and we can decode $\mathbf{y}$ as $\mathbf{y}|_I$.

2. Else, we search for a permutation $\pi \in S$ that satisfies $\mathrm{wt}_H(\pi(\mathbf{y}) - f(\pi(\mathbf{y})|_I)) \leq t$. If there is no such π, we conclude that more than t errors have occurred. Otherwise, if we have found such π, then the decoded vector is $\mathbf{x}|_I$, where

$$\mathbf{x} = \pi^{-1}(f(\pi(\mathbf{y})|_I)). \tag{3.11}$$

Example 22. *Consider the $\mathbb{Z}_8$-linear code $C = \Phi_3(\mathcal{C})$, where $\mathcal{C}$ is the $\mathbb{Z}_8$-additive code of type $(2,0,0)$ with the following generator matrix in standard form:*

$$\mathcal{G} = \begin{pmatrix} 1 & 0 & 7 & 6 & 5 & 4 & 3 & 2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}. \tag{3.12}$$

Note that C is of length 32. Following the method described in Section 3.2, we have the information set $I = \{1,2,3,5,6,7\}$. The minimum distance is $d(C) = 16$, so the error-correcting capability is $t = 7$. It can be checked, by using Magma *[BCFS19] and new functions included in [FTV23], that* $\mathrm{PAut}(C)$ *is a PD-set with respect to I having $2^{12} \cdot 3$ elements. However, we found a subgroup $S \subset \mathrm{PAut}(C)$ with 2^{10} elements which is also a PD-set, where $S = \langle \pi_1, \pi_2, \pi_3, \pi_4, \pi_5 \rangle$,*

$$\begin{aligned}
\pi_1 =&(5,16)(6,15)(7,30)(8,29)(13,24)(14,23)(21,32)(22,31),\\
\pi_2 =&(5,7)(6,8)(13,31)(14,32)(15,29)(16,30)(21,23)(22,24),\\
\pi_3 =&(5,6)(7,8)(13,14)(15,16)(21,22)(23,24)(29,30)(31,32),\\
\pi_4 =&(1,5)(2,22)(3,15)(4,32)(6,18)(7,10)(8,25)(9,24)(11,30)\\
&(12,13)(14,27)(16,20)(17,21)(19,31)(23,26)(28,29),\\
\pi_5 =&(2,18)(3,19)(6,22)(7,23)(10,26)(11,27)(14,30)(15,31).
\end{aligned}$$

Now, we can apply the permutation decoding algorithm. For example, consider the information vector $\mathbf{a} = (110011)$*. Using the systematic encoding given in Section 3.2, the corresponding codeword is*

$$\begin{aligned}\mathbf{x} = f(\mathbf{a}) &= \Phi_3(\sigma(\Phi^{-1}(\mathbf{a}))\mathcal{G}) = \Phi_3(\sigma(6,3)\mathcal{G}) = \Phi_3((6,3)\mathcal{G}) \\ &= \Phi_3(6,3,0,5,2,7,4,1) \\ &= (1100011000001010001110011111010 1).\end{aligned}$$

Note that, in this case, σ *is the identity map. Suppose that we received a vector* $\mathbf{y} = \mathbf{x}+\mathbf{e}$*, where the error vector is* $\mathbf{e} = (0110010000000000000000000000 0000)$*, that is,* $\mathbf{y} = (1010001000001010001110011111010 1)$*. Note that errors have occurred in the second, third and sixth coordinates, and all of them are information coordinates. Restricting* $\mathbf{y}$ *to the information set* I*, we have* $\mathbf{y}|_I = (101001)$*, so*

$$\begin{aligned}f(\mathbf{y}|_I) &= \Phi_3((5,2)\mathcal{G}) = \Phi_3(5,2,7,4,1,6,3,0) \\ &= (1010001110011111010111000 1100000).\end{aligned}$$

We have $\mathrm{wt}_H(\mathbf{y} - f(\mathbf{y}|_I)) = 13 > 7 = t$*. Therefore, by Theorem 21, there are errors in the information coordinates of* $\mathbf{y}$*. We consider the permutation* $\pi_5 \in S \subset \mathrm{PAut}(C)$*. We have*

$$\pi_5(\mathbf{y}) = (1010000001101100001110111 0010011).$$

Restricting $\pi_5(\mathbf{y})$ *to the information set,* $\pi_5(\mathbf{y})|_I = (101000)$*. Encoding this vector, we obtain the codeword*

$$\begin{aligned}f(\pi_5(\mathbf{y})|_I) &= \Phi_3((5,0)\mathcal{G}) = \Phi_3(5,0,3,6,1,4,7,2) \\ &= (1010000001101100010111111 0010011).\end{aligned}$$

We can see that $\mathrm{wt}_H(\pi_5(\mathbf{y}) - f(\pi_5(\mathbf{y})|_I)) = 3 < 7 = t$*. Therefore, we decode*

$\mathbf{y}$ *as the codeword*

$$\begin{aligned}\mathbf{x} &= \pi_5^{-1}(f(\pi_5(\mathbf{y})|_I)) \\ &= (1100011000001010001110011111010 1),\end{aligned}$$

so we assume that the information vector is $\mathbf{x}|_I = (110011)$*, which coincides with* $\mathbf{a}$*.*

Example 23. *Consider the same code as in Example 22. If instead of correcting up to* $t = 7$ *errors, we want to correct up to* $t_0 = 3$ *errors, the following* 3*-PD-set with only* 4 *elements can be used:* $S = \{id, \pi_1, \pi_2, \pi_3\}$*, where* id *is the identity permutation and*

$$\begin{aligned}\pi_1 =&(1,9,17,25)(2,10,18,26)(3,11,19,27)(4,12,20,28)\\ &(5,13,21,29)(6,14,22,30)(7,15,23,31)(8,16,24,32),\\ \pi_2 =&(1,17)(2,18)(3,19)(4,20)(5,21)(6,22)(7,23)(8,24)(9,25)\\ &(10,26)(11,27)(12,28)(13,29)(14,30)(15,31)(16,32),\\ \pi_3 =&(1,25,17,9)(2,26,18,10)(3,27,19,11)(4,28,20,12)\\ &(5,29,21,13)(6,30,22,14)(7,31,23,15)(8,32,24,16).\end{aligned}$$

Now, consider the information vector $\mathbf{a} = (010101)$*. Using the systematic encoding given in Section 3.2, we obtain the following codeword:*

$$\begin{aligned}\mathbf{x} = f(\mathbf{a}) &= \Phi_3((1,5)\mathcal{G}) = \Phi_3(1,5,1,5,1,5,1,5)\\ &= (0101101001011010010110100 1011010).\end{aligned}$$

Suppose that we receive the vector

$$\mathbf{y} = (0111001001011010010110100 1010010).$$

Note that there are errors in the coordinate positions 3*,* 5 *and* 29*. Two of these coordinate positions are in the information set. Since there are only* 3 *errors, there must be a permutation in* S *such that it moves the errors out of the information coordinates. Restricting* $\mathbf{y}$ *to the information set, we have*

$\mathbf{y}|_I = (011001)$ *and*

$$\begin{aligned} f(\mathbf{y}|_I) &= \Phi_3((3,2)\mathcal{G}) = \Phi_3(3,2,1,0,7,6,5,4) \\ &= (01100011010100001001110010101111). \end{aligned}$$

As expected, $\mathrm{wt}_H(\mathbf{y} - f(\mathbf{y}|_I)) = 15 > 7$. *By Theorem 21, this means that there are errors in the information coordinates of* $\mathbf{y}$. *It is clear that the identity permutation id does not work in this case, therefore we search among the other three permutations in* S. *Consider the permutation* $\pi_2 \in S$. *We have*

$$\pi_2(\mathbf{y}) = (01011010010100100111001001011010)$$

and

$$\begin{aligned} f(\pi_2(\mathbf{y})|_I) &= \Phi_3((1,5)\mathcal{G}) = \Phi_3(1,5,1,5,1,5,1,5) \\ &= (01011010010110100101101001011010). \end{aligned}$$

Note that $\mathrm{wt}_H(\pi_2(\mathbf{y}) - f(\pi_2(\mathbf{y})|_I)) = 3 < 7 = t$. *Therefore, there are no errors in the information coordinates and we decode* $\mathbf{y}$ *as the codeword*

$$\begin{aligned} \mathbf{x} &= \pi_2^{-1}(f(\pi_2(\mathbf{y})|_I)) \\ &= (01011010010110100101101001011010). \end{aligned}$$

Restricting $\mathbf{x}$ *to the information set, we obtain* $\mathbf{x}|_I = (010101)$, *which coincides with the original information vector* $\mathbf{a}$.

Chapter 4

r-PD-sets for $\mathbb{Z}_{p^s}$-linear GH codes

The efficiency of the permutation decoding method depends on the size of the r-PD-set $S \subseteq \mathrm{PAut}(C)$, since it needs to find the suitable permutation in S, for each received vector. In general, determining the structure of $\mathrm{PAut}(C)$ is very complex, making the search for r-PD-sets or PD-sets a difficult task. However, there are results that show how to find r-PD-sets of small size for certain families of codes [FKM12, BS13, BV18, BV19, CD21, BS23]. More specifically, in [BV18], it is shown how to find r-PD-sets of size $r+1$ for binary linear Hadamard codes and (nonlinear) $\mathbb{Z}_4$-linear Hadamard codes. A similar result for Hadamard codes over the field $\mathbb{F}_4$ is presented in [CD21]. In this chapter, we generalize the results given in [BV18] to $\mathbb{Z}_{p^s}$-linear GH codes with $s \geq 2$ and p prime. An earlier version of these results, for $p = 2$ and $s = 3$, was presented as a conference talk in [TV22b] and it was published in its proceedings. Then, the complete version was published as a journal paper in [TV24a].

The chapter is organized as follows. In Section 4.1, we explore the permutation automorphism group of $\mathbb{Z}_{p^s}$-linear codes and show that it is isomorphic to a group formed by matrices of the general linear group over $\mathbb{Z}_{p^s}$. In Section 4.2, we give an information set for the corresponding $\mathbb{Z}_{p^s}$-linear GH codes and we establish a criterion for finding r-PD-sets of size $r+1$. In Sections 4.3 we describe two explicit constructions that produce r-PD-sets of size $r+1$ for $\mathbb{Z}_{p^s}$-linear GH codes of type $(n; t_1, 0, \ldots, 0)$ and $(n; 1, 0, \ldots, 0, t_i, 0, \ldots, 0)$,

with $t_1 \geq 2$ and $t_i \geq 1$. These constructions give r-PD-sets for any r up to the upper bounds $f_p^{t_1,0,\ldots,0}$ and $f_p^{1,0,\ldots,0,t_i,0,\ldots,0}$, respectively. In Section 4.4, a recursive construction is also given in order to obtain r-PD-sets for $\mathbb{Z}_{p^s}$-linear GH codes of any type. However, the values of r only reach up to $\tilde{f}_p^{t_1,\ldots,t_s} \leq f_p^{t_1,\ldots,t_s}$. In Section 4.5, we present some computational results on a random search of these sets for the codes where the upper bound is not reached.

4.1 Permutation automorphism group of $\mathbb{Z}_{p^s}$-additive GH codes

The structure of the permutation automorphism group of $\mathbb{Z}_4$-additive Hadamard codes is described in [KV15, PPV14]. In this section, we describe the structure of the permutation automorphism group of the $\mathbb{Z}_{p^s}$-additive GH code $\mathcal{H}^{t_1,\ldots,t_s}$, that is, the structure of $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$. In particular, an isomorphism between $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ and a certain group of matrices of the general linear group over $\mathbb{Z}_{p^s}$ is found, and the order of $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ is computed.

Let $\mathrm{GL}(\kappa, \mathbb{Z}_{p^s})$ denote the *general linear group* of degree κ over $\mathbb{Z}_{p^s}$ and let $\mathcal{L}$ be the set consisting of all matrices over $\mathbb{Z}_{p^s}$ of the following form:

$$\begin{pmatrix} 1 & a_1 & pa_2 & p^2a_3 & \cdots & p^{s-2}a_{s-1} & p^{s-1}a_s \\ \mathbf{0} & A_{1,1} & pA_{1,2} & p^2A_{1,3} & \cdots & p^{s-2}A_{1,s-1} & p^{s-1}A_{1,s} \\ \mathbf{0} & A_{2,1} & A_{2,2} & pA_{2,3} & \cdots & p^{s-3}A_{2,s-1} & p^{s-2}A_{2,s} \\ \mathbf{0} & A_{3,1} & A_{3,2} & A_{3,3} & \cdots & p^{s-4}A_{3,s-1} & p^{s-3}A_{3,s} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & A_{s-1,1} & A_{s-1,2} & A_{s-1,3} & \cdots & A_{s-1,s-1} & pA_{s-1,s} \\ \mathbf{0} & A_{s,1} & A_{s,2} & A_{s,3} & \cdots & A_{s,s-1} & A_{s,s} \end{pmatrix}, \tag{4.1}$$

where $A_{1,1} \in \mathrm{GL}(t_1 - 1, \mathbb{Z}_{p^s})$, $A_{i,i} \in \mathrm{GL}(t_i, \mathbb{Z}_{p^s})$ for $i \in \{2, \ldots, s\}$, $A_{i,j}$ are matrices over $\mathbb{Z}_{p^s}$, for $i \neq j$, $a_1 \in \mathbb{Z}_{p^s}^{t_1-1}$ and $a_j \in \mathbb{Z}_{p^s}^{t_j}$, for $j \in \{2, \ldots, s\}$.

Lemma 24. *The set $\mathcal{L}$ is a subgroup of* $\mathrm{GL}(t_1 + \cdots + t_s, \mathbb{Z}_{p^s})$.

Proof. We first need to check that $\mathcal{L} \subseteq \mathrm{GL}(t_1 + \cdots + t_s, \mathbb{Z}_{p^s})$, i.e., that $\det(\mathcal{M}) \in \mathbb{Z}_{p^s} \backslash p\mathbb{Z}_{p^s}$ for all $\mathcal{M} \in \mathcal{L}$. Note that if $\mathcal{M}' \in \mathrm{GL}(\kappa, \mathbb{Z}_{p^s})$, then $\mathcal{M} = \mathcal{M}' + p\mathcal{R} \in \mathrm{GL}(\kappa, \mathbb{Z}_{p^s})$ for any $\mathcal{R}$. Thus, since $\det(\mathcal{M}') \in \mathbb{Z}_{p^s} \backslash p\mathbb{Z}_{p^s}$, we have that $\det(\mathcal{M}) \in \mathbb{Z}_{p^s} \backslash p\mathbb{Z}_{p^s}$, where

$$\mathcal{M}' = \begin{pmatrix} 1 & a_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{1,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{2,1} & A_{2,2} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{3,1} & A_{3,2} & A_{3,3} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & A_{s-1,1} & A_{s-1,2} & A_{s-1,3} & \cdots & A_{s-1,s-1} & \mathbf{0} \\ \mathbf{0} & A_{s,1} & A_{s,2} & A_{s,3} & \cdots & A_{s,s-1} & A_{s,s} \end{pmatrix}.$$

Finally, we prove that $\mathcal{L}$ is a subgroup. Let us denote by $\mathcal{M}_{i,j}$, for $i, j \in \{1, \ldots, s+1\}$, the submatrix in the ith row and jth column of the block matrix $\mathcal{M} \in \mathcal{L}$ as given in (4.1). Note that $\mathcal{M}_{1,j}$ is a multiple of p^{j-2} for $j \in \{2, \ldots, s+1\}$, and $\mathcal{M}_{i,j}$ is a multiple of p^{j-i} for $i, j \in \{2, \ldots, s+1\}$ and $j > i$. Then, consider the submatrix $\mathcal{Q}_{i,j}$ of $\mathcal{Q} = \mathcal{M}\mathcal{N}$, for $\mathcal{M}, \mathcal{N} \in \mathcal{L}$. Clearly, $\mathcal{Q}_{1,1} = 1$ and $\mathcal{Q}_{i,1} = \mathbf{0}$ for $i \in \{2, \ldots, s+1\}$. For the first row, we have $\mathcal{Q}_{1,j} = \sum_{k=1}^{s+1} \mathcal{M}_{1,k}\mathcal{N}_{k,j}$ for $j \in \{2, \ldots, s+1\}$. Note that $\mathcal{M}_{1,1}\mathcal{N}_{1,j} = \mathcal{N}_{1,j}$ is a multiple of p^{j-2}, $\mathcal{M}_{1,k}\mathcal{N}_{k,j}$ is a multiple of $p^{k-2}p^{j-k} = p^{j-2}$ for $k \in \{2, \ldots, j\}$, and a multiple of p^{k-2} for $k \in \{j+1, \ldots, s+1\}$. Therefore, $\mathcal{Q}_{i,j}$ is a multiple of p^{j-2}. For the rest of the rows, $\mathcal{Q}_{i,j} = \sum_{k=2}^{s+1} \mathcal{M}_{i,k}\mathcal{N}_{k,j}$ for $i, j \in \{2, \ldots, s+1\}$ and $j > i$. Note that $\mathcal{M}_{i,k}\mathcal{N}_{k,j}$ is a multiple of p^{j-k} for $k \in \{2, \ldots, i-1\}$, a multiple of $p^{k-i}p^{j-k} = p^{j-i}$ for $k \in \{i, \ldots, j\}$, and a multiple of p^{k-i} for $k \in \{j+1, \ldots, s+1\}$. Therefore, $\mathcal{Q}_{i,j}$ is also a multiple of p^{j-i}. Finally, the block submatrices in the diagonal are $\mathcal{Q}_{i,i} = \sum_{k=2}^{s+1} \mathcal{M}_{i,k}\mathcal{N}_{k,i}$ for $i \in \{2, \ldots, s+1\}$. Note that $\mathcal{M}_{i,i}\mathcal{N}_{i,i} \in \mathrm{GL}(t_i, \mathbb{Z}_{p^s})$ and $\mathcal{M}_{i,k}\mathcal{N}_{k,i}$ is a multiple of p^{i-k} for $k < i$ and a multiple of p^{k-i} for $k > i$, hence $\mathcal{Q}_{i,i} \in \mathrm{GL}(t_i, \mathbb{Z}_{p^s})$. □

Let ζ_i be the map from $\mathbb{Z}_{p^s}$ to $\mathbb{Z}_{p^s}$ defined as $\zeta_i(a) = a \mod p^i$, $i \in \{1, \ldots, s-1\}$. This map can be extended to matrices over $\mathbb{Z}_{p^s}$ by applying

ζ_i to each one of their entries. Let π be the map from $\mathcal{L}$ to $\mathcal{L}$ defined as

$$\pi(\mathcal{M}) = \begin{pmatrix} 1 & a_1 & pa_2 & \cdots & p^{s-2}a_{s-1} & p^{s-1}a_s \\ \mathbf{0} & A_{1,1} & pA_{1,2} & \cdots & p^{s-2}A_{1,s-1} & p^{s-1}A_{1,s} \\ \mathbf{0} & \zeta_{s-1}(A_{2,1}) & \zeta_{s-1}(A_{2,2}) & \cdots & \zeta_{s-1}(p^{s-3}A_{2,s-1}) & \zeta_{s-1}(p^{s-2}A_{2,s}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \zeta_2(A_{s-1,1}) & \zeta_2(A_{s-1,2}) & \cdots & \zeta_2(A_{s-1,s-1}) & \zeta_2(pA_{s-1,s}) \\ \mathbf{0} & \zeta_1(A_{s,1}) & \zeta_1(A_{s,2}) & \cdots & \zeta_1(A_{s,s-1}) & \zeta_1(A_{s,s}) \end{pmatrix}, \tag{4.2}$$

for any matrix $\mathcal{M} \in \mathcal{L}$ as given in (4.1). Let $\pi(\mathcal{L}) = \{\pi(\mathcal{M}) : \mathcal{M} \in \mathcal{L}\} \subseteq \mathrm{GL}(t_1 + \cdots + t_s, \mathbb{Z}_{p^s})$. By Lemma 24, it is clear that $\pi(\mathcal{L})$ is a group with the operation $*$ defined as $\mathcal{M} * \mathcal{N} = \pi(\mathcal{M}\mathcal{N})$ for all $\mathcal{M}, \mathcal{N} \in \pi(\mathcal{L})$. Note that the group operation $*$ is well defined, since $\pi(\mathcal{L}) \subseteq \mathcal{L}$. By a generalization of the proof of Theorem 2 in [KV15], one can show the following theorem.

Theorem 25. *Let $\mathcal{H}^{t_1,\ldots,t_s}$ be the $\mathbb{Z}_{p^s}$-additive GH code of type $(n; t_1, \ldots, t_s)$. Then, $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ is isomorphic to $\pi(\mathcal{L})$.*

Proof. Let R be the set $\mathbb{Z}_{p^s}^{t_1-1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$ and denote by $\mathcal{B}$ the set of all affine functions from R to $\mathbb{Z}_{p^s}$. We can see these $\mathbb{Z}_{p^s}$-valued affine functions as words of length $n = p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$ over $\mathbb{Z}_{p^s}$ by considering the image of all elements in the domain. Let us define $B = \{x : R \longrightarrow \mathbb{Z}_p^{p^{s-1}} \mid x(\cdot) = \phi(f(\cdot)) \text{ for some } f \in \mathcal{B}\}$. That is, the image of the words in $\mathcal{B}$ by the generalized Gray map Φ. By a straightforward generalization of Lemma 1 in [KV15], we know that B is a $\mathbb{Z}_{p^s}$-linear GH code of type $(n; t_1, t_2, \ldots, t_s)$. This means that $\mathcal{H}^{t_1,\ldots,t_s} = \mathcal{B}$ and $H^{t_1,\ldots,t_s} = B$, and we can see the elements of $\mathcal{H}^{t_1,\ldots,t_s}$ as affine functions.

Note that an affine function $f \in \mathcal{B}$ can be seen as a word w_f of length n over $\mathbb{Z}_{p^s}$, with the elements of R playing the role of coordinate positions. A permutation $\sigma \in \mathcal{S}_n$ acting on w_f, by permuting its coordinates, gives a word $\sigma(w_f)$ which corresponds to the function $f \circ \sigma^{-1}$ by considering $f(\sigma^{-1}(v))$ for any $v \in R$. Therefore, a permutation σ is said to be in $\mathrm{PAut}(\mathcal{B})$ if and only if $f \circ \sigma^{-1}$ is an affine function for any $f \in \mathcal{B}$. Naturally, $\mathrm{PAut}(\mathcal{B}) = \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$.

Now, we use an adaptation of Theorem 2 in [KV15] to prove that the group $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, or equivalently $\mathrm{PAut}(\mathcal{B})$, consists of all affine permutations of R. First, we have that any affine permutation belongs to $\mathrm{PAut}(\mathcal{B})$ since the composition of an affine permutation and an affine function is also an affine function. Next, we see that any permutation $\sigma \in \mathrm{PAut}(\mathcal{B})$ is affine. Let

$$\begin{aligned} \sigma_1^{(1)}, \ldots, \sigma_{t_1-1}^{(1)} : &\quad R \longrightarrow \mathbb{Z}_{p^s}, \\ \sigma_1^{(2)}, \ldots, \sigma_{t_2}^{(2)} : &\quad R \longrightarrow \mathbb{Z}_{p^{s-1}}, \\ &\quad \vdots \\ \sigma_1^{(s)}, \ldots, \sigma_{t_s}^{(s)} : &\quad R \longrightarrow \mathbb{Z}_{p}, \end{aligned}$$

be the components of σ^{-1}. That is,

$$\sigma^{-1}(v) = (\sigma_1^{(1)}(v), \ldots, \sigma_{t_1-1}^{(1)}(v), \sigma_1^{(2)}(v), \ldots, \sigma_{t_2}^{(2)}(v), \ldots, \sigma_1^{(s)}(v), \ldots, \sigma_{t_s}^{(s)}(v))$$

for any $v \in R$. Consider the following functions defined from R to $\mathbb{Z}_{p^s}$:

$$f_i^{(j)}(x_1^{(1)}, \ldots, x_{t_1-1}^{(1)}, x_1^{(2)}, \ldots, x_{t_2}^{(2)}, \ldots, x_1^{(s)}, \ldots, x_{t_s}^{(s)}) = p^{j-1} x_i^{(j)},$$
$$\begin{cases} \text{for } i \in \{1, \ldots, t_1 - 1\} \text{ if } j = 1, \\ \text{for } i \in \{1, \ldots, t_j\} \text{ if } j \in \{2, \ldots, s\}. \end{cases}$$

Note that $p^{j-1}x_i^{(j)}$ defines the inclusion of $x_i^{(j)} \in \mathbb{Z}_{p^{s-j+1}}$ in $\mathbb{Z}_{p^s}$. These functions are affine, hence $f_i^{(j)} \in \mathcal{B}$ and, since $\sigma \in \mathrm{PAut}(\mathcal{B})$, we have that $f_i^{(j)} \circ \sigma^{-1} \in \mathcal{B}$. Moreover, $f_i^{(j)}(\sigma^{-1}(v)) = p^{j-1}\sigma_i^{(j)}(v)$, therefore $\sigma_i^{(j)}$ is affine. Since all components are affine, σ^{-1} and σ are also affine.

Finally, we show that the group of affine permutations over R is isomorphic to $\pi(\mathcal{L})$. Let us denote the former by $\mathcal{S}$. Then, we see that $\mathcal{S}$ is isomorphic to $\pi(\mathcal{L})$ via the map ψ, defined from $\pi(\mathcal{L})$ to $\mathcal{S}$ as

$$\psi(\pi(\mathcal{M})) = \sigma(u_1, u_2, \ldots, u_s) = b + u_1 A_1 + u_2 p A_2 + \cdots + u_s p^{s-1} A_s,$$

where $b = (a_1, pa_2, \ldots, p^{s-1}a_s)$ and

$$\begin{aligned} A_1 &= (A_{1,1}, pA_{1,2}, \ldots, p^{s-1}A_{1,s}),\\ A_2 &= (\zeta_{s-1}(A_{2,1}), \zeta_{s-1}(A_{2,2}), \ldots, \zeta_{s-1}(p^{s-2}A_{2,s})),\\ &\vdots\\ A_s &= (\zeta_1(A_{s,1}), \zeta_1(A_{s,2}), \ldots, \zeta_1(A_{s,s})). \end{aligned}$$

Note that $A_1, pA_2, \ldots, p^{s-1}A_s$ are matrices over $\mathbb{Z}_{p^s}$ with linearly independent rows of order $p^s, p^{s-1}, \ldots, p$, respectively, spanning R. This is ensured due to $A_{j,j}$, for $j \in \{1, \ldots, s\}$, being invertible. The map ψ gives an isomorphism between $\pi(\mathcal{L})$ and $\mathcal{S}$, so $\pi(\mathcal{L})$ and $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ are isomorphic. □

Theorem 26. *Let $\mathcal{H}^{t_1,\ldots,t_s}$ be the $\mathbb{Z}_{p^s}$-additive GH code of type $(n; t_1, \ldots, t_s)$, where $n = p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$. Let $\bar{t}_1 = t_1 - 1$ and $\bar{t}_i = t_i$ for $i \in \{2, \ldots, s\}$. The order of its permutation automorphism group is*

$$|\,\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})| = p^E N_1 \cdots N_s, \tag{4.3}$$

where $N_i = |\mathrm{GL}(\bar{t}_i, \mathbb{Z}_{p^{s-i+1}})| = p^{(s-i)\bar{t}_i^2 + \frac{\bar{t}_i(\bar{t}_i-1)}{2}} \prod_{j=1}^{\bar{t}_i}(p^j - 1)$ and

$$E = s\bar{t}_1 + (s-1)\bar{t}_2 + \cdots + \bar{t}_s + \sum_{i=1}^{s}\sum_{j=i+1}^{s} 2(s-j+1)\bar{t}_i\bar{t}_j. \tag{4.4}$$

Proof. The order of $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ can be easily computed with a counting argument over the matrix representation, that is, over the elements of $\pi(\mathcal{L})$, given in (4.2).

The first row of a matrix $\mathcal{M} \in \pi(\mathcal{L})$ is a random tuple over $\mathbb{Z}_{p^s}^{t_1-1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$. There are $p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$ such tuples.

Note that $\zeta_{s-i+1}(p^{j-i}A_{i,j})$, for $i < j$, is a matrix of size $\bar{t}_i \times \bar{t}_j$ defined over $\mathbb{Z}_{p^s}$, with entries among $p^{s-i+1}/p^{j-i} = p^{s-j+1}$ elements in $\mathbb{Z}_{p^s}$. In the case $i = 1$, ζ_s represents the identity map from $\mathbb{Z}_{p^s}$ to $\mathbb{Z}_{p^s}$. Therefore, there are $p^{(s-j+1)\bar{t}_i\bar{t}_j}$ such matrices. Moreover, $\zeta_{s-j+1}(A_{j,i})$, for $i < j$, is a matrix of size $\bar{t}_j \times \bar{t}_i$ with entries among the same number of elements as $\zeta_{s-i+1}(p^{j-i}A_{i,j})$. Then, for each pair $i, j \in \{1, \ldots, s\}$ such that $i < j$, we have $p^{2(s-j+1)\bar{t}_i\bar{t}_j}$

different possibilities to choose the corresponding matrices.

All matrices in the diagonal are invertible matrices defined over $\mathbb{Z}_{p^s}$. Moreover, $\zeta_{s-i+1}(A_{i,i})$ can be represented as an invertible matrix over $\mathbb{Z}_{p^{s-i+1}}$ by considering $\{0,\dots,p^{s-i+1}-1\} \subset \mathbb{Z}_{p^s}$ as elements in $\mathbb{Z}_{p^{s-i+1}}$. Therefore, $\zeta_{s-i+1}(A_{i,i})$ is a matrix in the group $\mathrm{GL}(\bar{t}_i, \mathbb{Z}_{p^{s-i+1}})$. In [Han06], the order of the general linear group over integers modulo m is given. In particular, the order of $\mathrm{GL}(\kappa, \mathbb{Z}_{p^s})$, denoted by $\nu_\kappa(p^s)$, for p prime and integers κ and $s \geq 2$, satisfies $\nu_\kappa(p^s) = p^{(s-1)\kappa^2}\nu_\kappa(p)$, where $\nu_\kappa(p) = (p^\kappa - 1)(p^\kappa - p)\cdots(p^\kappa - p^{\kappa-1})$ is the order of the general linear group over the field $\mathbb{Z}_p$. Then, the order of $\mathrm{GL}(\bar{t}_i, \mathbb{Z}_{p^{s-i+1}})$ is $\nu_{\bar{t}_i}(\mathbb{Z}_{p^{s-i+1}}) = p^{(s-i)\bar{t}_i^2 + \frac{\bar{t}_i(\bar{t}_i-1)}{2}} \prod_{j=1}^{\bar{t}_i}(p^j - 1)$.

Considering all possible choices of submatrices, the result follows. □

Remark 27. *If we consider the case with $p = 2$ and $s = 2$, that is, $\mathbb{Z}_4$-additive Hadamard codes of type $(n; t_1, t_2)$, then (4.4) becomes $E = 2(t_1 - 1) + t_2 + 2(t_1 - 1)t_2$. We also have*

$$N_1 = |\mathrm{GL}(t_1 - 1, \mathbb{Z}_4)| = 2^{(t_1-1)^2 + \frac{(t_1-1)(t_1-2)}{2}} \prod_{j=1}^{t_1-1}(2^j - 1),$$

$$N_2 = |\mathrm{GL}(t_2, \mathbb{Z}_2)| = 2^{\frac{t_2(t_2-1)}{2}} \prod_{j=1}^{t_2}(2^j - 1).$$

Therefore,

$$|\,\mathrm{PAut}(\mathcal{H}^{t_1,t_2})| = 2^{\frac{3(t_1-1)^2}{2} + \frac{3(t_1-1)}{2} + 2(t_1-1)t_2 + \frac{t_2^2}{2} + \frac{t_2}{2}} \prod_{j=1}^{t_1-1}(2^j - 1) \prod_{j=1}^{t_2}(2^j - 1).$$

Note that this expression coincides with the one given in [BV18].

Example 28. *Consider the $\mathbb{Z}_{27}$-additive GH code $\mathcal{H}^{2,1,1}$. By Theorem 25,* $\mathrm{PAut}(\mathcal{H}^{2,1,1})$ *is isomorphic to the group $\pi(\mathcal{L}) \subseteq \mathrm{GL}(4, \mathbb{Z}_{27})$. The subgroup*

$\mathcal{L} \subseteq \mathrm{GL}(4, \mathbb{Z}_{27})$ is formed by all matrices in the form

$$\begin{pmatrix} 1 & a_1 & 3a_2 & 9a_3 \\ 0 & A_{1,1} & 3A_{1,2} & 9A_{1,3} \\ 0 & A_{2,1} & A_{2,2} & 3A_{2,3} \\ 0 & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix},$$

where $a_1, a_2, a_3, A_{i,j} \in \mathbb{Z}_{27}$, $i, j \in \{1, 2, 3\}$ with $i \neq j$, and $A_{1,1}, A_{2,2}, A_{3,3} \in \mathbb{Z}_{27} \setminus \{0, 3, 6, 9, 12, 15, 18, 21, 24\}$. For example, consider the following two matrices $\mathcal{M}, \mathcal{N} \in \mathcal{L}$:

$$\mathcal{M} = \begin{pmatrix} 1 & 1 & 21 & 9 \\ 0 & 2 & 3 & 18 \\ 0 & 14 & 14 & 0 \\ 0 & 9 & 16 & 22 \end{pmatrix}, \quad \mathcal{N} = \begin{pmatrix} 1 & 19 & 18 & 0 \\ 0 & 8 & 24 & 9 \\ 0 & 18 & 20 & 0 \\ 0 & 16 & 4 & 7 \end{pmatrix}.$$

The function π reduces the third row modulo 9 and the fourth row modulo 3, therefore

$$\pi(\mathcal{M}) = \begin{pmatrix} 1 & 1 & 21 & 9 \\ 0 & 2 & 3 & 18 \\ 0 & 5 & 5 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \pi(\mathcal{N}) = \begin{pmatrix} 1 & 19 & 18 & 0 \\ 0 & 8 & 24 & 9 \\ 0 & 0 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Note that, since $\mathcal{L}$ is a group, $\pi(\mathcal{L})$ is also a group with the operation $*$:

$$\pi(\mathcal{M}) * \pi(\mathcal{N}) = \pi(\pi(\mathcal{M})\pi(\mathcal{N})) = \pi\left(\begin{pmatrix} 1 & 9 & 12 & 18 \\ 0 & 7 & 18 & 9 \\ 0 & 13 & 22 & 18 \\ 0 & 1 & 3 & 1 \end{pmatrix}\right) = \begin{pmatrix} 1 & 9 & 12 & 18 \\ 0 & 7 & 18 & 9 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Theorem 26 gives the order of $\pi(\mathcal{L})$, or equivalently of $\mathrm{PAut}(\mathcal{H}^{2,1,1})$. Following the same notation as in the statement of the theorem, we have that

$$N_1 = |\mathrm{GL}(1, \mathbb{Z}_{27})| = 18,$$

$$N_2 = |\mathrm{GL}(1, \mathbb{Z}_9)| = 6,$$
$$N_3 = |\mathrm{GL}(1, \mathbb{Z}_3)| = 2,$$
$$E = 3 + 2 + 1 + 4 + 2 + 2 = 14.$$

Therefore, $|\mathrm{PAut}(\mathcal{H}^{2,1,1})| = 3^E N_1 N_2 N_3 = 3^{17} \cdot 2^3 = 1033121304$.

4.2 r-PD-sets for $\mathbb{Z}_{p^s}$-linear GH codes

In this section, we give an additive information set for the $\mathbb{Z}_{p^s}$-additive GH code $\mathcal{H}^{t_1,\ldots,t_s}$, and an information set for the corresponding $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,\ldots,t_s}$. Then, using the result given by Theorem 25, we establish a criterion in order to find r-PD-sets of size $r+1$ for $H^{t_1,\ldots,t_s}$, with r up to a given upper bound.

An ordered set $\mathcal{I} = \{i_1, \ldots, i_{t_1+\cdots+t_s}\} \subseteq \{1, \ldots, n\}$ of $t_1 + \cdots + t_s$ coordinate positions is said to be an *additive information set* for a $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$ of type $(n; t_1, \ldots, t_s)$ if $|\mathcal{C}_{\mathcal{I}}| = (p^s)^{t_1}(p^{s-1})^{t_2} \cdots p^{t_s}$. If the elements of $\mathcal{I}$ are ordered in such a way that, for any $k \in \{1, \ldots, s\}$, $|\mathcal{C}_{\{i_1,\ldots,i_{t_1+\cdots+t_k}\}}| = (p^s)^{t_1}(p^{s-1})^{t_2} \cdots (p^{s-k+1})^{t_k}$, then it can be seen that the set $\Phi(\mathcal{I})$, defined as

$$\begin{aligned}\Phi(\mathcal{I}) =& \Phi^{(1)}(\{i_1, \ldots, i_{t_1}\}) \cup \Phi^{(2)}(\{i_{t_1+1}, \ldots, i_{t_1+t_2}\}) \cup \\ & \cdots \cup \Phi^{(s)}(\{i_{t_1+\cdots+t_{s-1}+1}, \ldots, i_{t_1+\cdots+t_s}\}),\end{aligned} \tag{4.5}$$

where

$$\begin{aligned}\Phi^{(k)}(I) = \bigcup_{i \in I} \{& p^{s-1}(i-1) + 1, \\ & p^{s-1}(i-1) + p^{k-1} + 1, \\ & p^{s-1}(i-1) + p^{k-1+1} + 1, \\ & p^{s-1}(i-1) + p^{k-1+2} + 1, \\ & \ldots, \\ & p^{s-1}(i-1) + p^{s-2} + 1\},\end{aligned}$$

is an information set for $C = \Phi(\mathcal{C})$, as proven in Theorem 19. Note that

$s-2-(k-1)=s-k-1$, hence $\Phi^{(k)}(I)$ has $s-k+1$ coordinate positions for each element in I.

Example 29. *It is easy to see, from the matrix $\mathcal{G}^{1,1,1}$ given in Example 8, that the set $\mathcal{I}=\{1,2,10\}$ is an additive information set for the $\mathbb{Z}_{27}$-additive GH code $\mathcal{H}^{1,1,1}$, so $\Phi(\mathcal{I})=\Phi^{(1)}(\{1\})\cup\Phi^{(2)}(\{2\})\cup\Phi^{(3)}(\{10\})=\{1,2,4,10,13,82\}$ is an information set for $H^{1,1,1}=\Phi(\mathcal{H}^{1,1,1})$.*

In general, there is no unique way to obtain an additive information set for $\mathcal{H}^{t_1,\ldots,t_s}$. The following result provides a recursive and simple form to obtain such a set.

Proposition 30. *Let $\mathcal{I}$ be an additive information set for the $\mathbb{Z}_{p^s}$-additive GH code $\mathcal{H}^{t_1,\ldots,t_s}$ of type $(n;t_1,\ldots,t_s)$, where $n=p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$. Then $\mathcal{I}\cup\{n+1\}$ is an additive information set for each of the codes $\mathcal{H}^{t_1+1,t_2,\ldots,t_s}$, $\mathcal{H}^{t_1,t_2+1,\ldots,t_s}$, $\ldots$, $\mathcal{H}^{t_1,t_2,\ldots,t_s+1}$, obtained from $\mathcal{H}^{t_1,t_2,\ldots,t_s}$ by applying (2.21).*

Proof. Let $\mathcal{H}_k=\mathcal{H}^{t'_1,t'_2,\ldots,t'_s}$, $k\in\{1,\ldots,s\}$, where $t'_j=t_j$ for $j\neq k$ and $t'_k=t_k+1$. It is clear that an additive information set for $\mathcal{H}_k$ should have $t_1+t_2+\cdots+t_s+1=|\mathcal{I}|+1$ coordinate positions. Taking into account that $\mathcal{H}_k$ is constructed from $\mathcal{H}^{t_1,t_2,\ldots,t_s}$ by applying (2.21), we have that $|(\mathcal{H}_k)_{\mathcal{I}\cup\{x\}}|=(p^s)^{t_1}(p^{s-1})^{t_2}\cdots p^{t_s}p^{s+1-k}$ for all $x\in\{n+1,\ldots,2n\}$. In particular, $\mathcal{I}\cup\{n+1\}$ is an additive information set for $\mathcal{H}_k$. □

Let $\mathcal{I}$ be an additive information set for $\mathcal{H}^{t_1,\ldots,t_s}$ of type $(n;t_1,\ldots,t_s)$. Let $\mathcal{H}_k=\mathcal{H}^{t'_1,t'_2,\ldots,t'_s}$, $k\in\{1,\ldots,s\}$, where $t'_j=t_j$ for $j\neq k$ and $t'_k=t_k+1$. Although the additive information set $\mathcal{I}\cup\{n+1\}$, given by Proposition 30, is the same for all $\mathcal{H}_k$, the information sets for the corresponding $\mathbb{Z}_{p^s}$-linear codes over $\mathbb{Z}_p$, $H_k=\Phi(\mathcal{H}_k)$, differ for every $k\in\{1,\ldots,s\}$. In particular,

$$I^{(k)}=\Phi(\mathcal{I})\cup\{p^{s-1}n+1,p^{s-1}n+p^{k-1}+1,p^{k-1}n+p^k+1,\ldots,p^{s-1}n+p^{s-2}+1\}$$

is an information set for H_k.

We can label the ith coordinate position of a $\mathbb{Z}_{p^s}$-additive GH code $\mathcal{H}^{t_1,\ldots,t_s}$, with the ith column of its generator matrix $\mathcal{G}^{t_1,\ldots,t_s}$. Note that, by construction, all columns in $\mathcal{G}^{t_1,\ldots,t_s}$ are different and there are $n=$

$p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$ of them. Thus, any additive information set $\mathcal{I}$ for $\mathcal{H}^{t_1,\ldots,t_s}$ can also be considered as a set of vectors representing the positions in $\mathcal{I}$. Let e_i be the vector with all coordinates equal to 0 except the one in the ith position, which is equal to 1. Then, by Proposition 30, we have that the set

$$\begin{aligned}\mathcal{I}_{t_1,\ldots,t_s} =&\{e_1, e_1+e_2, \ldots, e_1+e_{t_1}\} \cup \{e_1+pe_{t_1+1}, \ldots, e_1+pe_{t_1+t_2}\} \cup \cdots \cup \\ &\{e_1+p^{s-1}e_{t_1+t_2+\cdots+t_{s-1}+1}, \ldots, e_1+p^{s-1}e_{t_1+t_2+\cdots+t_s}\}\end{aligned}$$

is a suitable additive information set for $\mathcal{H}^{t_1,\ldots,t_s}$. Depending on the context, $\mathcal{I}_{t_1,\ldots,t_s}$ is considered as a subset of $\{1,\ldots,n\}$ or as a subset $\{1\} \times \mathbb{Z}_{p^s}^{t_1-1} \times (p\mathbb{Z}_{p^s})^{t_2} \times \cdots \times (p^{s-1}\mathbb{Z}_{p^s})^{t_s}$.

Example 31. *Let $\mathcal{H}^{2,0,0}$ be the $\mathbb{Z}_{27}$-additive GH code of length 27 with generator matrix $\mathcal{G}^{2,0,0}$ given in Example 8. The set $\mathcal{I}_{2,0,0} = \{1,2\}$, or equivalently the set of vectors $\mathcal{I}_{2,0,0} = \{e_1, e_1+e_2\}$, is an additive information set for $\mathcal{H}^{2,0,0}$.*

By applying (2.21) over $\mathcal{G}^{2,0,0}$, we obtain matrices $\mathcal{G}^{3,0,0}$, $\mathcal{G}^{2,1,0}$ and $\mathcal{G}^{2,0,1}$ that generate the $\mathbb{Z}_{27}$-additive GH codes $\mathcal{H}^{3,0,0}$, $\mathcal{H}^{2,1,0}$ and $\mathcal{H}^{2,0,1}$ of length 729, 243 and 81, respectively. By Proposition 30, it follows that $\mathcal{I}_{2,0,0} \cup \{28\} = \{1,2,28\}$ is an additive information set for $\mathcal{H}^{3,0,0}$, $\mathcal{H}^{2,1,0}$ and $\mathcal{H}^{2,0,1}$. Although this additive information set is the same for these three codes, it is important to note that in terms of vectors representing these positions, we have that

$$\begin{aligned}\mathcal{I}_{3,0,0} &= \{(1,0,0),(1,1,0),(1,0,1)\},\\ \mathcal{I}_{2,1,0} &= \{(1,0,0),(1,1,0),(1,0,3)\}, \text{ and}\\ \mathcal{I}_{2,0,1} &= \{(1,0,0),(1,1,0),(1,0,9)\}.\end{aligned}$$

Finally,

$$\begin{aligned}I^{(1)} &= \Phi(\mathcal{I}_{2,0,0}) \cup \{244,245,247\} = \{1,2,4,10,11,13,244,245,247\},\\ I^{(2)} &= \Phi(\mathcal{I}_{2,0,0}) \cup \{244,247\} = \{1,2,4,10,11,13,244,247\}, \text{ and}\\ I^{(3)} &= \Phi(\mathcal{I}_{2,0,0}) \cup \{244\} = \{1,2,4,10,11,13,244\}\end{aligned}$$

are information sets for the corresponding $\mathbb{Z}_{27}$-linear GH codes $H^{3,0,0}$, $H^{2,1,0}$

and $H^{2,0,1}$, respectively.

Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of type $(n; t_1, \ldots, t_s)$, and let $C = \Phi(\mathcal{C})$ be the corresponding $\mathbb{Z}_{p^s}$-linear code of length $p^{s-1}n$. Let $\Phi : \mathrm{Sym}(n) \longrightarrow \mathrm{Sym}(p^{s-1}n)$ be the map defined as

$$\Phi(\tau)(i) = p^{s-1}\tau\left(\frac{i+\chi(i)}{p^{s-1}}\right) - \chi(i), \tag{4.6}$$

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

Lemma 32. *The map $\Phi : \mathrm{Sym}(n) \longrightarrow \mathrm{Sym}(p^{s-1}n)$ is a group monomorphism.*

Proof. We need to check that $\Phi(\sigma\tau) = \Phi(\sigma)\Phi(\tau)$ for all $\tau, \sigma \in \mathrm{Sym}(n)$. Let i be a coordinate position in $\{1, \ldots, p^{s-1}n\}$ and $\chi(i) = p^{s-1} - (i \bmod p^{s-1})$. Note that $\chi(p^{s-1}a - \chi(i)) = \chi(i)$ for any integer a. Then

$$\begin{aligned}
(\Phi(\sigma)\Phi(\tau))(i) &= \Phi(\sigma)\left(p^{s-1}\tau\left(\frac{i+\chi(i)}{p^{s-1}}\right) - \chi(i)\right) \\
&= p^{s-1}\sigma\left(\frac{p^{s-1}\tau\left(\frac{i+\chi(i)}{p^{s-1}}\right) - \chi(i) + \chi(i)}{p^{s-1}}\right) - \chi(i) \\
&= p^{s-1}\sigma\tau\left(\frac{i+\chi(i)}{p^{s-1}}\right) - \chi(i) \\
&= \Phi(\sigma\tau)(i).
\end{aligned}$$

Finally, it is easy to check that Φ is injective. □

By Theorem 25, we identify $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ with the group $\pi(\mathcal{L})$. Recall that we can label the ith coordinate position of $\mathcal{H}^{t_1,\ldots,t_s}$ with the ith column w_i of the generator matrix $\mathcal{G}^{t_1,\ldots,t_s}$ constructed via (2.21), $i \in \{1, \ldots, n\}$. Any matrix $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ sends columns of $\mathcal{G}^{t_1,\ldots,t_s}$ to other columns

of $\mathcal{G}^{t_1,\dots,t_s}$. Therefore, $\mathcal{M}$ can be seen as a permutation of coordinate positions $\tau \in \mathrm{Sym}(n)$, such that for all $i \in \{1,\dots,n\}$

$$\tau(i) = j \iff w_i\mathcal{M} = w_j, \quad j \in \{1,\dots,n\}. \tag{4.7}$$

For any $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$, we define $\Phi(\mathcal{M}) = \Phi(\tau) \in \mathrm{Sym}(p^{s-1}n)$ and, for any $\mathcal{P} \subseteq \mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$, we consider $\Phi(\mathcal{P}) = \{\Phi(\mathcal{M}) : \mathcal{M} \in \mathcal{P}\} \subseteq \mathrm{Sym}(p^{s-1}n)$.

Proposition 33. *Let $\mathcal{H}^{t_1,\dots,t_s}$ be the $\mathbb{Z}_{p^s}$-additive GH code of type $(n; t_1, \dots, t_s)$ and let $\mathcal{P} \subseteq \mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$. Then, $\Phi(\mathcal{P})$ is an r-PD-set for $H^{t_1,\dots,t_s}$ with information set $\Phi(\mathcal{I}_{t_1,\dots,t_s})$ if and only if for each r-set $\mathcal{E}$ of column vectors of $\mathcal{G}^{t_1,\dots,t_s}$ there is $\mathcal{M} \in \mathcal{P}$ such that $\{g\mathcal{M} : g \in \mathcal{E}\} \cap \mathcal{I}_{t_1,\dots,t_s} = \emptyset$.*

Proof. If $\Phi(\mathcal{P})$ is an r-PD-set with respect to the information set $\Phi(\mathcal{I}_{t_1,\dots,t_s})$, then, for each r-set $E \subseteq \{1,\dots,p^{s-1}n\}$, there is $\tau \in \mathcal{P} \subseteq \mathrm{Sym}(n)$ such that $\Phi(\tau)(E) \cap \Phi(\mathcal{I}_{t_1,\dots,t_s}) = \emptyset$. For every r-set $\mathcal{E} \subseteq \{1,\dots,n\}$, let $E_o = \{p^{s-1}(i-1)+1 : i \in \mathcal{E}\}$. We know that there is $\tau \in \mathcal{P}$ such that $\Phi(\tau)(E_o) \cap \Phi(\mathcal{I}_{t_1,\dots,t_s}) = \emptyset$. By the definition of Φ, we also have that $\tau(\mathcal{E}) \cap \mathcal{I}_{t_1,\dots,t_s} = \emptyset$, which is equivalent to the statement.

Conversely, we assume that for each r-set $\mathcal{E} \subseteq \{1,\dots,n\}$, there is $\tau \in \mathcal{P} \subseteq \mathrm{Sym}(n)$ such that $\tau(\mathcal{E}) \cap \mathcal{I}_{t_1,\dots,t_s} = \emptyset$. For every r-set $E \subseteq \{1,\dots,p^{s-1}n\}$, let $\mathcal{E}_o \subseteq \{1,\dots,n\}$ be an r-set such that $\{i : \exists k \in \{1,\dots,p^{s-1}\} \text{ s.t. } \varphi_k(i) \in E\} \subseteq \mathcal{E}_o$, where $\varphi_k(i) = p^{s-1}(i-1)+k$. Since there is $\tau \in \mathcal{P}$ such that $\tau(\mathcal{E}_o) \cap \mathcal{I}_{t_1,\dots,t_s} = \emptyset$, we have that $\Phi(\tau)(E) \cap \Phi(\mathcal{I}_{t_1,\dots,t_s}) = \emptyset$. □

A slight modification of the proof of Proposition 33 leads to a more general result that holds for any $\mathbb{Z}_{p^s}$-additive code, not only for the family of $\mathbb{Z}_{p^s}$-additive GH codes.

Proposition 34. *Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code, let $\mathcal{I}$ be an additive information set for $\mathcal{C}$, and let $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{C})$. Then, $\mathcal{S}$ satisfies that for each r-set $\mathcal{E} \subseteq \{1,\dots,n\}$ 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 r-set $E \subseteq \{1,\dots,p^{s-1}n\}$ there is $\sigma \in \Phi(\mathcal{S})$ such that $\sigma(E) \cap \Phi(\mathcal{I}) = \emptyset$.*

Definition 35. *Let $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ and let m_i be the ith row of $\mathcal{M}$, $i \in \{1,\ldots,t_1+\cdots+t_s\}$. We define $\mathcal{M}^*$ over $\mathbb{Z}_{p^s}$ as the matrix where the first row is m_1 and the ith row is m_1+m_i for $i \in \{2,\ldots,t_1\}$, m_1+pm_i for $i \in \{t_1+1,\ldots,t_1+t_2\}$, $m_1+p^2m_i$ for $i \in \{t_1+t_2+1,\ldots,t_1+t_2+t_3\}$ and so on until $m_1+p^{s-1}m_i$ for $i \in \{t_1+\cdots+t_{s-1}+1,\ldots,t_1+\cdots+t_s\}$.*

Theorem 36. *Let $\mathcal{H}^{t_1,\ldots,t_s}$ be the $\mathbb{Z}_{p^s}$-additive GH code of type $(n;t_1,\ldots,t_s)$. Let $\mathcal{P}_r=\{\mathcal{M}_i : 0 \le i \le r\}$ be a set of $r+1$ matrices in $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$. Then, $\Phi(\mathcal{P}_r)$ is an r-PD-set of size $r+1$ for $H^{t_1,\ldots,t_s}$ with information set $\Phi(\mathcal{I}_{t_1,\ldots,t_s})$ if and only if no two matrices $(\mathcal{M}_i^{-1})^*$ and $(\mathcal{M}_j^{-1})^*$ have a row in common, for $i,j \in \{0,\ldots,r\}$ and $i \neq j$.*

Proof. The result can be proved using Proposition 33 and is a generalization of a similar result given in [BV18] for $\mathbb{Z}_4$-linear Hadamard codes. However, we include the detailed proof for the convenience of the reader.

Suppose that the set $\mathcal{P}_r=\{\mathcal{M}_i : 0 \le i \le r\}$ satisfies that no two matrices $(\mathcal{M}_i^{-1})^*$ and $(\mathcal{M}_j^{-1})^*$, for $i,j \in \{0,\ldots,r\}$ and $i \neq j$, have a row in common. Assume that $\Phi(\mathcal{P}_r)$ is not an r-PD-set for $H^{t_1,\ldots,t_s}$ with information set $\Phi(\mathcal{I}_{t_1,\ldots,t_s})$. By Proposition 33, it follows that there exists an r-set $\mathcal{E} \subseteq \{1\} \times \mathbb{Z}_{p^s}^{t_1-1} \times (p\mathbb{Z}_{p^s})^{t_2} \times \cdots \times (p^{s-1}\mathbb{Z}_{p^s})^{t_s}$, that is, a set of r different column vectors of the generator matrix $\mathcal{G}^{t_1,\ldots,t_s}$, such that for each $i \in \{0,\ldots,r\}$, there is a $g_i \in \mathcal{E}$ so that $g_i\mathcal{M}_i \in \mathcal{I}_{t_1,\ldots,t_s}$. Note that there are $r+1$ values for i, but only r elements in $\mathcal{E}$. Therefore, $g\mathcal{M}_i \in \mathcal{I}_{t_1,\ldots,t_s}$ and $g\mathcal{M}_j \in \mathcal{I}_{t_1,\ldots,t_s}$ for some $g \in \mathcal{E}$ and $i \neq j$. Suppose $g\mathcal{M}_i = w_h$ and $g\mathcal{M}_j = w_t$, for $w_h, w_t \in \mathcal{I}_{t_1,\ldots,t_s}$. Then, $g = w_h\mathcal{M}_i^{-1} = w_t\mathcal{M}_j^{-1}$. Taking into account the form of the vectors in the information set $\mathcal{I}_{t_1,\ldots,t_s}$, by multiplying for such inverse matrices $\mathcal{M}_i^{-1}$ and $\mathcal{M}_j^{-1}$, we obtain 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 $\mathcal{P}_k \subseteq \mathcal{P}_r$ 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 $\Phi(\mathcal{P}_r)$ is an r-PD-set for $H^{t_1,\ldots,t_s}$ with information set $\Phi(\mathcal{I}_{t_1,\ldots,t_s})$, but does not satisfy the condition on the inverse matrices. Thus, there are two matrices $(\mathcal{M}_i^{-1})^*$ and $(\mathcal{M}_j^{-1})^*$, with $i,j \in \{0,\ldots,r\}$,

such that they share a common row, say the hth row of $(\mathcal{M}_i^{-1})^*$ and the tth row of $(\mathcal{M}_i^{-1})^*$, with $h, t \in \{1, \ldots, t_1 + \cdots + t_s\}$. In other words, we can define $g = e_h(\mathcal{M}_i^{-1})^* = e_t(\mathcal{M}_j^{-1})^*$. Therefore, $g = w_h\mathcal{M}_i^{-1} = w_t\mathcal{M}_j^{-1}$, where $w_h, w_t \in \mathcal{I}_{t_1,\ldots,t_s}$. Finally, we obtain that $g\mathcal{M}_i = w_h$ and $g\mathcal{M}_j = w_t$. Let $L = \{\ell \ : \ 0 \leq \ell \leq r, \ \ell \neq i, j\}$. For each $\ell \in L$, choose a row g_ℓ of the matrix $(\mathcal{M}_\ell^{-1})^*$. It is clear that $g_\ell = e_{h_\ell}(\mathcal{M}_\ell^{-1})^* = w_{h_\ell}\mathcal{M}_\ell^{-1}$, so $g_\ell\mathcal{M}_\ell = w_{h_\ell} \in \mathcal{I}_{t_1,\ldots,t_s}$. Finally, since some of the g_ℓ may repeat, we obtain a set $\mathcal{E} = \{g_\ell \ : \ \ell \in L\} \cup \{g\}$ of size at most r. Nevertheless, no matrix in $\mathcal{P}_r$ will map every member of $\mathcal{E}$ out of the additive information set $\mathcal{I}_{t_1,\ldots,t_s}$, which contradicts our assumption by Proposition 33. □

Corollary 37. *Let $\mathcal{P}_r$ be a set of $r+1$ matrices in* $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$. *If $\Phi(\mathcal{P}_r)$ is an r-*PD*-set of size $r+1$ for $H^{t_1,\ldots,t_s}$, then any ordering of elements in $\Phi(\mathcal{P}_r)$ provides nested k-*PD*-sets for $k \in \{1, \ldots, r\}$.*

Corollary 38. *Let $\mathcal{P}_r$ be a set of $r+1$ matrices in* $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$. *If $\Phi(\mathcal{P}_r)$ is an r-*PD*-set of size $r+1$ for $H^{t_1,\ldots,t_s}$, then $r \leq f_p^{t_1,\ldots,t_s}$, where*

$$f_p^{t_1,\ldots,t_s} = \left\lfloor \frac{p^{st_1+(s-1)t_2+\cdots+t_s-s} - t_1 - t_2 - \cdots - t_s}{t_1 + t_2 + \cdots + t_s} \right\rfloor.$$

Proof. Following the condition on sets of matrices to be r-PD-sets of size $r+1$, given by Theorem 36, we have to obtain certain $r+1$ matrices with no rows in common. Since the rows of length $t_1 + \cdots + t_s$ must have 1 in the first coordinate, elements from $\mathbb{Z}_{p^s}$ in the coordinates from 2 to t_1, and elements from $p^i\mathbb{Z}_{p^s}$ in the coordinates from $t_1 + \cdots + t_i + 1$ to $t_1 + \cdots + t_{i+1}$, for $i \in \{1, \ldots, s-1\}$, the number of possible rows is $p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$. Thus, taking this fact into account and counting the number of rows of each one of these $r+1$ matrices, we have that $(r+1)(t_1 + t_2 + \cdots + t_s) \leq p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$, and the result follows. □

4.3 Explicit construction of r-PD-sets of size $r+1$

In this section, by using Theorem 36, we create r-PD-sets of size $r+1$ for different infinite families of $\mathbb{Z}_{p^s}$-linear GH codes. First, we give an explicit construction for the $\mathbb{Z}_{p^s}$-linear GH codes $H^{t_1,0,\dots,0}$, with $t_1 \geq 2$ and $r \leq f_p^{t_1,0,\dots,0}$. Then, using a similar idea, we give an explicit construction for the $\mathbb{Z}_{p^s}$-linear GH codes $H_i = H^{1,t_2,\dots,t_s}$, where $i \in \{2,\dots,s\}$, $t_j = 0$ for all $j \neq i$, $t_i \geq 1$, and $r \leq f_p^{1,t_2,\dots,t_s}$. The main idea behind these constructions is to use a certain ordered set of vectors as rows of a set of matrices $\{\mathcal{N}_0^*,\dots,\mathcal{N}_r^*\}$, such that $\{\mathcal{N}_0^{-1},\dots,\mathcal{N}_r^{-1}\}$ is an r-PD-set. This method is a generalization of the one used in [BV18] for $\mathbb{Z}_4$-linear Hadamard codes, which, in turn, was based on a similar idea for simplex codes given in [FKM12].

Let $\mathcal{R} = \mathrm{GR}(p^{s(t_1-1)})$ be the *Galois extension* of dimension t_1-1 over $\mathbb{Z}_{p^s}$, which is isomorphic to any ring $\mathbb{Z}_{p^s}[x]/(h(x))$, where $h(x)$ is a monic basic irreducible polynomial over $\mathbb{Z}_{p^s}$ of degree t_1-1. A monic basic polynomial $h(x)$ over $\mathbb{Z}_{p^s}$ is called *irreducible* if $\bar{h}(x)$ is an irreducible polynomial over $\mathbb{Z}_p$, where $\bar{h}(x)$ is the polynomial obtained by taking the coefficients of $h(x)$ modulo p. Moreover, if $\bar{h}(x)$ is primitive, then $h(x)$ is said to be a *monic basic primitive polynomial* over $\mathbb{Z}_{p^s}$. If $f(x)$ is an irreducible polynomial dividing x^n-1 in $\mathbb{Z}_p[x]$, then there is a unique polynomial $h(x)$ over $\mathbb{Z}_{p^s}[x]$ that satisfies $\bar{h}(x) = f(x)$ and that divides x^n-1 in $\mathbb{Z}_{p^s}[x]$, which is called the Hensel lift of $f(x)$ to $\mathbb{Z}_{p^s}$. Moreover, if a polynomial of degree m is the Hensel lift of a monic primitive polynomial over $\mathbb{Z}_p$, then it always has a root of order p^m-1 [Wan03]. Let $h(x)$ be such a polynomial, with $m = t_1-1$. Let $\alpha \in \mathcal{R}$ be a root of $h(x)$ of order $\ell = p^{t_1-1}-1$. Then, the set $T = \{0,1,\alpha,\alpha^2,\dots,\alpha^{\ell-1}\}$ is called the *Teichmüller set*.

We can always represent an element $y \in \mathcal{R}$ in the following form:

$$y = a_1 + pa_2 + p^2a_3 + \cdots + p^{s-1}a_s,$$

where $a_i \in T$, for $i \in \{1,\dots,s\}$, which is called the *p-adic representation* of

y. Consider T as an ordered set. Then, we consider the following ordering of the elements of $\mathcal{R} = \{y_1, \ldots, y_{p^{s(t_1-1)}}\}$: $a_1 + pa_2 + \cdots + p^{s-1}a_s < b_1 + pb_2 + \cdots + p^{s-1}b_s$ if $a_j < b_j$ for the last j where a_j and b_j differ. We can also represent an element $y \in \mathcal{R}$ as a linear combination of some powers of α:

$$y = b_0 + b_1\alpha + b_2\alpha^2 + \cdots + b_{t_1-2}\alpha^{t_1-2},$$

where $b_j \in \mathbb{Z}_{p^s}$, for $j \in \{0, \ldots, t_1 - 2\}$. This is called the *additive representation* of y and it can be identified with the vector $(b_0, b_1, \ldots, b_{t_1-2}) \in \mathbb{Z}_{p^s}^{t_1-1}$.

Using the ordering given by the p-adic representation, we construct the set $\{\mathcal{N}_0^*, \ldots, \mathcal{N}_r^*\}$ of matrices of size $t_1 \times t_1$, where each one has the following form:

$$\mathcal{N}_i^* = \begin{pmatrix} 1 & y_{t_1 i+1} \\ 1 & y_{t_1 i+2} \\ \vdots & \vdots \\ 1 & y_{t_1(i+1)} \end{pmatrix},$$

with the elements y_j, for $j \in \{1, \ldots, p^{s(t_1-1)}\}$, given as vectors of $t_1 - 1$ components over $\mathbb{Z}_{p^s}$ by using the corresponding additive representation. Note that no two matrices have a row in common, and there are $\lfloor |\mathcal{R}|/t_1 \rfloor = f_p^{t_1,0,\ldots,0} + 1$ such matrices, where $f_p^{t_1,0,\ldots,0} = \lfloor (p^{st_1} - t_1)/t_1 \rfloor$ by Corollary 38, so $r \leq f_p^{t_1,0,\ldots,0}$.

Let $n_{i,j}^*$ be the jth row of the matrix $\mathcal{N}_i^*$, for any $i \in \{0, \ldots, r\}$ and $j \in \{1, \ldots, t_1\}$. In the context of the $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,0,\ldots,0}$, we define $\mathcal{N}_i$ as the matrix that has $n_{i,1}^*$ as the first row and $n_{i,j}^* - n_{i,1}^*$ as the jth row, for $j \in \{2, \ldots, t_1\}$. Note that this is consistent with Definition 35. Indeed, in the proof of Theorem 40, we see that $\mathcal{N}_0, \ldots, \mathcal{N}_r \subseteq \mathrm{PAut}(\mathcal{H}^{t_1,0,\ldots,0})$.

Lemma 39. *Let $K = \mathbb{Z}_p[x]/(f(x))$, where $f(x) \in \mathbb{Z}_p[x]$ is a primitive polynomial of degree m. Let $\alpha \in K$ be a root of $f(x)$. Then, $\alpha - 1, \alpha^2 - 1, \ldots, \alpha^m - 1$ are linearly independent vectors over $\mathbb{Z}_p$.*

Proof. Let $f(x) = x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0$, where $a_j \in \mathbb{Z}_p$ for all $j \in \{0, \ldots, m-1\}$. Since α is a root of $f(x)$, then $\alpha^m - 1 = -\sum_{j=0}^{m-1} a_j\alpha^j - 1$. Using the additive representation of the elements of K, we obtain the

following vectors over $\mathbb{Z}_p^m$: $\alpha^i - 1 = e_{i+1} - e_1$ for any $i \in \{1, \ldots, m-1\}$, and $\alpha^m - 1 = -\sum_{j=0}^{m-1} a_j e_{j+1} - e_1$. Consider the following $m \times m$ matrix over $\mathbb{Z}_p$ by taking these vectors as rows:

$$\begin{pmatrix} -\mathbf{1} & \mathrm{Id}_{m-1} \\ -a_0 - 1 & -a \end{pmatrix},$$

where $a = (a_1, \ldots, a_{m-1}) \in \mathbb{Z}_p^{m-1}$. This matrix has the following determinant: $(-1)^m(\sum_{j=0}^{m-1} a_j + 1)$. Since $f(x)$ is irreducible, then $f(1) = 1 + \sum_{j=0}^{m-1} a_j \neq 0$. Therefore, the determinant is non-zero and the vectors are linearly independent over $\mathbb{Z}_p$. □

Theorem 40. *Let $\mathcal{P}_r = \{\mathcal{N}_0^{-1}, \ldots, \mathcal{N}_r^{-1}\}$. Then, $\Phi(\mathcal{P}_r)$ is an r-PD-set of size $r+1$ for the $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,0,\ldots,0}$ with information set $\Phi(\mathcal{I}_{t_1,0,\ldots,0})$, for all $t_1 \geq 2$ and $2 \leq r \leq f_p^{t_1,0,\ldots,0}$.*

Proof. By construction, the matrices $\mathcal{N}_0^*, \ldots, \mathcal{N}_r^*$ do not share a row in common. Thus, if we prove that all matrices $\mathcal{N}_0^{-1}, \ldots, \mathcal{N}_r^{-1}$ are in $\mathrm{PAut}(\mathcal{H}^{t_1,0,\ldots,0})$, then $\Phi(\mathcal{P}_r)$, where $\mathcal{P}_r = \{\mathcal{N}_0^{-1}, \ldots, \mathcal{N}_r^{-1}\}$, would be an r-PD-set of size $r+1$ for $H^{t_1,0,\ldots,0}$, by Theorem 36. Since $\mathrm{PAut}(\mathcal{H}^{t_1,0,\ldots,0})$ is a group, it is enough to prove that $\mathcal{N}_0, \ldots, \mathcal{N}_r$ are in this group. Note that these matrices are in the form

$$\mathcal{N}_i = \begin{pmatrix} 1 & y_{t_1 i+1} \\ 0 & y_{t_1 i+2} - y_{t_1 i+1} \\ \vdots & \vdots \\ 0 & y_{t_1(i+1)} - y_{t_1 i+1} \end{pmatrix}, \tag{4.8}$$

for any $i \in \{0, \ldots, r\}$. As shown in (4.2), the elements in $\mathrm{PAut}(\mathcal{H}^{t_1,0,\ldots,0})$ have the form

$$\begin{pmatrix} 1 & a_1 \\ 0 & A_{1,1} \end{pmatrix}, \tag{4.9}$$

where $a_1 \in \mathbb{Z}_{p^s}^{t_1-1}$ and $A_{1,1} \in \mathrm{GL}(t_1 - 1, \mathbb{Z}_{p^s})$. By using the additive representation, $y_{y_1 i+1} \in \mathbb{Z}_{p^s}^{t_1-1}$, for any $i \in \{0, \ldots, r\}$. Then, we need to prove that the vectors $y_{t_1 i+2} - y_{t_1 i+1}, \ldots, y_{t_1(i+1)} - y_{t_1 i+1}$ are linearly independent over $\mathbb{Z}_{p^s}$, for any $i \in \{0, \ldots, r\}$. Taking into account that $\alpha^\ell = 1$ and $t_1 \leq p^{t_1-1}$

for $t_1 \geq 2$, the set of vectors $\{y_{t_1 i+2} - y_{t_1 i+1}, \ldots, y_{t_1(i+1)} - y_{t_1 i+1}\}$ is equal to one of the following three sets:

$$\begin{aligned}
L_1 &= \{1, \ldots, \alpha^{t_1-2}\},\\
L_2 &= \{\alpha^{k+1} - \alpha^k, \ldots, \alpha^{k+t_1-1} - \alpha^k\},\\
L_3 &= \{\alpha^{k+1} - \alpha^k, \ldots, \alpha^{\ell-1} - \alpha^k,\\
&\qquad - \alpha^k + pb,\\
&\qquad \alpha^{\ell} - \alpha^k + pb, \ldots, \alpha^{k+t_1-2} - \alpha^k + pb\},
\end{aligned}$$

for some $k \in \{0, \ldots, \ell - 1\}$ and some $b \in \mathcal{R}$. Clearly, L_1 is a set of linearly independent vectors over $\mathbb{Z}_{p^s}$.

For the second set L_2, suppose that $\sum_{i=1}^{t_1-1} \lambda_i(\alpha^{k+i} - \alpha^k) = 0$ for certain $\lambda_i \in \mathbb{Z}_{p^s}$, with some of them being non-zero. Note that, since α is a unit in $\mathcal{R}$, then $\sum_i \lambda_i(\alpha^i - 1) = 0$. Let m be the smallest integer in $\{0, \ldots, s-1\}$ for which there exists an $i \in \{1, \ldots, t_1 - 1\}$ such that $\lambda_i \in p^m\mathbb{Z}_{p^s}$ and $\lambda_i \notin p^{m+1}\mathbb{Z}_{p^s}$. For example, if all $\lambda_i \in p\mathbb{Z}_{p^s}$ and there is a certain $\lambda_i \notin p^2\mathbb{Z}_{p^s}$, then $m = 1$. Therefore, we can define $\lambda_i = p^m\lambda'_i$ for all i, and we obtain $p^m \sum_i \lambda'_i(\alpha^i - 1) = 0$, hence $\sum_i \lambda'_i(\alpha^i - 1) = p^{s-m}\lambda$ for a certain $\lambda \in \mathcal{R}$. Thus, by taking modulo p, we obtain $\sum_i \bar{\lambda}'_i(\bar{\alpha}^i - 1) = 0$ over $\mathbb{Z}_p$, with at least one $\bar{\lambda}'_i \neq 0$. Clearly, $\bar{\alpha}$ is a unit in $\mathbb{Z}_p[x]/(\bar{h}(x))$. Therefore, by applying Lemma 39 on the vectors $\bar{\alpha}^i - 1$ for $i \in \{1, \ldots, t_1 - 1\}$, we obtain a contradiction.

For the third set L_3, we follow a similar argument. Suppose that

$$-\lambda_{t_1-1}(\alpha^k - pb) + \sum_{i=1}^{\ell-k-1} \lambda_i(\alpha^{k+i} - \alpha^k) + \sum_{i=\ell-k}^{t_1-2} \lambda_i(\alpha^{k+i} - \alpha^k + pb) = 0$$

for certain $\lambda_i \in \mathbb{Z}_{p^s}$, with some of them being non-zero. With the same definition of m as in the previous case, we obtain $-p^m\lambda'_{t_1-1}(\alpha^k - pb) + p^m \sum_{i=1}^{\ell-k-1} \lambda'_i(\alpha^{k+i} - \alpha^k) + p^m \sum_{i=\ell-k}^{t_1-2} \lambda'_i(\alpha^{k+i} - \alpha^k + pb) = 0$, where $\lambda_i = p^m\lambda'_i$. Thus, $-\lambda'_{t_1-1}(\alpha^k - pb) + \sum_{i=1}^{\ell-k-1} \lambda'_i(\alpha^{k+i} - \alpha^k) + \sum_{i=\ell-k}^{t_1-2} \lambda'_i(\alpha^{k+i} - \alpha^k + pb) = p^{s-m}\lambda$ for some $\lambda \in \mathcal{R}$. Taking modulo p, $-\bar{\lambda}'_{t_1-1}\bar{\alpha}^k + \sum_i \bar{\lambda}'_i(\bar{\alpha}^{k+i} - \bar{\alpha}^k) = 0$ over $\mathbb{Z}_p$, with at least one $\bar{\lambda}'_i \neq 0$. Since $\bar{\alpha}$ is a unit, we obtain $-\bar{\lambda}'_{t_1-1} + \sum_i \bar{\lambda}'_i(\bar{\alpha}^i - 1) = 0$. We obtain a contradiction since the vectors

$-1, \bar{\alpha}-1, \ldots, \bar{\alpha}^{t_1-2}-1$ are linearly independent over $\mathbb{Z}_p$. □

Example 41. *Let $\mathcal{H}^{3,0,0}$ be the $\mathbb{Z}_{27}$-additive GH code of type $(3^6; 3, 0, 0)$. Let $\mathcal{R} = \mathrm{GR}(27^2)$ be the Galois ring over $\mathbb{Z}_{27}$, isomorphic to $\mathbb{Z}_{27}[x]/(h(x))$, where $h(x) = x^2 + 22x + 26$. This polynomial can be obtained as the Hensel lift of $f(x) = \bar{h}(x) = x^2 + x + 2$ over $\mathbb{Z}_3$. Note that $h(x)$ is a monic basic primitive polynomial dividing $x^8 - 1$ in $\mathbb{Z}_{27}[x]$. Let α be a root of $h(x)$ of order 8. Then, $T = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^7\}$ and we can order the elements of $\mathcal{R}$ as follows:*

$$\begin{aligned}\mathcal{R} = \{&0+3\cdot 0+9\cdot 0, 1+3\cdot 0+9\cdot 0, \alpha+3\cdot 0+9\cdot 0, \ldots, \alpha^7+3\cdot 0+9\cdot 0,\\ &0+3\cdot 1+9\cdot 0, 1+3\cdot 1+9\cdot 0, \alpha+3\cdot 1+9\cdot 0, \ldots, \alpha^7+3\cdot 1+9\cdot 0,\\ &\ldots\\ &0+3\cdot\alpha^7+9\cdot\alpha^7, 1+3\cdot\alpha^7+9\cdot\alpha^7, \alpha+3\cdot\alpha^7+9\cdot\alpha^7, \ldots,\\ &\quad \alpha^7+3\cdot\alpha^7+9\cdot\alpha^7\}\\ = \{&0, 1, \alpha, \ldots, 22+\alpha,\\ &3, 4, 3+\alpha, \ldots, 25+\alpha,\\ &\ldots\\ &21+12\alpha, 22+12\alpha, 21+13\alpha, \ldots, 16+13\alpha\}.\end{aligned}$$

By Theorem 40, we can find r-PD-sets of size $r+1$ for all $2 \le r \le f_3^{3,0,0} = 242$, by using the elements of $\mathcal{R}$. Indeed, we can construct up to 243 matrices taking all the elements of $\mathcal{R}$ in groups of 3, reaching the upper bound. Here, we show a smaller example by constructing an 11-PD-set formed by 12 matrices.

Consider the following 12 matrices, constructed by dividing the first 36 ordered elements of $\mathcal{R}$ in groups of 3:

$$\mathcal{N}_0^* = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad \mathcal{N}_1^* = \begin{pmatrix} 1 & 1 & 5 \\ 1 & 5 & 26 \\ 1 & 26 & 0 \end{pmatrix}, \quad \mathcal{N}_2^* = \begin{pmatrix} 1 & 0 & 26 \\ 1 & 26 & 22 \\ 1 & 22 & 1 \end{pmatrix},$$

$$\mathcal{N}_3^* = \begin{pmatrix} 1 & 3 & 0 \\ 1 & 4 & 0 \\ 1 & 3 & 1 \end{pmatrix}, \quad \mathcal{N}_4^* = \begin{pmatrix} 1 & 4 & 5 \\ 1 & 8 & 26 \\ 1 & 2 & 0 \end{pmatrix}, \quad \mathcal{N}_5^* = \begin{pmatrix} 1 & 3 & 26 \\ 1 & 2 & 22 \\ 1 & 25 & 1 \end{pmatrix},$$

$$\mathcal{N}_6^* = \begin{pmatrix} 1 & 0 & 3 \\ 1 & 1 & 3 \\ 1 & 0 & 4 \end{pmatrix}, \quad \mathcal{N}_7^* = \begin{pmatrix} 1 & 1 & 8 \\ 1 & 5 & 2 \\ 1 & 26 & 3 \end{pmatrix}, \quad \mathcal{N}_8^* = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 26 & 25 \\ 1 & 22 & 4 \end{pmatrix},$$

$$\mathcal{N}_9^* = \begin{pmatrix} 1 & 3 & 15 \\ 1 & 4 & 15 \\ 1 & 3 & 16 \end{pmatrix}, \quad \mathcal{N}_{10}^* = \begin{pmatrix} 1 & 4 & 20 \\ 1 & 8 & 14 \\ 1 & 2 & 15 \end{pmatrix}, \quad \mathcal{N}_{11}^* = \begin{pmatrix} 1 & 3 & 14 \\ 1 & 2 & 10 \\ 1 & 25 & 16 \end{pmatrix}.$$

Note that there are no repeated rows in the whole set of matrices. Let $\mathcal{P}_{11} = \{\mathcal{N}_0^{-1}, \ldots, \mathcal{N}_{11}^{-1}\}$, *where*

$$\mathcal{N}_0^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{N}_1^{-1} = \begin{pmatrix} 1 & 1 & 16 \\ 0 & 1 & 15 \\ 0 & 5 & 10 \end{pmatrix}, \quad \mathcal{N}_2^{-1} = \begin{pmatrix} 1 & 1 & 16 \\ 0 & 22 & 17 \\ 0 & 1 & 16 \end{pmatrix},$$

$$\mathcal{N}_3^{-1} = \begin{pmatrix} 1 & 24 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{N}_4^{-1} = \begin{pmatrix} 1 & 25 & 25 \\ 0 & 1 & 15 \\ 0 & 5 & 10 \end{pmatrix}, \quad \mathcal{N}_5^{-1} = \begin{pmatrix} 1 & 16 & 19 \\ 0 & 22 & 17 \\ 0 & 1 & 16 \end{pmatrix},$$

$$\mathcal{N}_6^{-1} = \begin{pmatrix} 1 & 0 & 24 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{N}_7^{-1} = \begin{pmatrix} 1 & 13 & 13 \\ 0 & 1 & 15 \\ 0 & 5 & 10 \end{pmatrix}, \quad \mathcal{N}_8^{-1} = \begin{pmatrix} 1 & 25 & 22 \\ 0 & 22 & 17 \\ 0 & 1 & 16 \end{pmatrix},$$

$$\mathcal{N}_9^{-1} = \begin{pmatrix} 1 & 24 & 12 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{N}_{10}^{-1} = \begin{pmatrix} 1 & 4 & 10 \\ 0 & 1 & 15 \\ 0 & 5 & 10 \end{pmatrix}, \quad \mathcal{N}_{11}^{-1} = \begin{pmatrix} 1 & 1 & 22 \\ 0 & 22 & 17 \\ 0 & 1 & 16 \end{pmatrix}.$$

The matrices of $\mathcal{P}_{11}$ *can also be represented as permutations of coordinate positions as shown in (4.7). Let* $\tau_i \in \mathrm{Sym}(729)$ *be the one corresponding to* $\mathcal{N}_i^{-1}$, $i \in \{0, \ldots, 11\}$. *Recall that* $\Phi(\mathcal{N}_i^{-1}) = \Phi(\tau_i)$ *as defined in (4.6). Then, by Theorem 40,* $\Phi(\mathcal{P}_{11}) = \{\Phi(\mathcal{N}_i^{-1}) : i \in \{0, \ldots, 11\}\} \subseteq \mathrm{Sym}(6561)$ *is an* 11*-PD-set of size* 12 *for the* $\mathbb{Z}_{27}$*-linear GH code* $H^{3,0,0} = \Phi(\mathcal{H}^{3,0,0})$ *with information set* $\Phi(\mathcal{I}_{3,0,0}) = \{1, 2, 4, 10, 11, 13, 244, 245, 247\}$, *given in Example 31.*

Remark 42. *By Corollary 38,* $f_p^{t_1,0,\ldots,0}$ *is the maximum number of errors that can be corrected using r-PD-sets of size* $r + 1$ *of the form* $\Phi(\mathcal{P}_r)$, *where*

$\mathcal{P}_r \subseteq \mathrm{PAut}(\mathcal{H}^{t_1,0,\ldots,0})$. *However, higher values of* r *could be achieved by considering elements in* $\mathrm{PAut}(H^{t_1,0,\ldots,0})$ *that are not the* Φ *image of elements in* $\mathrm{PAut}(\mathcal{H}^{t_1,0,\ldots,0})$.

Following a similar reasoning to the one used in Theorem 40, we can also obtain r-PD-sets of size $r+1$ for the $\mathbb{Z}_{p^s}$-linear GH codes $H_i = H^{1,t_2,\ldots,t_s}$, $i \in \{2,\ldots,s\}$, of type $(n; 1, t_2, \ldots, t_s)$, where $t_j = 0$ for all $j \neq i$, $t_i \geq 1$, and $r \leq f_p^{1,t_2,\ldots,t_s}$. Let $\mathcal{R}_i = \mathrm{GR}(p^{(s-i+1)t_i})$ be the Galois extension of dimension t_i over $\mathbb{Z}_{p^{s-i+1}}$, isomorphic to $\mathbb{Z}_{p^{s-i+1}}[x]/(h(x))$, with $h(x)$ being a monic basic primitive polynomial of degree t_i dividing $x^{p^{t_i}-1}-1$ in $\mathbb{Z}_{p^{s-i+1}}[x]$. Let $\alpha \in \mathcal{R}_i$ be a root of $h(x)$ of order $\ell = p^{t_i}-1$ and $T = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{\ell-1}\}$. The p-adic representation of an element $y \in \mathcal{R}_i$ is $y = a_1 + pa_2 + p^2a_3 + \cdots + p^{s-i}a_{s-i+1}$, where $a_k \in T$ for $k \in \{1,\ldots,s-i+1\}$. Using this representation, we define the ordered set $\{y_1, \ldots, y_{p^{(s-i+1)t_i}}\}$ with all the elements in $\mathcal{R}_i$. Consider the set of matrices $\{\mathcal{M}_0^*, \ldots, \mathcal{M}_r^*\}$ of size $(t_i+1)\times(t_i+1)$ over $\mathbb{Z}_{p^s}$, where

$$\mathcal{M}_j^* = \begin{pmatrix} 1 & p^{i-1}y_{(t_i+1)j+1} \\ 1 & p^{i-1}y_{(t_i+1)j+2} \\ \vdots & \vdots \\ 1 & p^{i-1}y_{(t_i+1)(j+1)} \end{pmatrix},$$

for $j \in \{0,\ldots,r\}$. Note that

$$r \leq f_p^{1,t_2,\ldots,t_s} = \lfloor \frac{p^{(s-i+1)t_i}-1-t_i}{1+t_i} \rfloor$$

by Corollary 38. The elements $y \in \mathcal{R}_i$ are given as vectors over $\mathbb{Z}_{p^{s-i+1}}$ by using the additive representation. Then, we consider the inclusion of vectors y over $\mathbb{Z}_{p^{s-i+1}}$ to vectors over $\mathbb{Z}_{p^s}$ as $p^{i-1}y$.

Let $(1, p^{i-1}m_{j,k}^*)$ be the kth row of the matrix $\mathcal{M}_j^*$, for any $j \in \{0,\ldots,r\}$. In the context of the $\mathbb{Z}_{p^s}$-linear GH code H_i, we define $\mathcal{M}_j$ as the matrix that has $(1, p^{i-1}m_{j,1}^*)$ as the first row and $(0, m_{j,k}^* - m_{j,1}^*)$ as the kth row, for $k \in \{2,\ldots,t_i+1\}$. Note that this is consistent with Definition 35. Indeed, in the proof of Corollary 43, we see that $\mathcal{M}_0,\ldots,\mathcal{M}_r \subseteq \mathrm{PAut}(\mathcal{H}_i)$, where $\mathcal{H}_i$ is the $\mathbb{Z}_{p^s}$-additive code such that $\Phi(\mathcal{H}_i) = H_i$.

Corollary 43. *Let* $\mathcal{P}_r = \{\mathcal{M}_0^{-1}, \ldots, \mathcal{M}_r^{-1}\}$. *Then,* $\Phi(\mathcal{P}_r)$ *is an* r*-PD-set of size* $r+1$ *for the* $\mathbb{Z}_{p^s}$*-linear GH code* $H_i = H^{1,t_2,\ldots,t_s}$, *where* $t_i \geq 1$, $i \in \{2,\ldots,s\}$, *and* $t_j = 0$ *for all* $j \in \{2,\ldots,s\}$ *such that* $j \neq i$, *with information set* $\Phi(\mathcal{I}_{1,t_2,\ldots,t_s})$, *for all* $2 \leq r \leq f_p^{1,t_2,\ldots,t_s}$.

Proof. Since $\mathbb{Z}_{p^{s-i+1}}$ is isomorphic to $p^{i-1}\mathbb{Z}_{p^s}$, the matrices $\mathcal{M}_0^*, \ldots, \mathcal{M}_r^*$ do not share a row in common. Then, the matrix $\mathcal{M}_j$ is in the form

$$\mathcal{M}_j = \begin{pmatrix} 1 & p^{i-1}y_{(t_i+1)j+1} \\ 0 & y_{(t_i+1)j+2} - y_{(t_i+1)j+1} \\ \vdots & \vdots \\ 0 & y_{(t_i+1)(j+1)} - y_{(t_i+1)j+1} \end{pmatrix}, \tag{4.10}$$

for any $j \in \{0,\ldots,r\}$. As shown in (4.2), the elements in $\mathrm{PAut}(\mathcal{H}^{1,0,\ldots,0,t_i,0,\ldots,0})$ have the form

$$\begin{pmatrix} 1 & p^{i-1}a_i \\ 0 & \zeta_{s-i+1}(A_{i,i}) \end{pmatrix}, \tag{4.11}$$

where $a_i \in \mathbb{Z}_{p^s}^{t_i}$ and $A_{i,i} \in \mathrm{GL}(t_i, \mathbb{Z}_{p^s})$. Note that $\zeta_{s-i+1}(A_{i,i})$ can be seen as an element of $\mathrm{GL}(t_i, \mathbb{Z}_{p^{s-i+1}})$ and, in fact, it can be any element in $\mathrm{GL}(t_i, \mathbb{Z}_{p^{s-i+1}})$.

Following the same argument as in the proof of Theorem 40, over the Galois ring $\mathcal{R}_i = \mathrm{GR}(p^{(s-i+1)t_i})$ instead of $\mathrm{GR}(p^{s(t_1-1)})$, it can be proven that the vectors $y_{(t_i+1)j+2} - y_{(t_i+1)j+1}, \ldots, y_{(t_i+1)(j+1)} - y_{(t_i+1)j+1}$ are linearly independent over $\mathbb{Z}_{p^{s-i+1}}$. □

Example 44. *Let* $\mathcal{H}^{1,3,0}$ *be the* $\mathbb{Z}_8$*-additive Hadamard code of type* $(2^6; 1, 3, 0)$. *Let* $\mathcal{R}_2 = \mathrm{GR}(4^3)$ *be the Galois ring over* $\mathbb{Z}_4$, *isomorphic to* $\mathbb{Z}_4[x]/(h(x))$, *where* $h(x) = x^3 + 2x^2 + x + 3$. *This polynomial can be obtained as the Hensel lift of* $f(x) = \bar{h}(x) = x^3 + x + 1$ *over* $\mathbb{Z}_2$. *Note that* $h(x)$ *is a monic basic primitive polynomial dividing* $x^7 - 1$ *in* $\mathbb{Z}_4[x]$. *Let* α *be a root of* $h(x)$ *of order* 7. *Then,* $T = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^6\}$ *and we can order the* 64 *elements of* $\mathcal{R}_2$ *as follows:*

$$\begin{aligned} \mathcal{R}_2 = \{&0 + 2\cdot 0, 1 + 2\cdot 0, \alpha + 2\cdot 0, \ldots, \alpha^6 + 2\cdot 0, \\ &0 + 2\cdot 1, 1 + 2\cdot 1, \alpha + 2\cdot 1, \ldots, \alpha^6 + 2\cdot 1, \end{aligned}$$

$$
\begin{aligned}
&\ldots,\\
&0+2\cdot\alpha^6, 1+2\cdot\alpha^6, \alpha+2\cdot\alpha^6, \ldots, \alpha^6+2\cdot\alpha^6\}\\
=\{&0,1,\alpha,\ldots,1+2\alpha+\alpha^2,\\
&2,3,2+\alpha,\ldots,3+2\alpha+\alpha^2,\\
&\ldots\\
&2+2\alpha^2, 3+2\alpha^2, 2+\alpha+2\alpha^2, \ldots, 3+2\alpha+3\alpha^2\}.
\end{aligned}
$$

By Corollary 43, we can find r-PD-sets of size $r+1$ for all $2 \leq r \leq f_2^{1,3,0} = 15$, by using the elements of $\mathcal{R}_2$. Indeed, we can construct up to 16 matrices taking all the elements of $\mathcal{R}_2$, multiplied by 2, in groups of 4 in order to reach the upper bound. Here, we just show a smaller example by constructing an 8-PD-set formed by 9 matrices.

Consider the following 9 matrices over $\mathbb{Z}_8$, constructed by taking the first 36 ordered elements of $\mathcal{R}_2$ in groups of 4 and multiplying them by 2 as elements of $\mathbb{Z}_8^3$:

$$
\mathcal{M}_0^* = \begin{pmatrix} 1&0&0&0\\ 1&2&0&0\\ 1&0&2&0\\ 1&0&0&2 \end{pmatrix}, \quad
\mathcal{M}_1^* = \begin{pmatrix} 1&2&6&4\\ 1&4&6&6\\ 1&6&6&2\\ 1&2&4&2 \end{pmatrix}, \quad
\mathcal{M}_2^* = \begin{pmatrix} 1&4&0&0\\ 1&6&0&0\\ 1&4&2&0\\ 1&4&0&2 \end{pmatrix},
$$

$$
\mathcal{M}_3^* = \begin{pmatrix} 1&6&6&4\\ 1&0&6&6\\ 1&2&6&2\\ 1&6&4&2 \end{pmatrix}, \quad
\mathcal{M}_4^* = \begin{pmatrix} 1&0&4&0\\ 1&2&4&0\\ 1&0&6&0\\ 1&0&4&2 \end{pmatrix}, \quad
\mathcal{M}_5^* = \begin{pmatrix} 1&2&2&4\\ 1&4&2&6\\ 1&6&2&2\\ 1&2&0&2 \end{pmatrix},
$$

$$
\mathcal{M}_6^* = \begin{pmatrix} 1&0&0&4\\ 1&2&0&4\\ 1&0&2&4\\ 1&0&0&6 \end{pmatrix}, \quad
\mathcal{M}_7^* = \begin{pmatrix} 1&2&6&0\\ 1&4&6&2\\ 1&6&6&6\\ 1&2&4&6 \end{pmatrix}, \quad
\mathcal{M}_8^* = \begin{pmatrix} 1&4&4&0\\ 1&6&4&0\\ 1&4&6&0\\ 1&4&4&2 \end{pmatrix}.
$$

Note that there are no repeated rows in the whole set of matrices. Let $\mathcal{P}_8 =$

$\{\mathcal{M}_0^{-1}, \ldots, \mathcal{M}_8^{-1}\}$, *where*

$$\mathcal{M}_0^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathcal{M}_1^{-1} = \begin{pmatrix} 1 & 6 & 4 & 6 \\ 0 & 3 & 3 & 0 \\ 0 & 2 & 3 & 3 \\ 0 & 2 & 1 & 0 \end{pmatrix}, \ \mathcal{M}_2^{-1} = \begin{pmatrix} 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathcal{M}_3^{-1} = \begin{pmatrix} 1 & 2 & 0 & 6 \\ 0 & 3 & 3 & 0 \\ 0 & 2 & 3 & 3 \\ 0 & 2 & 1 & 0 \end{pmatrix}, \ \mathcal{M}_4^{-1} = \begin{pmatrix} 1 & 0 & 4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathcal{M}_5^{-1} = \begin{pmatrix} 1 & 6 & 0 & 2 \\ 0 & 3 & 3 & 0 \\ 0 & 2 & 3 & 3 \\ 0 & 2 & 1 & 0 \end{pmatrix},$$

$$\mathcal{M}_6^{-1} = \begin{pmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathcal{M}_7^{-1} = \begin{pmatrix} 1 & 6 & 0 & 6 \\ 0 & 3 & 3 & 0 \\ 0 & 2 & 3 & 3 \\ 0 & 2 & 1 & 0 \end{pmatrix}, \ \mathcal{M}_8^{-1} = \begin{pmatrix} 1 & 4 & 4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The matrices of $\mathcal{P}_8$ *can also be represented as permutations of coordinate positions as shown in (4.7). Let* $\tau_i \in \mathrm{Sym}(64)$ *be the one corresponding to* $\mathcal{M}_i^{-1}$, $i \in \{0, \ldots, 8\}$. *For example, matrix* $\mathcal{M}_1^{-1}$ *is equivalent to the permutation*

$$\begin{aligned} \tau_1 = &(1, 60, 19, 56, 37, 46)(2, 55, 42, 23, 34, 63)(3, 50, 49, 54, 47, 16) \\ &(4, 61, 12, 21, 44, 29)(5, 38, 41, 28, 27, 32)(6, 33, 52, 59, 24, 45) \\ &(7, 48, 11, 26, 17, 62)(8, 43, 18, 57, 30, 15)(9, 20, 51, 64, 13, 14) \\ &(10, 31)(22, 39, 40, 35, 58, 25)(36, 53). \end{aligned}$$

Recall that $\Phi(\mathcal{M}_j^{-1}) = \Phi(\tau_j)$ *as defined in (4.6). Then, by Corollary 43,* $\Phi(\mathcal{P}_8) = \{\Phi(\mathcal{M}_j^{-1}) : j \in \{0, \ldots, 8\}\} \subseteq \mathrm{Sym}(256)$ *is an* 8*-PD-set of size* 9 *for the* $\mathbb{Z}_8$*-linear Hadamard code* $H^{1,3,0} = \Phi(\mathcal{H}^{1,3,0})$ *with information set* $\Phi(\mathcal{I}_{1,3,0}) = \{1, 2, 3, 5, 7, 17, 19, 65, 67\}$.

4.4 Recursive constructions of r-PD-sets

In this section, given an r-PD-set of size ℓ for a $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,\ldots,t_s}$, we show that one can easily obtain an r-PD-set of size ℓ for the $\mathbb{Z}_{p^s}$-linear

GH code $H^{t_1+i_1,\dots,t_s+i_s}$, for all $i_1,\dots,i_s \geq 0$. In particular, this is useful to obtain r-PD-sets for any code $H^{t_1,\dots,t_s}$, including those of type different to $(n;t_1,0,\dots,0)$ and $(n;1,0,\dots,0,t_i,0,\dots,0)$, $i \in \{2,\dots,s\}$, which have been already considered in Section 4.3.

We present two different constructions that produce a similar result. One uses the matrix representation of the elements in $\mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$ and the other one uses the permutation representation. These constructions are a generalization of the ones given in [BV18] for $\mathbb{Z}_4$-linear Hadamard codes.

4.4.1 Matrix representation

In this first construction, we consider the elements of $\mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$ as matrices in the subgroup $\pi(\mathcal{L})$ of $\mathrm{GL}(t_1+\cdots+t_s,\mathbb{Z}_{p^s})$, described in Section 4.1. Consider a matrix $\mathcal{M} \in \mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$ in the form given in (4.2) and s positive integers $\kappa_1,\dots,\kappa_s$. Then, we define the matrix $\mathcal{M}(\kappa_1,\dots,\kappa_s)$ as

$$\begin{pmatrix} 1 & a'_1 & pa'_2 & \cdots & p^{s-2}a'_{s-1} & p^{s-1}a'_s \\ \mathbf{0} & A'_{1,1} & pA'_{1,2} & \cdots & p^{s-2}A'_{1,s-1} & p^{s-1}A'_{1,s} \\ \mathbf{0} & \zeta_{s-1}(A'_{2,1}) & \zeta_{s-1}(A'_{2,2}) & \cdots & \zeta_{s-1}(p^{s-3}A'_{2,s-1}) & \zeta_{s-1}(p^{s-2}A'_{2,s}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \zeta_2(A'_{s-1,1}) & \zeta_2(A'_{s-1,2}) & \cdots & \zeta_2(A'_{s-1,s-1}) & \zeta_2(pA'_{s-1,s}) \\ \mathbf{0} & \zeta_1(A'_{s,1}) & \zeta_1(A'_{s,2}) & \cdots & \zeta_1(A'_{s,s-1}) & \zeta_1(A'_{s,s}) \end{pmatrix},$$

where $a'_1 = (a_1,\mathbf{0}) \in \mathbb{Z}_{p^s}^{t_1-1+\kappa_1}$, $a'_j = (a_j,\mathbf{0}) \in \mathbb{Z}_{p^s}^{t_j+\kappa_j}$ for $j \in \{2,\dots,s\}$, $A'_{1,1} = \begin{pmatrix} A_{1,1} & \mathbf{0} \\ \mathbf{0} & \mathrm{Id}_{\kappa_1} \end{pmatrix} \in \mathrm{GL}(t_1-1+\kappa_1)$, $A'_{i,i} = \begin{pmatrix} A_{i,i} & \mathbf{0} \\ \mathbf{0} & \mathrm{Id}_{\kappa_i} \end{pmatrix} \in \mathrm{GL}(t_i+\kappa_i,\mathbb{Z}_{p^s})$ for $i \in \{2,\dots,s\}$, and $A'_{i,j} = \begin{pmatrix} A_{i,j} & \mathbf{0} \end{pmatrix}$, $A'_{j,i} = \begin{pmatrix} A_{j,i} \\ \mathbf{0} \end{pmatrix}$ are matrices over $\mathbb{Z}_{p^s}$ for $i,j \in \{1,\dots,s\}$ with $i<j$, respectively. Note that $\mathcal{M}(\kappa_1,\dots,\kappa_s) \in \mathrm{GL}(t_1+\cdots+t_s+\kappa_1+\cdots+\kappa_s,\mathbb{Z}_{p^s})$.

Proposition 45. *Let $\mathcal{P}_r = \{\mathcal{M}_0,\dots,\mathcal{M}_r\} \subseteq \mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$ such that $\Phi(\mathcal{P}_r)$ is an r-PD-set of size $r+1$ for $H^{t_1,\dots,t_s}$ with information set $\Phi(\mathcal{I}_{t_1,\dots,t_s})$. Then, $\mathcal{Q}_r = \{(\mathcal{M}_i^{-1}(\kappa_1,\dots,\kappa_s))^{-1} : i \in \{0,\dots,r\}\} \subseteq \mathrm{PAut}(\mathcal{H}^{t_1+\kappa_1,\dots,t_s+\kappa_s})$ and*

$\Phi(\mathcal{Q}_r)$ is an r-PD-set of size $r+1$ for $H^{t_1+\kappa_1,\ldots,t_s+\kappa_s}$ with information set $\Phi(\mathcal{I}_{t_1+\kappa_1,\ldots,t_s+\kappa_s})$, for any $\kappa_1,\ldots,\kappa_s \geq 0$.

Proof. Since $\mathcal{M}_i^{-1} \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, $\mathcal{M}_i^{-1}(\kappa_1,\ldots,\kappa_s) \in \mathrm{PAut}(\mathcal{H}^{t_1+\kappa_1,\ldots,t_s+\kappa_s})$, so $\mathcal{Q}_r \subseteq \mathrm{PAut}(\mathcal{H}^{t_1+\kappa_1,\ldots,t_s+\kappa_s})$. Moreover, if $\Phi(\mathcal{P}_r)$ is an r-PD-set for $\mathcal{H}^{t_1,\ldots,t_s}$, by Theorem 36, the matrices $(\mathcal{M}_i^{-1})^*$ for $i \in \{0,\ldots,r\}$ share no row in common. Clearly, the extended matrices $(\mathcal{M}_i^{-1}(\kappa_1,\ldots,\kappa_s))^*$ do not share any row either. □

4.4.2 Permutation representation

In the second construction, the elements of $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ are considered as permutations in $\mathrm{Sym}(n)$, where $n = p^{s(t_1-1)+(s-1)t_2+\cdots+t_s}$. Let $\sigma \in \mathrm{Sym}(n)$ and let q be a positive integer, then we define $q\sigma \in \mathrm{Sym}(qn)$ as the permutation that acts as σ in each of the following sets of coordinate positions: $\{1,\ldots,n\}$, $\{n+1,\ldots,2n\}$, $\{2n+1,\ldots,3n\}$, $\ldots$, $\{(q-1)n+1,\ldots,qn\}$.

Proposition 46. *Let S be an r-PD-set of size ℓ for $H^{t_1,\ldots,t_s}$ of length n with information set I. Then, $pS = \{p\sigma : \sigma \in S\}$ is an r-PD-set of size ℓ for $H^{t_1,\ldots,t_{s-1},t_s+1}$, with respect to any information set $I' = I \cup \{j+n\}$ with $j \in I$.*

Proof. Let $\mathcal{H} = \mathcal{H}^{t_1,\ldots,t_s}$, $H = \Phi(\mathcal{H})$, $\mathcal{H}^{(s)} = \mathcal{H}^{t_1,\ldots,t_{s-1},t_s+1}$ and $H^{(s)} = \Phi(\mathcal{H}^{(s)})$. Using the recursive construction given in (2.21), we obtain

$$\begin{aligned} H^{(s)} &= \{\Phi((h,h,h,\ldots,h) + \lambda(\mathbf{0},\mathbf{p}^{s-1},2\mathbf{p}^{s-1},\ldots,(p-1)\mathbf{p}^{s-1})) : \\ &\qquad h \in \mathcal{H}, \lambda \in \mathbb{Z}_p\} \\ &= \{(\Phi(h),\Phi(h+\lambda\mathbf{p}^{s-1}),\Phi(h+\lambda 2\mathbf{p}^{s-1}),\ldots,\Phi(h+\lambda(p-1)\mathbf{p}^{s-1})) : \\ &\qquad h \in \mathcal{H}, \lambda \in \mathbb{Z}_p\}, \end{aligned}$$

where $\mathbf{0}$ and $\mathbf{p}^{s-1}$ are the vectors with 0 and p^{s-1} in all components, respectively. We have that $\Phi(h+\lambda\mu\mathbf{p}^{s-1}) = \Phi(h) + \lambda\mu\Phi(\mathbf{p}^{s-1}) = \Phi(h) + \lambda\mu\mathbf{1}$ for any $\lambda,\mu \in \mathbb{Z}_p$ [BFV22a]. Therefore,

$$H^{(s)} = \{(h', h'+\lambda\mathbf{1}, h'+\lambda\mathbf{2},\ldots,h'+\lambda(\mathbf{p}-\mathbf{1})) : h' \in H, \lambda \in \mathbb{Z}_p\}.$$

If $\sigma \in \mathrm{PAut}(H)$, then $\sigma(x) = y \in H$ for any $x \in H$. Consider an element $\mathbf{x} = (x, x + \lambda\mathbf{1}, \ldots, x + \lambda(\mathbf{p} - \mathbf{1})) \in H^{(s)}$. Then,

$$\begin{aligned}(p\sigma)(x, x + \lambda\mathbf{1}, \ldots, x+\lambda(\mathbf{p} - \mathbf{1})) \\ &= (\sigma(x), \sigma(x) + \sigma(\lambda\mathbf{1}), \ldots, \sigma(x) + \sigma(\lambda(\mathbf{p} - \mathbf{1}))) \\ &= (y, y + \lambda\mathbf{1}, \ldots, y + \lambda(\mathbf{p} - \mathbf{1})) \in H^{(s)},\end{aligned}$$

which means that $p\sigma \in \mathrm{PAut}(H^{(s)})$.

Let $I \subseteq \{1, \ldots, n\}$ be an information set for H. Define $I' = I \cup \{j + n\}$, for any $j \in I$. We have that $\mathbf{x}|_{I'} = (x|_I, x_j + \lambda)$, for any $\mathbf{x} = (x, x + \lambda\mathbf{1}, \ldots, x + \lambda(\mathbf{p} - \mathbf{1})) \in H^{(s)}$, where $x \in H$ and $\lambda \in \mathbb{Z}_p$. Since there are $p^{st_1+(s-1)t_2+\cdots+t_s}$ different possible values of $x|_I$ and p possible values of λ, we obtain $p^{st_1+(s-1)t_2+\cdots+2t_{s-1}+t_s+1}$ different elements $\mathbf{x}|_{I'}$, which means that I' is an information set for $H^{(s)}$.

Consider an error vector $e = (e^1, \ldots, e^p) \in \mathbb{Z}_p^{pn}$ of weight $\mathrm{wt}_H(e) \leq r$, where $e^k = (e_1^k, \ldots, e_n^k) \in \mathbb{Z}_p^n$ for $k \in \{1, \ldots, p\}$. In order for pS to be an r-PD-set for $H^{(s)}$ with respect to I', there must be an element $p\sigma \in pS$ such that $\mathrm{wt}_H((p\sigma)(e)|_{I'}) = 0$. Note that $p\sigma(e) = (\sigma(e^1), \ldots, \sigma(e^p))$. Consider the vector $\hat{e} = (\hat{e}_1, \ldots, \hat{e}_n) \in \mathbb{Z}_p^n$ such that $\hat{e}_i = 1$ if $e_i^1 > 0$ or $e_i^2 > 0$, and $\hat{e}_i = 0$ otherwise, $i \in \{1, \ldots, n\}$. Since $\mathrm{wt}_H(\hat{e}) \leq r$, there exists $\sigma \in S$ such that $\mathrm{wt}_H(\sigma(\hat{e})|_I) = 0$. Therefore, $\mathrm{wt}_H(\sigma(e^1)|_I) = 0$ and $\mathrm{wt}_H(\sigma(e^2)|_I) = 0$, hence $\mathrm{wt}_H((p\sigma(e))|_{I\cup\{j+n:j\in I\}}) = 0$. Since $I' \subseteq I \cup \{j + n : j \in I\}$, we obtain that pS is an r-PD-set for $H^{(s)}$ with information set I'. □

Note that Proposition 46 uses directly permutations from $\mathrm{PAut}(H^{t_1,\ldots,t_s})$, without assuming that they come from elements in $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$. This means that one can also use r-PD-sets for an r that may exceed the upper bound given by Corollary 38. This is not the case in the following proposition, since if S is an r-PD-set for $H^{t_1,\ldots,t_s}$, then it is not always true that $p^{s-i+1}S$ is an r-PD-set for $H^{(i)}$ for $i \in \{1, \ldots, s-1\}$, where $H^{(i)} = \Phi(\mathcal{H}^{(i)})$ and $\mathcal{H}^{(i)} = \mathcal{H}^{t_1,\ldots,t_{i-1},t_i+1,t_{i+1},\ldots,t_s}$. This is because if $\sigma \in \mathrm{PAut}(H^{t_1,\ldots,t_s})$, it is generally not true that $p^{s-i+1}\sigma \in \mathrm{PAut}(H^{(i)})$. Instead, we have to assume that the r-PD-sets come from sets in $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ and extend each permutation

$\sigma \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ as $p^{s-i+1}\sigma \in \mathrm{PAut}(\mathcal{H}^{(i)})$ before applying the map Φ to obtain permutations in $\mathrm{PAut}(H^{(i)})$.

Proposition 47. *Let $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ such that $\Phi(\mathcal{S})$ is an r-PD-set of size ℓ for $H^{t_1,\ldots,t_s}$ with information set $I = \Phi(\mathcal{I})$, where $\mathcal{I}$ is an additive information set for $\mathcal{H}^{t_1,\ldots,t_s}$. Then, for any $i \in \{1,\ldots,s\}$, $\Phi(p^{s-i+1}\mathcal{S})$ is an r-PD-set of size ℓ for $H^{t'_1,\ldots,t'_s}$, with $t'_i = t_i + 1$ and $t'_j = t_j$ for any $j \neq i$, with respect to any information set $I' = \Phi(\mathcal{I} \cup \{j+n\})$ with $j \in I$, where n is the length of $\mathcal{H}^{t_1,\ldots,t_s}$.*

Proof. We follow a similar argument to the one given in Proposition 46, with the difference that $\mathcal{S}$ is a subset of $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ and not of $\mathrm{PAut}(H^{t_1,\ldots,t_s})$. Let $\mathcal{H} = \mathcal{H}^{t_1,\ldots,t_s}$, $H = \Phi(\mathcal{H})$, $\mathcal{H}^{(i)} = \mathcal{H}^{t'_1,\ldots,t'_s}$ and $H^{(i)} = \Phi(\mathcal{H}^{(i)})$. Taking into account that $\mathcal{H}^{(i)}$ is constructed using (2.21),

$$H^{(i)} = \{(\Phi(h), \Phi(h+\lambda\mathbf{p}^{i-1}), \Phi(h+\lambda 2\mathbf{p}^{i-1}), \ldots, \Phi(h+\lambda(p^{s-i+1}-1)\mathbf{p}^{i-1})) \\ : h \in \mathcal{H}, \lambda \in \mathbb{Z}_{p^{s-i+1}}\}.$$

If $\tau \in \mathrm{PAut}(\mathcal{H})$, then

$$\begin{aligned}(p^{s-i+1}\tau)(\mathbf{h}) &= (\sigma(h), \sigma(h)+\sigma(\lambda\mathbf{p^{i-1}}), \ldots, \sigma(x)+\sigma(\lambda(p^{s-i+1}-1)\mathbf{p^{i-1}})) \\ &= (\sigma(h), \sigma(h)+\lambda\mathbf{p^{i-1}}, \ldots, \sigma(h)+\lambda(p^{s-i+1}-1)\mathbf{p^{i-1}}) \in \mathcal{H}^{(i)},\end{aligned}$$

for any $\mathbf{h} = (h, h+\lambda\mathbf{p^{i-1}}, \ldots, h+\lambda(p^{s-i+1}-1)\mathbf{p^{i-1}}) \in \mathcal{H}^{(i)}$, with $h \in \mathcal{H}$ and $\lambda \in \mathbb{Z}_{p^{s-i+1}}$. Therefore, $(p^{s-i+1}\tau) \in \mathrm{PAut}(\mathcal{H}^{(i)})$ and $\Phi(p^{s-i+1}\tau) \in \mathrm{PAut}(H^{(i)})$.

By Proposition 30, the set $\mathcal{I} \cup \{n+1\}$ is an additive information set for $\mathcal{H}^{(i)}$. In fact, in the proof we also show that any set $\mathcal{I} \cup \{x\}$, for $x \in \{n+1,\ldots,2n\}$ is also an information set. In particular $\mathcal{I}' = \mathcal{I} \cup \{j+n\}$, for any $j \in \mathcal{I}$, is an information set for $\mathcal{H}^{(i)}$ and $I' = \Phi(\mathcal{I}')$ is an information set for $H^{(i)}$. Note that I' has $s-i+1$ more coordinates than I.

Finally, consider an error vector $e = (e^1, \ldots, e^{p^{s-i+1}})$ of weight $\mathrm{wt}_H(e) \leq r$, where $e^k = (e^k_1, \ldots, e^k_n) \in \mathbb{Z}_p^n$ for $k \in \{1,\ldots,p^{s-i+1}\}$. Define the vector $\hat{e} = (\hat{e}_1, \ldots, \hat{e}_n) \in \mathbb{Z}_p^n$ that satisfies $\hat{e}_m = 1$ if $e^1_m \neq 0$ or $e^2_m \neq 0$, and $\hat{e}_m = 0$

otherwise, $m \in \{1, \ldots, n\}$. Since $\mathrm{wt}_H(\hat{e}) \leq r$, there exists $\tau \in \mathcal{S}$ such that $\mathrm{wt}_H(\Phi(\tau)(\hat{e})|_I) = 0$. Thus, $\mathrm{wt}_H(\Phi(\tau)(e^1)|_I) = 0$ and $\mathrm{wt}_H(\Phi(\tau)(e^2)|_I) = 0$. Note that $\Phi(p^{s-i+1}\tau)(e) = (\Phi(\tau)(e^1), \ldots, \Phi(\tau)(e^{p^{s-i+1}}))$. Therefore,

$$\mathrm{wt}_H(\Phi(p^{s-i+1}\tau)(e)|_{I\cup\{j+p^{s-1}n : j\in I\}}) = \mathrm{wt}_H(\Phi(\tau)(e^1)|_I) + \mathrm{wt}_H(\Phi(\tau)(e^2)|_I) = 0.$$

Since $I' \subseteq I \cup \{j + p^{s-1}n : j \in I\}$, this implies that $\Phi(p^{s-i+1}\mathcal{S})$ is an r-PD-set for $H^{(i)}$ with information set I'. □

Remark 48. *By the definition of $p^{s-i+1}\tau$ and Φ, we have that $\Phi(p^{s-i+1}\tau) = p^{s-i+1}\Phi(\tau)$, for any $\tau \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ and $i \in \{1, \ldots, s\}$. By the proof of Proposition 47, $p^{s-i+1}\tau \in \mathrm{PAut}(\mathcal{H}^{(i)})$, so $p^{s-i+1}\Phi(\tau) \in \mathrm{PAut}(H^{(i)})$, where $\mathcal{H}^{(i)}, H^{(i)}$ are defined as in this proof.*

Example 49. *Consider the $\mathbb{Z}_{27}$-linear GH code $H^{3,0,0}$ as in Example 41. We have that $\Phi(\mathcal{P}_{11})$ is an 11-PD-set of size 12, where $\mathcal{P}_{11}$ can be identified by the set of permutations $\{\mathrm{id}, \tau_1, \ldots, \tau_{11}\} \subseteq \mathrm{Sym}(729)$. Then, by Proposition 46 or Proposition 47, we know that the following subset of* $\mathrm{Sym}(19683)$*:*

$$\{3\Phi(\mathrm{id}), 3\Phi(\tau_1), \ldots, 3\Phi(\tau_{11})\}$$

is an 11-PD-set for the $\mathbb{Z}_{27}$-linear GH code $H^{3,0,1}$, with information set $\Phi(\mathcal{I}_{3,0,1}) = \Phi(\mathcal{I}_{3,0,0}) \cup \Phi^{(3)}(\{730\}) = \{1, 2, 4, 10, 11, 13, 244, 245, 247, 6562\}$. Similarly, by Proposition 47 and Remark 48, we know that the following subset of $\mathrm{Sym}(59049)$*:*

$$\{9\Phi(\mathrm{id}), 9\Phi(\tau_1), \ldots, 9\Phi(\tau_{11})\}$$

is an 11-PD-set for the $\mathbb{Z}_{27}$-linear GH code $H^{3,1,0}$, with information set $\Phi(\mathcal{I}_{3,1,0}) = \Phi(\mathcal{I}_{3,0,0}) \cup \Phi^{(2)}(\{730\}) = \{1, 2, 4, 10, 11, 13, 244, 245, 247, 6562, 6565\}$. In general, we can construct r-PD-sets for $H^{3,0,1}$ and $H^{3,1,0}$ for any $r \leq f_3^{3,0,0} = 242$.

We could also use Proposition 47 in order to obtain an 11-PD-set for the $\mathbb{Z}_{27}$-linear GH code $H^{4,0,0}$, or in general an r-PD-set for any $r \leq f_3^{3,0,0} = 242$. However, in this case, we can construct an r-PD-set directly, from the explicit construction presented in Section 4.3, for any $r \leq f_3^{4,0,0} = 4919$.

Corollary 50. *Let $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ such that $\Phi(\mathcal{S})$ is an r-PD-set of size ℓ for $H^{t_1,\ldots,t_s}$ with information set I. Then, $\Phi(p^{si_1+(s-1)i_2+\cdots+i_s}\mathcal{S})$ is an r-PD-set of size ℓ for $H^{t_1+i_1,t_2+i_2,\ldots,t_s+i_s}$, with the information set obtained by applying recursively Proposition 30, for any $i_1, i_2, \ldots, i_s \geq 0$.*

Corollary 51. *If $\mathcal{P}_r = \{\mathcal{N}_0^{-1}, \ldots, \mathcal{N}_r^{-1}\}$, as defined in Section 4.3 for the $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,0,\ldots,0}$, then $\Phi(p^{(s-1)t_2+\cdots+t_s}\mathcal{P}_r)$ is an r-PD-set of size $r+1$ for the $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,\ldots,t_s}$, for all $t_2, \ldots, t_s \geq 0$, $t_1 \geq 2$ and $2 \leq r \leq f_p^{t_1,0,\ldots,0}$. Similarly, if $\mathcal{P}_r = \{\mathcal{M}_0^{-1}, \ldots, \mathcal{M}_r^{-1}\}$, as defined in Section 4.3 for the $\mathbb{Z}_{p^s}$-linear GH code $H^{1,0,\ldots,0,t_i,0,\ldots,0}$. Then, $\Phi(p^a\mathcal{P}_r)$, where $a = s(t_1-1)+\cdots+(s-i+2)t_{i-1}+(s-i)t_{i+1}+\cdots+t_s$, is an r-PD-set of size $r+1$ for the $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,\ldots,t_s}$, for all $t_2, \ldots, t_s \geq 0$, $t_1, t_i \geq 1$ and $2 \leq r \leq f_p^{1,0,\ldots,0,t_i,0,\ldots,0}$.*

Depending on the type of the $\mathbb{Z}_{p^s}$-linear GH code, the largest r allowed by Corollary 51 may be either $f_p^{t_1,0,\ldots,0}$ or one of $f_p^{1,0,\ldots,0,t_i,0,\ldots,0}$, for $i \in \{2, \ldots, s\}$. Let us define

$$\tilde{f}_p^{t_1,\ldots,t_s} = \max\{f_p^{t_1,0,\ldots,0}, f_p^{1,t_2,0,\ldots,0}, \ldots, f_p^{1,0,\ldots,0,t_s}\} \leq f_p^{t_1,\ldots,t_s}. \tag{4.12}$$

If $\tilde{f}_p^{t_1,\ldots,t_s} = f_p^{t_1,0,\ldots,0}$, we achieve the largest r using the explicit construction to obtain $\mathcal{P}_r$ for $H^{t_1,0\ldots,0}$ and then extending the r-PD-set as $\Phi(p^{(s-1)t_2+\cdots+t_s}\mathcal{P}_r)$. However, if $\tilde{f}_p^{t_1,\ldots,t_s} = f_p^{1,0,\ldots,0,t_i,0,\ldots,0}$, we achieve the largest r by using the explicit construction to obtain $\mathcal{P}_r$ for $H^{1,0,\ldots,0,t_i,0,\ldots,0}$ and then extending the PD-set as $\Phi(p^{s(t_1-1)+\cdots+(s-i+2)t_{i-1}+(s-i)t_{i+1}+\cdots+t_s}\mathcal{P}_r)$. We also define a parameter, $h_p^{t_1,\ldots,t_s}$, to indicate which construction is selected. If $\tilde{f}_p^{t_1,\ldots,t_s} = f_p^{t_1,0,\ldots,0}$, then $h_p^{t_1,\ldots,t_s} = 1$. Otherwise, $h_p^{t_1,\ldots,t_s} = i \in \{2, \ldots, s\}$, where i is the minimum index such that $\tilde{f}_p^{t_1,\ldots,t_s} = f_p^{1,0,\ldots,0,t_i,0,\ldots,0}$.

Example 52. *Consider the $\mathbb{Z}_8$-linear Hadamard codes $H^{3,3,7}$, $H^{3,4,7}$ and $H^{3,3,8}$. By Corollary 38, we have $f_2^{3,0,0} = 20$, $f_2^{1,3,0} = 15$, $f_2^{1,0,7} = 15$, $f_2^{1,4,0} = 50$, $f_2^{1,0,8} = 27$. Therefore, for example,*

- *for $H^{3,3,7}$, since $\tilde{f}_2^{3,3,7} = \max\{f_2^{3,0,0}, f_2^{1,3,0}, f_2^{1,0,7}\} = f_2^{3,0,0} = 20$, it is better to start by using the explicit construction for $H^{3,0,0}$.*

- *for* $H^{3,4,7}$*, since* $\tilde{f}_2^{3,4,7} = f_2^{1,4,0} = 50$*, it is better to start with the explicit construction for* $H^{1,4,0}$*.*
- *for* $H^{3,3,8}$*, since* $\tilde{f}_2^{3,3,8} = f_2^{1,0,8} = 27$*, it is better to start with the explicit construction for* $H^{1,0,8}$*.*

4.5 Computational results

The explicit constructions presented in Section 4.3 give r-PD-sets of size $r+1$ with an r that reaches up to the upper bound given by Corollary 38. However, these constructions are only defined for some specific $\mathbb{Z}_{p^s}$-linear GH codes: $H^{t_1,0,\dots,0}$ and $H^{1,0,\dots,0,t_i,0,\dots,0}$, with $t_1 \geq 2$ and $t_i \geq 1$, respectively. The recursive constructions presented in Section 4.4 allow to obtain r-PD-sets for all $\mathbb{Z}_{p^s}$-linear GH codes $H^{t_1,\dots,t_s}$, but they may not achieve the upper bound. Indeed, for the codes where the explicit constructions can not be applied, $r \leq \tilde{f}_p^{t_1,\dots,t_s} < f_p^{t_1,\dots,t_s}$, so other strategies are necessary in order to achieve a value of r closer to the theoretical upper bound $f_p^{t_1,\dots,t_s}$.

In this section, we present some computational results, obtained by using the computer algebra system Magma [BCFS19]. These results show that we can increase the value of r for $\mathbb{Z}_{p^s}$-linear GH codes $H^{t_1,\dots,t_s}$, by looking for r-PD-sets randomly. We follow a similar method as the one used in [BV18]. That is, we generate sets $\mathcal{P}_r = \{\mathcal{M}_0, \dots, \mathcal{M}_r\}$ of $r+1$ random matrices in $\mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$ such that all rows from the matrices of $\{\mathcal{M}_0^*, \dots, \mathcal{M}_r^*\}$ are different. The sets are constructed incrementally, starting from different initial matrices $\mathcal{M}_0$ until the target value of r is achieved. Initially, the target value of r is defined as the upper bound $f_p^{t_1,\dots,t_s}$. If the method has generated $k < r$ matrices $\mathcal{M}_0^*, \dots, \mathcal{M}_k^*$, and fails to generate $\mathcal{M}_{k+1}^*$ in a defined time constraint, then it starts again from another initial matrix $\mathcal{M}_0^*$. If the target value r is not attained after a certain number of different initial matrices, then r is decreased by one and the process starts again. If, by decreasing r, it reaches the value of $\tilde{f}_p^{t_1,\dots,t_s}$, then the r-PD-set given by Corollary 51 is returned.

t_1	t_2	r_{old}	r	$f_2^{t_1,t_2}$
3	0	4	**4**	4
	1	6	**7**	7
	2	10	**11**	11
	3	16	**18**	20
	4	26	**31**	35
	5	42	**50**	63
4	0	15	**15**	15
	1	23	**23**	24
	2	36	**38**	41
	3	56	**62**	72
	4	91	**103**	127
	5	150	**172**	226
5	0	50	**50**	50
	1	72	**76**	84
	2	116	**124**	145
	3	187	**199**	255
	4	312	**321**	454
	5	518	**551**	818

Table 4.1: Maximum value r for which r-PD-sets were found for some codes H^{t_1,t_2}, with $p = 2$, using a non-deterministic method. Comparison with previous results, r_{old}, given in [BV18] and the upper bound $f_2^{t_1,t_2}$.

Table 4.1 shows the maximum values of r obtained for $\mathbb{Z}_4$-linear Hadamard codes H^{t_1,t_2}, with $3 \leq t_1 \leq 5$ and $0 \leq t_2 \leq 5$. They are compared with the values given in [BV18] and the upper bound $f_2^{t_1,t_2}$. The results from [BV18] were obtained by using a method that is currently implemented in the MAGMA function `PDSetHadamardCodeZ4(t1, t2 : AlgMethod := "Nondeterministic")` included in the official distribution [BCFS19]. We have corrected an error found in the implementation of this function and made some improvements, which has allow us to achieve larger values of r in this case. Then, we have generalized these functions to deal with $\mathbb{Z}_{p^s}$-linear GH codes. Table 4.2 shows the maximum values of r obtained for $\mathbb{Z}_8$-linear Hadamard codes H^{t_1,t_2,t_3}, with $t_1 = 3$, $0 \leq t_2 \leq 2$ and $0 \leq t_3 \leq 3$. The upper bounds $\tilde{f}_2^{3,t_2,t_3}$ and f_2^{3,t_2,t_3} are also shown in order to see the improvement with respect to the recursive construction, which is bounded by $\tilde{f}_2^{3,t_2,t_3}$, and

with respect to the theoretical maximum, given by f_2^{3,t_2,t_3}.

t_2	t_3	$\tilde{f}_2^{3,t_2,t_3}$	r	f_2^{3,t_2,t_3}
0	0	20	**20**	20
	1	20	**30**	31
	2	20	**46**	50
	3	20	**73**	84
1	0	20	**61**	63
	1	20	**94**	101
	2	20	**149**	169
	3	20	**242**	291
2	0	20	**189**	203
	1	20	**299**	340
	2	20	**476**	584
	3	20	**773**	1023

Table 4.2: Maximum value r for which r-PD-sets were found for some codes H^{3,t_2,t_3}, with $p = 2$, using a non-deterministic method. Comparison with the upper bound of the recursive constructions $\tilde{f}_2^{3,t_2,t_3}$ and the upper bound f_2^{3,t_2,t_3}.

The MAGMA function developed to construct r-PD-sets of size $r + 1$ for $\mathbb{Z}_{p^s}$-linear GH codes has been included in a new MAGMA package to deal with linear codes over $\mathbb{Z}_{p^s}$ [FTV23]. This package also allows the construction of $\mathbb{Z}_{p^s}$-linear GH codes, and includes functions related to generalized Gray maps, information sets, the process of encoding and decoding using permutation decoding, among others. The package is described in Chapter 7.

Chapter 5

Improving r-PD-sets for $\mathbb{Z}_{p^s}$-linear GH codes

In Chapter 4, a criterion was given to find r-PD-sets of size $r+1$ for a $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,\ldots,t_s}$, with an r up to the upper bound $f_p^{t_1,\ldots,t_s}$. Moreover, two explicit constructions were given which produce r-PD-sets of size $r+1$ for $\mathbb{Z}_{p^s}$-linear GH codes of type $(n; t_1, 0, \ldots, 0)$ and $(n; 1, 0, \ldots, 0, t_i, 0, \ldots, 0)$, with $t_1 \geq 2$ and $t_i \geq 1$. These two types of codes are either free or can be treated as such in regard to its permutation automorphism group. For the rest of $\mathbb{Z}_{p^s}$-linear GH codes, i.e. non-free codes, a recursive construction was given for r-PD-sets of size $r+1$, with an r only reaching up to $\tilde{f}_p^{t_1,\ldots,t_s} \leq f_p^{t_1,\ldots,t_s}$.

In this chapter, we present a new construction that improves the results of Chapter 4, by producing r-PD-sets of size $r+1$ for $\mathbb{Z}_{p^s}$-linear GH codes of any type, for values of r larger than $\tilde{f}_p^{t_1,\ldots,t_s}$ and closer to the theoretical upper bound $f_p^{t_1,\ldots,t_s}$. An earlier version of these results, for $\mathbb{Z}_8$-linear GH codes, was presented as a conference talk at *Rijeka Conference on Combinatiorial Objects and their Applications (RICCOTA 2023)*. The final version, for $\mathbb{Z}_{p^s}$-linear GH codes, has been accepted as a journal paper in [RTV24].

The chapter is organized as follows. In Section 5.1, new explicit constructions of r-PD-sets of size $r+1$, for values of r closer to a known upper bound, are described. In Section 5.2, we compare the obtained values of r with the

theoretical upper bound and also with the computational results, given in Chapter 4.

5.1 New r-PD-sets for non-free codes

In this section, we present new constructions of r-PD-sets of size $r+1$, which are also suitable for any non-free $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,\ldots,t_s}$, that is with $t_2+\cdots+t_s>0$. Recall that for free $\mathbb{Z}_{p^s}$-linear GH codes, r-PD-sets of size $r+1$ up to the upper bound were given in Chapter 4. In Section 5.2, we show that depending on the type of the code, these new constructions allow us to improve the previous results, that is, to obtain r-PD-sets for values of r larger than $\tilde{f}_p^{t_1,\ldots,t_s}$ and closer to the theoretical upper bound $f_p^{t_1,\ldots,t_s}$. In order to present the new constructions (Theorem 55 and Corollary 59), first we need to introduce the elements of a specific Galois ring with a structure that will be useful in the proof of Theorem 55.

Let $\mathcal{R}=\mathrm{GR}(p^{s(t_1-1)})$ be the *Galois extension* of dimension t_1-1 over $\mathbb{Z}_{p^s}$, which is isomorphic to any ring $\mathbb{Z}_{p^s}[x]/(h(x))$, where $h(x)$ is a monic basic irreducible polynomial over $\mathbb{Z}_{p^s}$ of degree t_1-1. A monic basic polynomial $h(x)$ over $\mathbb{Z}_{p^s}$ is called *irreducible* if $\bar{h}(x)$ is an irreducible polynomial over $\mathbb{Z}_p$, where $\bar{h}(x)$ is the polynomial obtained by taking the coefficients of $h(x)$ modulo p. Moreover, if $\bar{h}(x)$ is primitive, then $h(x)$ is said to be a *monic basic primitive polynomial* over $\mathbb{Z}_{p^s}$. If $f(x)$ is an irreducible polynomial dividing x^n-1 in $\mathbb{Z}_p[x]$, then there is a unique polynomial $h(x)$ over $\mathbb{Z}_{p^s}[x]$ such that $h(x)\mid(x^n-1)$ in $\mathbb{Z}_{p^s}[x]$ and $\bar{h}(x)=f(x)$. This unique polynomial $h(x)$ is called the *Hensel lift* of $f(x)$ to $\mathbb{Z}_{p^s}$. Moreover, if a polynomial of degree m is the Hensel lift of a monic primitive polynomial over $\mathbb{Z}_p$, then it always has a root of order p^m-1 [Wan03]. Let $h(x)$ be such a polynomial, with $m=t_1-1$. Let $\alpha\in\mathcal{R}$ be a root of $h(x)$ of order $\ell=p^{t_1-1}-1$. Then, the set $\mathcal{T}=\{0,1,\alpha,\alpha^2,\ldots,\alpha^{\ell-1}\}$ is called the *Teichmüller set*.

The *p-adic representation* of an element $y\in\mathcal{R}$ is given by

$$y=y_0+py_1+p^2y_2+\cdots+p^{s-1}y_{s-1},$$

where $y_0, \ldots, y_{s-1} \in \mathcal{T}$. Consider the sequence of elements $r_1, \ldots, r_{p^{s(t_1-1)}} \in \mathcal{R}$ lexicographically ordered. That is, $a_0 + pa_1 + \cdots + p^{s-1}a_{s-1} < b_0 + pb_1 + \cdots + p^{s-1}b_{s-1}$ if $a_j < b_j$ for the last j where a_j and b_j differ. From now on we refer to this order as *lexicographical order*.

We structure the ordered elements of $\mathcal{R}$ in s different tables: $A, A_p, \ldots, A_{p^{s-1}}$. First, we divide all elements in blocks of p^{t_1-1} consecutive elements, and then we place each block as a column of a table, denoted by A. Note that any two elements r_i, r_j from the same row of A satisfy that $i - j$ is a multiple of p^{t_1-1}, which implies that $r_i - r_j \in (p) \subset \mathcal{R}$. In order to use Lemma 54 in the construction of the r-PD-sets, we take sequences of t_1 consecutive elements in $\mathcal{R}$. Let d_p and h_p be the quotient and the remainder of the division of p^{t_1-1} by t_1, respectively. The last h_p rows of this table are discarded, resulting in a table of $t_1 d_p$ rows and $p^{(s-1)(t_1-1)}$ columns, denoted by A_p. Table A_{p^k}, for $k \in \{2, \ldots, s-1\}$, is constructed by taking as the i-th column the vertical concatenation of consecutive columns in $A_{p^{k-1}}$, from the $(p^{t_1-1}(i-1)+1)$-th column up to the $(p^{t_1-1}i)$-th. This process results in a table A_{p^k} with $t_1 d_{p^{k-1}} p^{t_1-1} = t_1 d_{p^k}$ rows and $p^{(s-k+1)(t_1-1)}/p^{t_1-1} = p^{(s-k)(t_1-1)}$ columns, where $d_{p^k} = p^{t_1-1} d_{p^{k-1}} = p^{(k-1)(t_1-1)} d_p$. Note that any two elements r_i, r_j from the same row of A_{p^k} satisfy that $i-j$ is a multiple of $p^{k(t_1-1)}$, which implies that $r_i - r_j \in (p^k) \subset \mathcal{R}$.

Example 53. *For $t_1 = 3$, $s = 3$, and $p = 2$, we have that $|\mathcal{R}| = 8^{t_1-1} = 64$, $d_2 = 1$, and $d_4 = 2^{t_1-1} d_2 = 4$. Tables A, A_2 and A_4 are of size $2^{t_1-1} \times 4^{t_1-1} = 4 \times 16$, $t_1 d_2 \times 4^{t_1-1} = 3 \times 16$, and $t_1 d_4 \times 2^{t_1-1} = 12 \times 4$, respectively. Below appears a representation of Tables A, A_2, and A_4, where instead of the elements $r_i \in \mathcal{R}$, only the corresponding index i is shown:*

$$A: \begin{array}{cccc} 1 & 5 & \cdots & 61 \\ 2 & 6 & \cdots & 62 \\ 3 & 7 & \cdots & 63 \\ \hline 4 & 8 & \cdots & 64 \end{array}, \quad A_2: \begin{array}{cccc} 1 & 5 & \cdots & 61 \\ 2 & 6 & \cdots & 62 \\ 3 & 7 & \cdots & 63 \\ \hline \end{array}, \quad A_4: \begin{array}{cccc} 1 & 17 & 33 & 49 \\ 2 & 18 & 34 & 50 \\ 3 & 19 & 35 & 51 \\ \hline 5 & 21 & 37 & 53 \\ \cdots & \cdots & \cdots & \cdots \\ 15 & 31 & 47 & 63 \end{array}.$$

Lemma 54. *Let $t_1 \geq 2$. Let $r_{i_1}, \ldots, r_{i_{t_1}}$ be a sequence of elements in $\mathcal{R}$. If they are consecutive in the lexicographical order, then $\{r_{i_2} - r_{i_1}, \ldots, r_{i_{t_1}} - r_{i_1}\}$ is a set of linearly independent vectors in their additive representation. Moreover, any permutation of the indices $i_1, \ldots, i_{t_1}$ preserves the linear independence of the set of vectors.*

Proof. Theorem 40 implies that any matrix

$$\mathcal{N}_i^* = \begin{pmatrix} 1 & r_{t_1 i+1} \\ 1 & r_{t_1 i+2} \\ \vdots & \vdots \\ 1 & r_{t_1(i+1)} \end{pmatrix}$$

satisfies that $\mathcal{N}_i^{-1} \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, where

$$\mathcal{N}_i = \begin{pmatrix} 1 & r_{t_1 i+1} \\ 0 & r_{t_1 i+2} - r_{t_1 i+1} \\ \vdots & \vdots \\ 0 & r_{t_1(i+1)} - r_{t_1 i+1} \end{pmatrix}.$$

In particular, $\mathcal{N}_i$ is invertible. Therefore, $\{r_{t_1 i+2} - r_{t_1 i+1}, \ldots, r_{t_1(i+1)} - r_{t_1 i+1}\}$ is a set of linearly independent vectors. The same argument applies for any sequence of t_1 consecutive elements in $\mathcal{R}$.

Assume $\{r_{i_2} - r_{i_1}, \ldots, r_{i_{t_1}} - r_{i_1}\}$ is a set of linearly independent vectors. Any permutation of the indices $i_2, \ldots, i_{t_1}$ preserves the set of vectors. We just need to consider the transposition of one of these indices with i_1. Without loss of generality, we choose index i_2 and consider the set $\{r_{i_1} - r_{i_2}, r_{i_3} - r_{i_2} \ldots, r_{i_{t_1}} - r_{i_2}\}$. If these are not linearly independent vectors, then

$$\lambda_1(r_{i_1} - r_{i_2}) + \lambda_3(r_{i_3} - r_{i_2}) + \cdots + \lambda_{t_1}(r_{i_{t_1}} - r_{i_2}) = 0$$

for certain $\lambda_1, \lambda_3, \ldots, \lambda_{t_1} \in \mathbb{Z}_{p^s}$, with some of them being non-zero. This equation can be rewritten, as

$$-\lambda_1(r_{i_2} - r_{i_1}) + \lambda_3(r_{i_3} - r_{i_1}) + \cdots + \lambda_{t_1}(r_{i_{t_1}} - r_{i_1})$$

$$- \lambda_3(r_{i_2} - r_{i_1}) - \cdots - \lambda_{t_1}(r_{i_2} - r_{i_1}) = 0$$
$$(-\lambda_1 - \lambda_3 - \cdots - \lambda_{t_1})(r_{i_2} - r_{i_1}) + \lambda_3(r_{i_3} - r_{i_1}) + \cdots + \lambda_{t_1}(r_{i_{t_1}} - r_{i_1}) = 0.$$

This contradicts the initial assumption of $\{r_{i_2} - r_{i_1}, \ldots, r_{i_{t_1}} - r_{i_1}\}$ being a set of linearly independent vectors. □

Next theorem gives an explicit construction of r-PD-sets of size $r+1$ for any $\mathbb{Z}_{p^s}$-linear GH code $H^{t_1,\ldots,t_s}$ with $t_1 \geq 2$, which allows us to improve previous known results, as shown in Tables 5.2 and 5.3.

Theorem 55. *Let $H^{t_1,\ldots,t_s}$ be a $\mathbb{Z}_{p^s}$-linear GH code of type $(n; t_1, \ldots, t_s)$ with $t_1 \geq 2$ and $t_2 + \cdots + t_s > 0$. There exist r-PD-sets of size $r+1$ for $H^{t_1,\ldots,t_s}$, with respect to the information set $\Phi(\mathcal{I}_{t_1,\ldots,t_s})$, for every*

$$r \leq g_p^{t_1,\ldots,t_s} = p^{(s-1)t_2+(s-2)t_3+\cdots+t_s}\alpha - 1, \tag{5.1}$$

where $\alpha = \tau d_{p^{s-1}}$ is the maximum value multiple of $d_{p^{s-1}} = p^{(s-2)(t_1-1)}d_p$, with $d_p = \lfloor \frac{p^{t_1}-1}{t_1} \rfloor$, such that the following condition is satisfied for each $k \in \{1, \ldots, s-1\}$:

$$\alpha \leq t_1 d_{p^{s-k}} \left\lfloor \frac{p^{(k-1)(t_1-1)}(p^{t_1-1} - \tau)}{t_{s-k+1} + \cdots + t_s} \right\rfloor \text{ when } t_{s-k+1} + \cdots + t_s > 0. \tag{5.2}$$

Proof. Let $\mathcal{R} = \mathrm{GR}(p^{s(t_1-1)})$ be the Galois ring of degree $t_1 - 1$ over $\mathbb{Z}_{p^s}$ and consider the sequence of elements $r_1, \ldots, r_{p^{s(t_1-1)}}$ of $\mathcal{R}$, following the lexicographical order. In order to use the result given by Theorem 36, we need to produce a set of matrices $\{\mathcal{M}_0^*, \ldots, \mathcal{M}_r^*\}$, such that $\mathcal{M}_i^{-1} \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, or equivalently $\mathcal{M}_i \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, for $0 \leq i \leq r$, and such that no two different matrices $(\mathcal{M}_i^{-1})^*, (\mathcal{M}_j^{-1})^*$, with $0 \leq i, j \leq r$ and $i \neq j$, have a row in common.

Consider the following matrix:

$$\mathcal{M}_i^* = \left(\begin{array}{c|c|c|c|c|c} 1 & r_{i_1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \hline 1 & r_{i_2} & & & & \\ \vdots & \vdots & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ 1 & r_{i_{t_1}} & & & & \\ \hline 1 & r_{i_{t_1+1}} & & & & \\ \vdots & \vdots & pI_{t_2} & \mathbf{0} & \dots & \mathbf{0} \\ 1 & r_{i_{t_1+t_2}} & & & & \\ \hline 1 & r_{i_{t_1+t_2+1}} & & & & \\ \vdots & \vdots & \mathbf{0} & p^2I_{t_3} & \dots & \mathbf{0} \\ 1 & r_{i_{t_1+t_2+t_3}} & & & & \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 1 & r_{i_{t_1+\cdots+t_{s-1}+1}} & & & & \\ \vdots & \vdots & \mathbf{0} & \mathbf{0} & \dots & p^{s-1}I_{t_s} \\ 1 & r_{i_{t_1+\cdots+t_{s-1}+t_s}} & & & & \end{array}\right), \tag{5.3}$$

where $r_{i_1}, \dots, r_{i_{t_1+\cdots+t_s}} \in \mathcal{R}$. Then,

$$\mathcal{M}_i = \left(\begin{array}{c|c|c|c|c|c} 1 & r_{i_1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \hline 0 & r_{i_2} - r_{i_1} & & & & \\ \vdots & \vdots & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ 0 & r_{i_{t_1}} - r_{i_1} & & & & \\ \hline 0 & \chi_1^{-1}(r_{i_{t_1+1}} - r_{i_1}) & & & & \\ \vdots & \vdots & I_{t_2} & \mathbf{0} & \dots & \mathbf{0} \\ 0 & \chi_1^{-1}(r_{i_{t_1+t_2}} - r_{i_1}) & & & & \\ \hline 0 & \chi_2^{-1}(r_{i_{t_1+t_2+1}} - r_{i_1}) & & & & \\ \vdots & \vdots & \mathbf{0} & I_{t_3} & \dots & \mathbf{0} \\ 0 & \chi_2^{-1}(r_{i_{t_1+t_2+t_3}} - r_{i_1}) & & & & \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 0 & \chi_{s-1}^{-1}(r_{i_{t_1+\cdots+t_{s-1}+1}} - r_{i_1}) & & & & \\ \vdots & \vdots & \mathbf{0} & \mathbf{0} & \dots & I_{t_s} \\ 0 & \chi_{s-1}^{-1}(r_{i_{t_1+\cdots+t_{s-1}+t_s}} - r_{i_1}) & & & & \end{array}\right), \tag{5.4}$$

where $\chi_k(a) = p^k a$, for every $a \in \mathbb{Z}_{p^s}$ and $1 \leq k \leq s-1$. Thus, the construction of $\mathcal{M}_i$ is only well-defined if $r_{i_{t_1+\cdots+t_k+j}} - r_{i_1} \in (p^k)$ for $1 \leq j \leq t_{k+1}$. Moreover, in order to ensure that $\mathcal{M}_i \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, the vectors $r_{i_2} - r_{i_1}, \ldots, r_{i_{t_1}} - r_{i_1}$ must be linearly independent. By Lemma 54, this is fulfilled if $r_{i_1}, \ldots, r_{i_{t_1}}$ are consecutive, following the lexicographical order. Therefore, the proof is reduced to determine the indices $i_1, \ldots, i_{t_1+\cdots+t_s} \in \{1, \ldots, p^{s(t_1-1)}\}$ for each matrix $\mathcal{M}_i$, $0 \leq i \leq r$, such that the following conditions are satisfied:

(i) the elements $r_{i_1}, \ldots, r_{i_{t_1}}$ must be consecutive in the lexicographically ordered sequence $r_1, \ldots, r_{p^{s(t_1-1)}}$,

(ii) $r_{i_{t_1+\cdots+t_h+j}} - r_{i_1} \in (p^h)$ for $1 \leq j \leq t_{h+1}$, $1 \leq h \leq s-1$, and

(iii) all indices $i_1, \ldots, i_{t_1+\cdots+t_s}$ must be distinct.

We begin by constructing the first t_1 rows of matrices $\mathcal{M}_i^*$, for $i \in \{0, \ldots \alpha - 1\}$. Since α is a multiple of $d_{p^{s-1}}$, we have $\alpha = \tau d_{p^{s-1}}$. First, we split the table $A_{p^{s-1}}$ in two subtables: $A^{(1)}_{p^{s-1}}$, containing the first τ columns, and $A^{(2)}_{p^{s-1}}$, containing the last $p^{t_1-1} - \tau$ columns. Then, we take the sequence of elements, beginning with the first element in the first column of $A^{(1)}_{p^{s-1}}$ and finishing with the last element in the column τ of $A^{(1)}_{p^{s-1}}$. We have a sequence of $t_1 d_{p^{s-1}} \tau = t_1 \alpha$ elements. The first t_1 elements of this sequence are placed in the first t_1 rows of matrix $\mathcal{M}_0^*$, the next t_1 elements are placed in the first t_1 rows of matrix $\mathcal{M}_1^*$, and so on, until matrix $\mathcal{M}_{\alpha-1}^*$. This ensures that condition (i) is satisfied for every $0 \leq i \leq \alpha - 1$. The elements of $A^{(2)}_{p^{s-1}}$ will be used later to fill the last $t_2 + \cdots + t_s$ rows of matrices $\mathcal{M}_i^*$, for $i \in \{0, \ldots \alpha - 1\}$.

Since $\alpha = \tau d_{p^{s-1}} = \tau p^{t_1-1} d_{p^{s-2}} = \tau p^{2(t_1-1)} d_{p^{s-3}} = \cdots = p^{(s-2)(t_1-1)} d_p$, the index $i \in \{0, \ldots, \alpha - 1\}$ can be decomposed, in a unique way, as

$$i = b_1 d_{p^{s-1}} + b_2 d_{p^{s-2}} + \cdots + b_{s-1} d_p + b_s, \tag{5.5}$$

where $b_1 \in \{0, \ldots, \tau - 1\}$, $b_2, \ldots, b_{s-1} \in \{0, \ldots, p^{t_1-1} - 1\}$, and $b_s \in \{0, \ldots, d_p - 1\}$. Similarly, the index $j \in \{1, \ldots, t_1 d_{p^{s-1}}\}$ corresponding to the j-th

row of $A^{(1)}_{p^{s-1}}$ can be decomposed in a unique way as

$$j = a_1 t_1 d_{p^{s-2}} + a_2 t_1 d_{p^{s-3}} + \cdots + a_{s-2} t_1 d_p + a_{s-1} t_1 + a_s + 1, \tag{5.6}$$

where $a_1, \ldots, a_{s-2} \in \{0, \ldots, p^{t_1-1} - 1\}$, $a_{s-1} \in \{0, \ldots, d_p - 1\}$, and $a_s \in \{0, \ldots, t_1 - 1\}$.

Any of the t_1 elements $r_{i_1}, \ldots, r_{i_{t_1}}$ can act as the first row of $\mathcal{M}^*_i$, $i \in \{0, \ldots, \alpha - 1\}$, by applying a permutation of rows, by Lemma 54. We refer to the element selected to be in the first row as the *leader* of $\mathcal{M}^*_i$. The leader plays an important role in each matrix, since it determines which elements $r_{i_j} \in \mathcal{R}$, $j \in \{t_1+1, \ldots, t_1+\cdots+t_s\}$, satisfy condition (ii). We take as leader of $\mathcal{M}^*_i$, the element in the x-th position, r_{i_x}, where

$$x = \left[(b_1 + b_2\tau + b_3\tau p^{t_1-1} + \cdots + b_s\tau p^{(s-2)(t_1-1)}) \pmod{t_1}\right] + 1 \tag{5.7}$$

and i is as in (5.5). Note that the leader r_{i_x} of $\mathcal{M}^*_i$ belongs to the j-th row of $A^{(1)}_{p^{s-1}}$, where $j = [it_1 \pmod{t_1 d_{p^{s-1}}}] + x$. Hence,

$$\begin{aligned} j =& b_2 t_1 d_{p^{s-2}} + b_3 t_1 d_{p^{s-3}} + \cdots + b_{s-1} t_1 d_p + b_s t_1 \\ &+ \left[(b_1 + b_2\tau + b_3\tau p^{t_1-1} + \cdots + b_s\tau p^{(s-2)(t_1-1)}) \pmod{t_1}\right] + 1. \end{aligned} \tag{5.8}$$

Consider two matrices $\mathcal{M}^*_i$ and $\mathcal{M}^*_{i'}$, where $i = b_1 d_{p^{s-1}} + b_2 d_{p^{s-2}} + \cdots + b_{s-1} d_p + b_s$ and $i' = b'_1 d_{p^{s-1}} + b'_2 d_{p^{s-2}} + \cdots + b'_{s-1} d_p + b'_s$, and their respective leaders $r_{i_x}, r_{i'_{x'}}$, where

$$\begin{aligned} x &= \left[(b_1 + b_2\tau + b_3\tau p^{t_1-1} + \cdots + b_s\tau p^{(s-2)(t_1-1)}) \pmod{t_1}\right] + 1 \text{ and} \\ x' &= \left[(b'_1 + b'_2\tau + b'_3\tau p^{t_1-1} + \cdots + b'_s\tau p^{(s-2)(t_1-1)}) \pmod{t_1}\right] + 1. \end{aligned}$$

Clearly, by the uniqueness of the decompositions given in (5.5) and (5.6), if $b_k \neq b'_k$ for some $k \in \{2, \ldots, s\}$, then r_{i_x} and $r_{i'_{x'}}$ belong to different rows of $A_{p^{s-1}}$. However, if $b_k = b'_k$ for all $k \in \{2, \ldots, s\}$, then r_{i_x} and $r_{i'_{x'}}$ belong to the same row of $A_{p^{s-1}}$ if and only if $b_1 = b'_1 \pmod{t_1}$. Since $b_1 \in \{0, \ldots, \tau - 1\}$, then each row of $A_{p^{s-1}}$ contains at most $\lceil \frac{\tau}{t_1} \rceil$ leaders.

Denote by $S^{(j)}_{p^{s-1}}$, for $j \in \{1, \ldots, t_1 d_{p^{s-1}}\}$, the set containing the elements

in the j-th row of $A^{(2)}_{p^{s-1}}$. For each matrix $\mathcal{M}^*_i$, if its leader belongs to the j-th row of $A_{p^{s-1}}$, then a subset of t_s distinct elements of $S^{(j)}_{p^{s-1}}$ is taken to fill the last t_s rows of this matrix. Note that any $s' \in S^{(j)}_{p^{s-1}}$ satisfies that $s' - r' \in (p^{s-1}) \subseteq \mathcal{R}$, where r' is the leader of $\mathcal{M}^*_i$, so the above condition (ii) is satisfied for $h = s-1$. In order to satisfy condition (iii), the elements of $S^{(j)}_{p^{s-1}}$ can only be selected once. Thus, if more than one matrix have a leader in the same row j of $A_{p^{s-1}}$, then disjoint subsets of t_s elements of $S^{(j)}_{p^{s-1}}$ are selected, one for each matrix. Since each row j of $A_{p^{s-1}}$ may have up to $\lceil \frac{\tau}{t_1} \rceil$ leaders, then we must ensure that

$$\lceil \frac{\tau}{t_1} \rceil \leq \frac{|S^{(j)}_{p^{s-1}}|}{t_s}.$$

It is easy to see that this is guaranteed by condition (5.2) for $k = 1$.

So far, we have selected the elements $r_{i_1}, \ldots, r_{i_{t_1}}$ and $r_{i_{t_1+\cdots+t_{s-1}+1}}, \ldots, r_{i_{t_1+\cdots+t_s}}$, for every $i \in \{0, \ldots, \alpha-1\}$, satisfying conditions (i), (ii), and (iii). In particular, in order to satisfy condition (ii) for $h = s-1$, we have used Table $A_{p^{s-1}}$, since any two elements r_i, r_j in the same row of $A_{p^{s-1}}$ satisfy $r_i - r_j \in (p^{s-1})$. After this step, the leaders are fixed for every matrix and the elements that have already been selected cannot be selected again in order to satisfy condition (iii).

In the next step, we select the elements $r_{i_{t_1+\cdots+t_{s-2}+1}}, \ldots, r_{i_{t_1+\cdots+t_{s-1}}}$, using the structure of the table $A_{p^{s-2}}$. They are chosen from the same row of $A_{p^{s-2}}$ as r_{i_x}, satisfying condition (ii) for $h = s-2$. In general, an iterative process takes place, for $k \in \{2, \ldots, s-1\}$, where $r_{i_{t_1+\cdots+t_{s-k}+1}}, \ldots, r_{i_{t_1+\cdots+t_{s-k+1}}}$ are selected from the same row of $A_{p^{s-k}}$ as the leader of the corresponding matrix, so that condition (ii) is satisfied for $h = s-k$. The remaining part of the proof ensures that this is possible, that is, it is seen that there are enough elements to select all r_{i_j}, for $j \in \{1, \ldots, t_1 + \cdots + t_s\}$ and $i \in \{0, \ldots, \alpha-1\}$, while satisfying these conditions.

Now, recall that the table $A_{p^{s-2}}$ has $t_1 d_{p^{s-2}}$ rows and $p^{2(t_1-1)}$ columns. Every element in the first $\frac{\alpha}{d_{p^{s-2}}} = p^{t_1-1}\tau$ columns of $A_{p^{s-2}}$ has already been selected as one of the elements $r_{i_1}, \ldots, r_{i_{t_1}}$, for some $i \in \{0, \ldots, \alpha-1\}$.

Moreover, some elements in the last $p^{2(t_1-1)} - p^{t_1-1}\tau$ columns of $A_{p^{s-2}}$ may have been selected as one of the elements $r_{i_{t_1+\cdots+t_{s-1}+1}}, \ldots, r_{i_{t_1+\cdots+t_s}}$, but some are still available in order to fill the remaining $t_2 + \cdots + t_{s-1}$ rows in each matrix.

Let $A^{(2)}_{p^{s-2}}$ be the subtable consisting of the last $p^{2(t_1-1)} - p^{t_1-1}\tau$ columns of $A_{p^{s-2}}$, and let $S^{(\ell)}_{p^{s-2}}$, for $\ell \in \{1, \ldots, t_1 d_{p^{s-2}}\}$, be the set containing the elements in the ℓ-th row of $A^{(2)}_{p^{s-2}}$. By construction, the ℓ-th row of $A_{p^{s-2}}$ and $A^{(2)}_{p^{s-2}}$ contains all elements from each j_{a_1}-th row of $A_{p^{s-1}}$ and $A^{(2)}_{p^{s-1}}$, respectively, where $j_{a_1} = a_1 t_1 d_{p^{s-2}} + \ell$ and $a_1 \in \{0, \ldots, p^{t_1-1} - 1\}$. Thus,

$$S^{(\ell)}_{p^{s-2}} = \bigcup_{0 \le a_1 \le p^{t_1-1}-1} S^{(a_1 t_1 d_{p^{s-2}} + \ell)}_{p^{s-1}}.$$

Note that ℓ can be decomposed in a unique way as $\ell = a_2 t_1 d_{p^{s-3}} + \cdots + a_{s-2} t_1 d_p + a_{s-1} t_1 + a_s + 1$, where $a_2, \ldots, a_{s-2} \in \{0, \ldots, p^{t_1-1} - 1\}$, $a_{s-1} \in \{0, \ldots, d_p - 1\}$, and $a_s \in \{0, \ldots, t_1 - 1\}$. Recall that for a matrix $\mathcal{M}^*_i$, where i is as in (5.5), we selected as leader the element in the x-th position, r_{i_x}, where x is as in (5.7), which belongs to the j-th row of $A_{p^{s-1}}$, where j is as in (5.8). At the same time, r_{i_x} also belongs to the ℓ-th row of $A_{p^{s-2}}$, where $\ell = j \pmod{t_1 d_{p^{s-2}}}$. That is,

$$\begin{aligned} \ell =& a_2 t_1 d_{p^{s-3}} + \cdots + a_{s-2} t_1 d_p + a_{s-1} t_1 + a_s + 1 \\ =& b_3 t_1 d_{p^{s-3}} + \cdots + b_{s-1} t_1 d_p + b_s t_1 \\ &+ \left[(b_1 + b_2\tau + b_3 \tau p^{t_1-1} + \cdots + b_s \tau p^{(s-2)(t_1-1)}) \pmod{t_1} \right] + 1. \end{aligned} \tag{5.9}$$

Consider two matrices $\mathcal{M}^*_i$ and $\mathcal{M}^*_{i'}$, where $i = b_1 d_{p^{s-1}} + b_2 d_{p^{s-2}} + \cdots + b_{s-1} d_p + b_s$ and $i' = b'_1 d_{p^{s-1}} + b'_2 d_{p^{s-2}} + \cdots + b'_{s-1} d_p + b'_s$, and their respective leaders $r_{i_x}, r_{i'_{x'}}$. Clearly, if $b_k \neq b'_k$ for a $k \in \{3, \ldots, s\}$, then r_{i_x} and $r_{i'_{x'}}$ belong to different rows of $A_{p^{s-1}}$. However, if $b_k = b'_k$ for all $k \in \{3, \ldots, s\}$, then r_{i_x} and $r_{i'_{x'}}$ belong to the same row ℓ of $A_{p^{s-2}}$ if and only if $b_1 + b_2\tau = (b'_1 + b'_2\tau) \pmod{t_1}$. Since $b_1 \in \{0, \ldots, \tau - 1\}$ and $b_2 \in \{0, \ldots, p^{t_1-1} - 1\}$, then each row of $A_{p^{s-2}}$ contains at most $\lceil \frac{p^{t_1-1}\tau}{t_1} \rceil$ leaders.

For each matrix $\mathcal{M}^*_i$, if its leader is in the ℓ-th row of $A_{p^{s-2}}$, then t_{s-1} distinct elements of $S^{(\ell)}_{p^{s-2}}$ are selected to serve as $r_{i_{t_1+\cdots+t_{s-2}+1}}, \ldots, r_{i_{t_1+\cdots+t_{s-1}}}$.

Note that any $s' \in S^{(\ell)}_{p^{s-2}}$ satisfies that $s' - r' \in (p^{s-2}) \subseteq \mathcal{R}$, for any r' in the ℓ-th row of $A_{p^{s-2}}$, so the above condition (ii) is satisfied for $h = s-2$. To ensure condition (iii), the elements of $S^{(\ell)}_{p^{s-2}}$ can only be selected once, so $t_{s-1} + t_s$ different elements from $S^{(\ell)}_{p^{s-2}}$ must be selected for each leader in the ℓ-th row of $A_{p^{s-2}}$. Since each row ℓ of $A_{p^{s-2}}$ may have up to $\lceil \frac{p^{t_1-1}\tau}{t_1} \rceil$ leaders, then we must ensure that

$$\lceil \frac{p^{t_1-1}\tau}{t_1} \rceil \leq \frac{|S^{(\ell)}_{p^{s-2}}|}{t_{s-1} + t_s}.$$

It is easy to see that this is guaranteed by condition (5.2) for $k = 2$.

Similarly, with an increasing ordering in $k \in \{3, \ldots, s-1\}$, we select the elements $r_{i_{t_1+\cdots+t_{s-k}+1}}, \ldots, r_{i_{t_1+\cdots+t_{s-k+1}}}$ using the structure provided by $A_{p^{s-k}}$. Let $A^{(2)}_{p^{s-k}}$ be the subtable with the last $p^{(k-1)(t_1-1)}(p^{t_1-1} - \tau)$ columns of $A_{p^{s-k}}$, and let $S^{(\ell)}_{p^{s-k}}$ be the set containing the elements in the ℓ-th row of $A^{(2)}_{p^{s-k}}$. By construction, we have

$$\begin{aligned} S^{(\ell)}_{p^{s-k}} &= \bigcup_{0 \leq a_{k-1} \leq p^{t_1-1}-1} S^{(a_{k-1}t_1 d_{p^{s-k}} + \ell)}_{p^{s-k+1}} \\ &\vdots \\ &= \bigcup_{0 \leq a_1, \ldots, a_{k-1} \leq p^{t_1-1}-1} S^{(a_1 t_1 d_{p^{s-2}} + \cdots + a_{k-1}t_1 d_{p^{s-k}} + \ell)}_{p^{s-1}}. \end{aligned} \tag{5.10}$$

Using a similar argument to the one used for $k = 2$, we see that each row of $A_{p^{s-k}}$ contains at most $\lceil \frac{p^{(k-1)(t_1-1)}\tau}{t_1} \rceil$ leaders. Moreover, any $s' \in S^{(\ell)}_{p^{s-k}}$ satisfies that $s' - r' \in (p^{s-k}) \subseteq \mathcal{R}$, for any r' in the ℓ-th row of $A_{p^{s-k}}$, so the above condition (ii) is satisfied for $h = s-k$. Since $S^{(\ell)}_{p^{s-k}}$ satisfies all equalities in (5.10), for each leader in the ℓ-th row of $A_{p^{s-k}}$, we must select $t_{s-k+1} + \cdots + t_s$ different elements in $S^{(\ell)}_{p^{s-k}}$. Therefore, we must ensure that

$$\lceil \frac{p^{(k-1)(t_1-1)}\tau}{t_1} \rceil \leq \frac{|S^{(\ell)}_{p^{s-k}}|}{t_{s-k+1} + \cdots + t_s},$$

which is guaranteed by condition (5.2).

By using this construction, we obtain a set of matrices $\{\mathcal{M}_0^*, \ldots, \mathcal{M}_{\alpha-1}^*\}$ such that $\mathcal{M}_i^* \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$ for all $i \in \{0, \ldots, \alpha-1\}$. Furthermore, for each matrix $\mathcal{M}_i^*$, we can obtain $p^{(s-1)t_2+\cdots+t_s}$ different matrices, $\mathcal{M}_{i,k}^* \in \mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, such that all rows from all matrices in $\{\mathcal{M}_{i,k}^* : 0 \le i \le \alpha-1, 0 \le k \le p^{(s-1)t_2+\cdots+t_s}-1\}$ are different. Define

$$\mathcal{M}_{i,k} = \left(\begin{array}{c|c|c|c|c|c} 1 & r_{i_1} & \mathbf{u}_2^{(k)} & \mathbf{u}_3^{(k)} & \ldots & \mathbf{u}_s^{(k)} \\ \hline 0 & r_{i_2}-r_{i_1} & & & & \\ \vdots & \vdots & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ 0 & r_{i_{t_1}}-r_{i_1} & & & & \\ \hline 0 & \chi_1^{-1}(r_{i_{t_1+1}}-r_{i_1}) & & & & \\ \vdots & \vdots & I_{t_2} & \mathbf{0} & \ldots & \mathbf{0} \\ 0 & \chi_1^{-1}(r_{i_{t_1+t_2}}-r_{i_1}) & & & & \\ \hline 0 & \chi_2^{-1}(r_{i_{t_1+t_2+1}}-r_{i_1}) & & & & \\ \vdots & \vdots & \mathbf{0} & I_{t_3} & \ldots & \mathbf{0} \\ 0 & \chi_2^{-1}(r_{i_{t_1+t_2+t_3}}-r_{i_1}) & & & & \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 0 & \chi_{s-1}^{-1}(r_{i_{t_1+\cdots+t_{s-1}+1}}-r_{i_1}) & & & & \\ \vdots & \vdots & \mathbf{0} & \mathbf{0} & \ldots & I_{t_s} \\ 0 & \chi_{s-1}^{-1}(r_{i_{t_1+\cdots+t_{s-1}+t_s}}-r_{i_1}) & & & & \end{array}\right), \quad (5.11)$$

where $\mathbf{u}_j^{(k)}$, $2 \le j \le s$, is a vector with t_j coordinates over $p^{j-1}\mathbb{Z}_{p^s}$. Note that there are $p^{(s-j+1)t_j}$ different vectors $\mathbf{u}_j^{(k)}$. Let $\mathcal{P} = \{\mathcal{M}_{i,k}^{-1} : 0 \le i \le \alpha-1, 0 \le k \le p^{(s-1)t_2+\cdots+t_s}-1\}$. By Theorem 36, $\Phi(\mathcal{P})$ is an r-PD-set of size $r+1$ for $H^{t_1,\ldots,t_s}$, with respect to the information set $\Phi(\mathcal{I}_{t_1,\ldots,t_s})$, for every $r \le p^{(s-1)t_2+\cdots+t_s}\alpha - 1$. □

Example 56. *Using the construction given by the proof of Theorem 55, we can construct a* 12287*-PD-set of size* 12288 *for the* $\mathbb{Z}_8$*-linear Hadamard code* $H^{4,2,4}$*. In this case, we have* $d_2 = 2$*,* $h_2 = 0$*, and* $d_4 = 16$*. First, tables* A_2 *and* A_4 *are constructed. The elements of* $\mathcal{R} = \mathrm{GR}(8^3)$*, the Galois ring of dimension* 3 *over* $\mathbb{Z}_8$*, are distributed in* A_2*, by columns, so that for any*

two elements $r_i, r_j \in \mathcal{R}$ in the same row, $r_i - r_j \in (2)$. Thus, table A_2 has $t_1 d_2 = 8$ rows and $4^{t_1-1} = 64$ columns. Since $h_2 = 0$, A_2 contains all the elements of $\mathcal{R}$. The elements of A_2 are also distributed in a table A_4, where each column is formed by the elements in $2^{t_1-1} = 8$ consecutive columns of A_2, so that for any two elements r_i, r_j in the same row, $r_i - r_j \in (4)$. Thus, table A_4 has $t_1 d_4 = 64$ rows and $2^{t_1-1} = 8$ columns. Figure 5.1 shows table A_4 and the transpose of table A_2, giving only the index i for each element $r_i \in \mathcal{R}$.

The maximum value of α satisfying conditions (5.12) and (5.13) is $\alpha = 48$. From the first $\alpha = 48$ blocks of $t_1 = 4$ consecutive elements of $\mathcal{R}$, which are placed in the first $\tau = \alpha/d_4 = 3$ columns of A_4, we construct the first t_1 rows of matrices $\mathcal{M}_0^*, \ldots, \mathcal{M}_{47}^*$. The bordered elements in table A_4 of Figure 5.1 are selected as the leaders for the corresponding matrices, and the ones with a light gray background are selected to construct the last $t_3 = 4$ rows of these matrices. The elements with a dark gray background in table A_2 of Figure 5.1 are selected to construct the remaining $t_2 = 2$ rows, 5-th and 6-th rows, of these matrices. By construction, the leaders $1, 66, 131, 12, 73, 138, \ldots$ are distributed cyclically among the t_1 positions of the blocks. Moreover, they are also distributed in a balanced way among the first t_1 rows of A_2. This ensures that there are enough elements of each class in order to fill the last $t_2 + t_3 = 6$ rows of these matrices. For example, the first matrix $\mathcal{M}_0^*$ is constructed as follows:

$$\mathcal{M}_0^* = \begin{pmatrix} 1 & r_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & r_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & r_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & r_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & r_{217} & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & r_{249} & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & r_{193} & 0 & 0 & 4 & 0 & 0 & 0 \\ 1 & r_{257} & 0 & 0 & 0 & 4 & 0 & 0 \\ 1 & r_{321} & 0 & 0 & 0 & 0 & 4 & 0 \\ 1 & r_{385} & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 6 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 6 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 0 \\ 1 & 4 & 4 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 1 & 0 & 4 & 4 & 0 & 0 & 0 & 0 & 4 & 0 \\ 1 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}.$$

Then,

$$\mathcal{M}_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Finally, we can obtain $4^2 2^4 = 256$ *different matrices* $\mathcal{M}^*_{i,k}$, $0 \leq k \leq 255$, *for each one of the matrices* $\mathcal{M}^*_i$, $0 \leq i \leq 47$, *which give us a* 12287*-PD-set of size* $48 \cdot 256 = 12288$ *for the code* $H^{4,2,4}$.

Remark 57. *The* $s-1$ *conditions* (5.2) *given by Theorem 55 for* $k \in \{1, \ldots, s-1\}$ *are independent and must be satisfied in order to obtain a valid value for* α. *For example, consider the case* $p = 2$ *and* $s = 3$. *From* (5.2) *we obtain the following two conditions:*

$$\text{for } k = 2, \quad \alpha \leq t_1 d_2 \left\lfloor \frac{4^{t_1-1} - 2^{t_1-1}\tau}{t_2 + t_3} \right\rfloor \text{ when } t_2 + t_3 > 0, \tag{5.12}$$

$$\text{for } k = 1, \quad \alpha \leq t_1 d_4 \left\lfloor \frac{2^{t_1-1} - \tau}{t_3} \right\rfloor \text{ when } t_3 > 0. \tag{5.13}$$

It is easy to see that condition (5.12) does not imply condition (5.13), and vice versa. For instance, for the $\mathbb{Z}_8$*-linear Hadamard code* $H^{4,2,4}$, *which is considered in Example 56, the maximum multiple of* $d_4 = 16$ *that satisfies both conditions is* $\alpha = 48$. *Let us denote the right-hand side of both restrictions by* $f_1(\alpha;\, t_1, t_2, t_3)$ *and* $f_2(\alpha;\, t_1, t_2, t_3)$, *respectively. Then,*

$$f_1(48;\, 4, 2, 4) = 48, \quad f_1(64;\, 4, 2, 4) = 40 < \alpha = 64,$$
$$f_2(48;\, 4, 2, 4) = 64, \quad f_2(64;\, 4, 2, 4) = 64.$$

Note that if $\alpha = 64$, which is the next multiple of $d_4 = 16$, condition (5.13) is fulfilled, but condition (5.12) is not satisfied. On the other hand, for the $\mathbb{Z}_8$-linear Hadamard code $H^{4,0,4}$, the maximum feasible value for α is $\alpha = 64$. For this value and the next multiple of $d_4 = 16$, $\alpha = 80$, the following restrictions are obtained:

$$f_1(64;\, 4, 0, 2) = 128, \quad f_1(80;\, 4, 0, 2) = 96,$$
$$f_2(64;\, 4, 0, 2) = 128, \quad f_2(80;\, 4, 0, 2) = 64 < \alpha = 80.$$

Thus, if $\alpha = 80$, condition (5.12) is fulfilled, but condition (5.13) is not satisfied.

Note that Theorem 55 can only be applied when $t_1 \geq 2$. For the $\mathbb{Z}_{p^s}$-linear GH codes $H^{1,t_2,\ldots,t_s} = H^{1,0,\ldots,0,t_j,\ldots,t_s}$, where $j = \min\{i \mid i \in \{2,\ldots,s\}, t_i > 0\}$, it is possible to obtain r-PD-sets of size $r+1$ by applying the recursive constructions presented in Chapter 4 as follows. Let $j' = \min\{i \mid i \in \{j,\ldots,s\}, t_i > 1\}$. First, we use Theorem 55 to obtain an r-PD-set for $H^{t_{j'},\ldots,t_s}$ with $r \leq g_p^{t_{j'},\ldots,t_s}$, and then, we use Corollary 50 to extend it to an r-PD-set for $H^{1,0,\ldots,0,t_j,\ldots,t_s}$. Next proposition allows us to present a new construction to obtain r-PD-sets of size $r+1$ for the $\mathbb{Z}_{p^s}$-linear GH codes $H^{1,t_2,\ldots,t_s} = H^{1,0,\ldots,0,t_j,\ldots,t_s}$ (see Corollary 59), which gives an r-PD-set with $r \leq g_p^{t_j+1,t_{j+1},\ldots,t_s}$. Note that $j \leq j'$ and $g_p^{t_j+1,t_{j+1},\ldots,t_s} \geq g_p^{t_{j'},\ldots,t_s}$.

Proposition 58. *Let $\mathcal{H} = \mathcal{H}^{t_1,\ldots,t_s}$ be a $\mathbb{Z}_{p^s}$-additive GH code of type $(n; t_1, t_2, \ldots, t_s)$ with $t_1 \geq 2$, and let $\mathcal{H}' = \mathcal{H}^{1,0,\ldots,0,t_1-1,\ldots,t_s}$ be a $\mathbb{Z}_{p^{s'}}$-additive GH code of type $(n'; 1, 0, \ldots, 0, t_1 - 1, t_2, \ldots, t_s)$ with $s' > s$. If there exists a set $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{H})$ such that $\Phi(\mathcal{S})$ is an r-PD-set of size $r+1$ for $H = \Phi(\mathcal{H})$, then there exists a set $\mathcal{P} \subseteq \mathrm{PAut}(\mathcal{H}')$ such that $\Phi(\mathcal{P})$ is an r-PD-set of size $r+1$ for $H' = \Phi(\mathcal{H}')$.*

Proof. Let $\mathcal{S} = \{\mathcal{M}_1^{-1}, \ldots, \mathcal{M}_{r+1}^{-1}\} \subseteq \mathrm{PAut}(\mathcal{H})$ such that $\Phi(\mathcal{S})$ is an r-PD-set for $H = \Phi(\mathcal{H})$. Since $\mathcal{M}_i$ is as in (4.2), we can partition it as

$$\mathcal{M}_i = \begin{pmatrix} 1 & a \\ \mathbf{0} & A \end{pmatrix}.$$

Then, we define the following matrix over $\mathbb{Z}_{p^{s'}}$:

$$\mathcal{N}_i = \begin{pmatrix} 1 & \chi(a) \\ \mathbf{0} & \iota(A) \end{pmatrix},$$

where χ and ι are maps from $\mathbb{Z}_{p^s}$ to $\mathbb{Z}_{p^{s'}}$ defined as $\chi(a) = p^{s'-s}a$ and $\iota(a) = a$, respectively. Clearly, if $\mathcal{M}_i \in \mathrm{PAut}(\mathcal{H})$, then $\mathcal{N}_i \in \mathrm{PAut}(\mathcal{H}')$.

Let $m_{i,j} = (a_j, \bar{m}_{i,j})$ and $n_{i,j} = (a_j, \bar{n}_{i,j})$ be the j-th rows of $\mathcal{M}_i$ and $\mathcal{N}_i$, respectively, where $a_1 = 1$ and $a_j = 0$ if $j > 1$. Note that $\bar{n}_{i,1} = p^{s'-s}\bar{m}_{i,1}$ for any $i \in \{1, \ldots, r+1\}$, and $\bar{n}_{i,j} = \bar{m}_{i,j}$ for any $i \in \{1, \ldots, r+1\}$ and $j \in \{2, \ldots, t_1 + \cdots + t_s\}$.

Consider also the j-th rows $m^*_{i,j} = (1, \bar{m}^*_{i,j})$ and $n^*_{i,j} = (1, \bar{n}^*_{i,j})$ of $\mathcal{M}^*_i$ and $\mathcal{N}^*_i$, respectively. We have

$$\begin{aligned}
&m^*_{i,1} = m_{i,1} = (1, \bar{m}_{i,1}),\\
&m^*_{i,j} = m_{i,1} + m_{i,j} = (1, \bar{m}_{i,1} + \bar{m}_{i,j}) \text{ if } j \in \{2, \ldots, t_1\},\\
&m^*_{i,j} = m_{i,1} + p^{k-1}m_{i,j} = (1, \bar{m}_{i,1} + p^{k-1}\bar{m}_{i,j})\\
&\qquad \text{if } j \in \{t_1 + \cdots + t_{k-1} + 1, \ldots, t_1 + \cdots + t_k\}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
&n^*_{i,1} = n_{i,1} = (1, \bar{n}_{i,1}),\\
&n^*_{i,j} = n_{i,1} + p^{s'-s}n_{i,j} = (1, \bar{n}_{i,1} + p^{s'-s}\bar{n}_{i,j}) \text{ if } j \in \{2, \ldots, t_1\},\\
&n^*_{i,j} = n_{i,1} + p^{s'-s+k-1}n_{i,j} = (1, \bar{n}_{i,1} + p^{s'-s+k-1}\bar{n}_{i,j})\\
&\qquad \text{if } j \in \{t_1 + \cdots + t_{k-1} + 1, \ldots, t_1 + \cdots + t_k\}.
\end{aligned}$$

Therefore, $\bar{n}^*_{i,j} = p^{s'-s}\bar{m}^*_{i,j}$ for any $i \in \{1, \ldots, r+1\}$ and $j \in \{1, \ldots, t_1 + \cdots + t_s\}$. By Theorem 36, all rows $m^*_{i,j} = (1, \bar{m}^*_{i,j})$ are different over $\mathbb{Z}_{p^s}$. Thus, all rows $n^*_{i,j} = (1, p^{s'-s}\bar{m}^*_{i,j})$ are different over $\mathbb{Z}_{p^{s'}}$. Using again Theorem 36, we obtain that $\Phi(\mathcal{P}) = \Phi(\{\mathcal{N}_1^{-1}, \ldots, \mathcal{N}_{r+1}^{-1}\})$ is an r-PD-set for $H' = \Phi(\mathcal{H}')$. □

Corollary 59. *Let $H^{1,t_2,\ldots,t_s}$ be a $\mathbb{Z}_{p^s}$-linear GH code of type $(n; 1, t_2, \ldots, t_s)$ and let $j \in \{2, \ldots, s\}$ be the minimum index such that $t_j > 0$. If $t_{j+1} + \cdots +$*

$t_s > 0$, then there exist r-PD-sets of size $r+1$ for $H^{1,t_2,\ldots,t_s}$, with respect to the information set $\Phi(\mathcal{I}_{1,t_2,\ldots,t_s})$, for every $r \leq g_p^{1,t_2,\ldots,t_s} = g_p^{t_j+1,t_{j+1},\ldots,t_s}$, where $g_p^{t_j+1,t_{j+1},\ldots,t_s}$ is defined as in (5.1).

Proof. By Theorem 55, there exist r-PD-sets of size $r+1$ for $H^{t_j+1,\ldots,t_s}$, with respect to the information set $\Phi(\mathcal{I}_{t_j+1,\ldots,t_s})$, for every $r \leq g_p^{t_j+1,\ldots,t_s}$. In fact, in the proof of Theorem 55, we see that these r-PD-sets, say S, can be obtained as $S = \Phi(\mathcal{S})$, where $\mathcal{S} \subseteq \mathrm{PAut}(\mathcal{H}^{t_j+1,\ldots,t_s})$. Therefore, Proposition 58 guarantees the existence of a set $\mathcal{P} \subseteq \mathrm{PAut}(\mathcal{H}^{1,t_2,\ldots,t_s})$ such that $\Phi(\mathcal{P})$ is an r-PD-set of size $r+1$ for $H^{1,t_2,\ldots,t_s}$ with respect to the information set $\Phi(\mathcal{I}_{1,t_2,\ldots,t_s})$. □

Example 60. *By Corollary 59, from the r-PD-set of size $r+1$ given in Example 56, we can obtain a 12287-PD-set of size 12288 for the $\mathbb{Z}_{2^s}$-linear Hadamard codes $H^{1,3,2,4}$, $H^{1,0,3,2,4}$, $H^{1,0,0,3,2,4}$, ... with $s = 4, 5, 6, \ldots$, respectively.*

Corollary 59 can be seen as a generalization of the construction of r-PD-sets of size $r+1$ for $\mathbb{Z}_{p^s}$-linear GH codes $H_i = H^{1,0,\ldots,0,t_i,0,\ldots,0}$ with $t_i > 0$ for all $i \in \{2, \ldots, s\}$, given by Corollary 43. The combination of both results implies that we can obtain an r-PD-set of size $r+1$ for any code $H^{1,t_2,\ldots,t_s}$.

5.2 Upper bound comparative analysis

Using the construction proposed in Theorem 55 and Corollary 59, r-PD-sets of size $r+1$ with $r \leq g_p^{t_1,\ldots,t_s}$ can be obtained for the $\mathbb{Z}_{p^s}$-linear codes $H^{t_1,\ldots,t_s}$ with $t_2 + \cdots + t_s > 0$. By Corollary 38, we have that $g_p^{t_1,\ldots,t_s} \leq f_p^{t_1,\ldots,t_s}$. In this section, we find new values of $p, t_1, \ldots, t_s$ for which this theoretical upper bound $f_p^{t_1,\ldots,t_s}$ is tight. Moreover, even when the upper bound is not reached, $g_p^{t_1,\ldots,t_s}$ approaches $f_p^{t_1,\ldots,t_s}$ considerably. Finally, computational results given in Section 4.5 are compared with the values of $g_p^{t_1,\ldots,t_s}$.

Table 5.1 shows the values of $g_2^{t_1,t_2,t_3}$ and $f_2^{t_1,t_2,t_3}$, where $t_1 \in \{3,4,5\}$ and $t_2, t_3 \in \{0,1,2,3,4\}$. Gray colored cells indicate that the upper bound is reached, that is, $g_2^{t_1,t_2,t_3} = f_2^{t_1,t_2,t_3}$. Note that $g_2^{t_1,0,0}$ is not defined for any

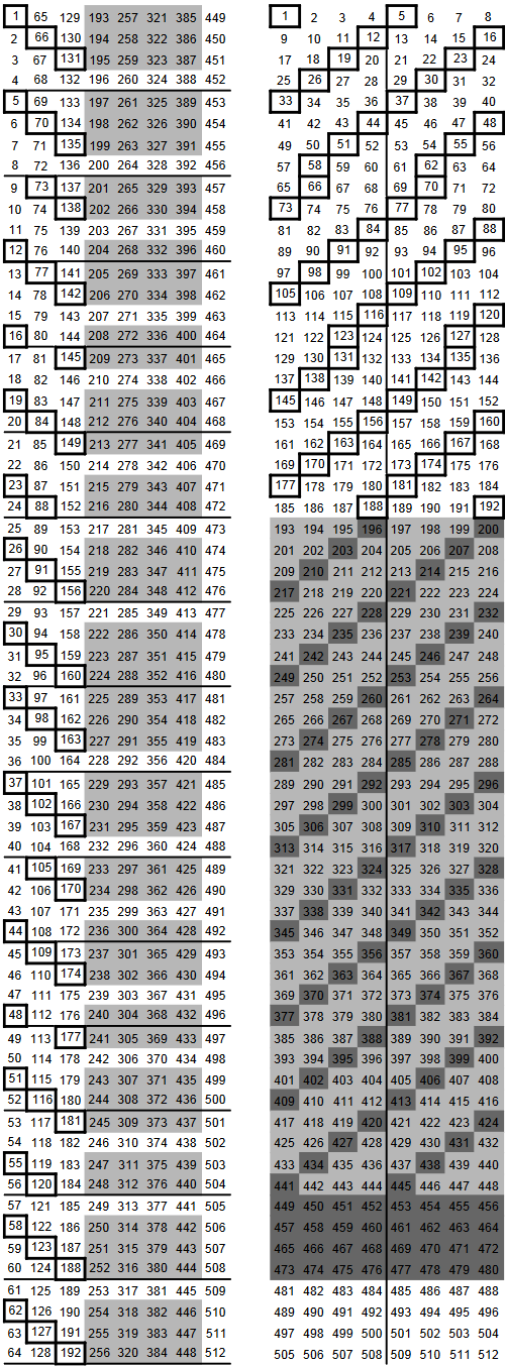

Figure 5.1: Table A_4 (left) and the transpose of A_2 (right), used for the construction of an 12287-PD-set of size 12288 for $H^{4,2,4}$ in Example 56.

$t_1 \geq 1$. We use the symbol $-$ to represent this undefined value. Moreover, note that there are cases where $g_2^{t_1,t_2,t_3}$ is defined but it is equal to -1, which means that the construction given by Theorem 55 is not able to produce an r-PD-set. As an illustration, see Example 56, where we construct an r-PD-set of size $r+1$ for the code $H^{4,2,4}$, with $r = g_2^{4,2,4} = 12287$, which does not reach the upper bound $f_2^{4,2,4} = 13106$, so there could be r-PD-sets of size $r+1$ for $H^{4,2,4}$ such that $g_2^{4,2,4} = 12287 < r \leq 13106 = f_2^{4,2,4}$.

		$t_3=0$		$t_3=1$		$t_3=2$		$t_3=3$		$t_3=4$	
t_1	t_2	g_2	f_2	g_2	f_2	g_2	f_2	g_2	f_2	g_2	f_2
3	0	$-$	20	23	31	31	50	31	84	-1	145
3	1	47	63	63	101	63	169	127	291	-1	511
3	2	127	203	127	340	255	584	511	1023	-1	1819
3	3	255	681	511	1169	1023	2047	2047	3639	-1	6552
3	4	1023	2339	2047	4095	4095	7280	-1	13106	-1	23830

		$t_3=0$		$t_3=1$		$t_3=2$		$t_3=3$		$t_3=4$	
t_1	t_2	g_2	f_2	g_2	f_2	g_2	f_2	g_2	f_2	g_2	f_2
4	0	$-$	127	191	203	255	340	511	584	1023	1023
4	1	383	408	639	681	1023	1169	2047	2047	3071	3639
4	2	1279	1364	2047	2339	4095	4095	6143	7280	12287	13106
4	3	4095	4680	8191	8191	12287	14562	24575	26213	32767	47661
4	4	16383	16383	24575	29126	49151	52427	65535	95324	131071	174761

		$t_3=0$		$t_3=1$		$t_3=2$		$t_3=3$		$t_3=4$	
t_1	t_2	g_2	f_2	g_2	f_2	g_2	f_2	g_2	f_2	g_2	f_2
5	0	$-$	818	1247	1364	1919	2339	3839	4095	6143	7280
5	1	2495	2729	4223	4680	7679	8191	12287	14562	21503	26213
5	2	8447	9361	15359	16383	24575	29126	43007	52427	86015	95324
5	3	30719	32767	49151	58253	86015	104856	172031	190649	294911	349524
5	4	98303	116507	172031	209714	344063	381299	589823	699049	1179647	1290554

Table 5.1: Columns g_2 and f_2 contain the values of $g_2^{t_1,t_2,t_3}$ and $f_2^{t_1,t_2,t_3}$, for $t_1 \in \{3,4,5\}$, $t_2, t_3 \in \{0,1,2,3,4\}$, respectively.

The results given in this chapter, using Theorem 55 and Corollary 59, allow us to achieve r-PD-sets of size $r+1$ up to $r \leq g_p^{t_1,\ldots,t_s}$. These results are usually better than the ones obtained in Chapter 4, where the given r-PD-sets are of size $r+1$ up to $r \leq \tilde{f}_p^{t_1,\ldots,t_s}$. However, there are some isolated cases where $g_p^{t_1,\ldots,t_s} < \tilde{f}_p^{t_1,\ldots,t_s}$, such as when $g_p^{t_1,\ldots,t_s} = -1$. There are some even more isolated cases in which $g_p^{t_1,\ldots,t_s}$ is not defined, such as when $t_2 = \cdots = t_s = 0$ for any t_1. In the latter case, $\tilde{f}_p^{t_1,0,\ldots,0} = f_p^{t_1,0,\ldots,0}$, so the upper bound can be achieved instead by using the explicit construction given

in Theorem 40.

Table 5.2 shows the values of g_2^{3,t_2,t_3}, $\tilde{f}_2^{3,t_2,t_3}$ and h_2^{3,t_2,t_3}, as defined in (4.12), where $t_2, t_3 \in \{0,1,2,3,4\}$. This table considers the same cases as the first subtable in Table 5.1, where $t_1 = 3$. As mentioned above, $g_2^{3,0,0}$ is not defined, but in this case the code is free, so we can use the explicit construction given in Theorem 40, obtaining $\tilde{f}_p^{3,0,0} = f_p^{3,0,0} = 20$. Note that $\tilde{f}_2^{3,4,t_3} = f_2^{1,4,0} = 50$ for any $t_3 \in \{0,1,2,3,4\}$, that is, $h_2^{3,4,t_3} = 2$. However, if $t_2 < 4$, then $\tilde{f}_2^{3,t_2,t_3} = f_2^{3,0,0} = 20$ for any $t_3 \in \{0,1,2,3,4\}$, that is, $h_2^{3,t_2,t_3} = 1$. Similarly, Table 5.3 shows the values of g_3^{2,t_2,t_3}, $\tilde{f}_3^{2,t_2,t_3}$ and h_3^{2,t_2,t_3}, as defined in (4.12), where $t_2, t_3 \in \{0,1,2\}$. In both tables, the maximum value between $g_p^{t_1,t_2,t_3}$ and $\tilde{f}_p^{t_1,t_2,t_3}$ is shown in bold type.

	$t_3=0$		$t_3=1$		$t_3=2$		$t_3=3$		$t_3=4$	
t_2	g_2	$\tilde{f}_2, h_2$	g_2	$\tilde{f}_2, h_2$	g_2	$\tilde{f}_2, h_2$	g_2	$\tilde{f}_2, h_2$	g_2	$\tilde{f}_2, h_2$
0	–	**20**, 1	**23**	20, 1	**31**	20, 1	**31**	20, 1	-1	**20**, 1
1	**47**	20, 1	**63**	20, 1	**63**	20, 1	**127**	20, 1	-1	**20**, 1
2	**127**	20, 1	**127**	20, 1	**255**	20, 1	**511**	20, 1	-1	**20**, 1
3	**255**	20, 1	**511**	20, 1	**1023**	20, 1	**2047**	20, 1	-1	**20**, 1
4	**1023**	50, 2	**2047**	50, 2	**4095**	50, 2	-1	**50**, 2	-1	**50**, 2

Table 5.2: Columns $g_2, \tilde{f}_2$, and h_2 contain the values of g_2^{3,t_2,t_3}, $\tilde{f}_2^{3,t_2,t_3}$, and h_2^{3,t_2,t_3}, for $t_2, t_3 \in \{0,1,2,3,4\}$, respectively.

	$t_3=0$		$t_3=1$		$t_3=2$	
t_2	g_3	$\tilde{f}_3, h_3$	g_3	$\tilde{f}_3, h_3$	g_3	$\tilde{f}_3, h_3$
0	–	**12**, 1	**17**	12, 1	**26**	12, 1
1	**53**	12, 1	**80**	12, 1	**80**	12, 1
2	**242**	26, 2	**728**	26, 2	-1	**26**, 2

Table 5.3: Columns $g_3, \tilde{f}_3$, and h_3 contain the values of g_3^{2,t_2,t_3}, $\tilde{f}_3^{2,t_2,t_3}$, and h_3^{2,t_2,t_3}, for $t_2, t_3 \in \{0,1,2\}$, respectively.

In Section 4.5, some computational results showed that r-PD-sets can be obtained with $\tilde{f}_p^{t_1,\ldots,t_s} \leq r \leq f_p^{t_1,\ldots,t_s}$. In fact, some of those computational results improve the values of $g_p^{t_1,\ldots,t_s}$. For example, a 73-PD-set of size 74 was found computationally for the $\mathbb{Z}_8$-linear code $H^{3,0,3}$. Note that $g_2^{3,0,3} = 31 < 73 < 84 = f_2^{3,0,3}$. Indeed, for all $\mathbb{Z}_8$-linear codes H^{3,t_2,t_3} with $t_2 \in \{0,1,2\}$ and

$t_3 \in \{0, 1, 2, 3\}$, the computational results given in Table 4.2 are better than the values of g_2^{3,t_2,t_3}. On the other hand, for the $\mathbb{Z}_4$-linear codes H^{4,t_2} with $t_2 \in \{3, 4, 5\}$ and H^{5,t_2} with $t_2 \in \{1, 3, 4, 5\}$, the values of g_p^{4,t_2} and g_p^{5,t_2} are better than the computational results given in Table 4.1, which are denoted by r_c. Table 5.4 shows the values of r_c and the values of $g_2^{t_1,t_2}$ and $f_2^{t_1,t_2}$ for the $\mathbb{Z}_4$-linear Hadamard codes H^{t_1,t_2} with $t_1 \in \{3, 4, 5\}$ and $t_2 \in \{0, 1, 2, 3, 4, 5\}$. The maximum between r_c and $g_2^{t_1,t_2}$ is shown in bold type, and the gray coloured cells indicate the case where the upper bound is reached.

t_1	t_2	r_c	$g_2^{t_1,t_2}$	$f_2^{t_1,t_2}$
3	0	**4**	–	4
	1	**7**	5	7
	2	**11**	7	11
	3	**18**	7	20
	4	**31**	-1	35
	5	**50**	-1	63
4	0	**15**	–	15
	1	**23**	**23**	24
	2	**38**	31	41
	3	62	**63**	72
	4	103	**127**	127
	5	172	**191**	226
5	0	**50**	–	50
	1	76	**77**	84
	2	**124**	119	145
	3	199	**239**	255
	4	321	**383**	454
	5	551	**575**	818

Table 5.4: Values r_c for which r_c-PD-sets for $\mathbb{Z}_4$-linear Hadamard codes H^{t_1,t_2} with $t_1 \in \{3, 4, 5\}$ and $t_2 \in \{0, 1, 2, 3, 4, 5\}$ were found in Chapter 4 using a non-deterministic method. The corresponding values of $g_2^{t_1,t_2}$ and $f_2^{t_1,t_2}$ are also given.

Chapter 6

Computation of a parity-check matrix for $\mathbb{Z}_{p^s}$-linear codes

In [HKC+94], it is shown that any quaternary linear code of type $(n; t_1, t_2)$ is permutation equivalent to a quaternary linear code with a generator matrix of the form

$$G = \begin{pmatrix} \mathrm{Id}_{t_1} & R & S \\ \mathbf{0} & 2\mathrm{Id}_{t_2} & 2T \end{pmatrix}, \tag{6.1}$$

where R and T are matrices over $\mathbb{Z}_4$ with all entries in $\{0, 1\} \subset \mathbb{Z}_4$, and S is a matrix over $\mathbb{Z}_4$. In this case, we say that G is in standard form. In the same paper, it is shown that if the generator matrix G of a quaternary linear code $\mathcal{C}$ is as in (6.1), then a parity-check matrix of $\mathcal{C}$ can be computed as follows:

$$H = \begin{pmatrix} -(S + RT)^T & T^T & \mathrm{Id}_{n-t_1-t_2} \\ 2R^T & 2\mathrm{Id}_{t_2} & \mathbf{0} \end{pmatrix}. \tag{6.2}$$

More generally, in [CS95], it is shown that if $\mathcal{C}$ is a linear code of type $(t_1, t_2, \ldots, t_s)$ with a generator matrix as in (2.10), then a parity-check matrix

for $\mathcal{C}$ is of the form

$$H = \begin{pmatrix} H_{1,1} & H_{2,1} & H_{3,1} & \cdots & H_{s,1} & \mathrm{Id}_{n-t} \\ pH_{1,2} & pH_{2,2} & pH_{3,2} & \cdots & p\mathrm{Id}_{t_s} & \mathbf{0} \\ \vdots & \vdots & \vdots & \iddots & \mathbf{0} & \mathbf{0} \\ p^{s-2}H_{1,s-1} & p^{s-2}H_{2,s-1} & p^{s-2}\mathrm{Id}_{t_3} & \cdots & \mathbf{0} & \mathbf{0} \\ p^{s-1}H_{1,s} & p^{s-1}\mathrm{Id}_{t_2} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{6.3}$$

where the column blocks have the same size as in (2.10).

In this chapter, we present the construction of a parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes in terms of a generator matrix in standard form. This can be seen as a generalization of matrix (6.2) for quaternary linear codes. The chapter is organized as follows. In Section 6.1, we describe two similar methods for obtaining a parity-check matrix for a $\mathbb{Z}_{p^s}$-additive code, one is based on the computation of block-minors of a matrix associated to the generator matrix in standard form, and the other is based on a recursive construction that uses previously computed matrices. In Section 6.2, we describe an algorithm for each method. We also carry out a performance comparison between an implementation of these algorithms in MAGMA, and the general function by MAGMA for computing the parity-check matrix of any linear code over a ring.

The faster method was included in a MAGMA package for $\mathbb{Z}_{p^s}$-additive codes [FTV23] (see Chapter 7). This work has been presented as a conference paper [FTVV24a] and as a journal paper [FTVV24b].

6.1 Computation of a parity-check matrix

In this section, we present two different approaches to construct a parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes from a generator matrix in standard form (see Theorems 74 and 77). First, we present some results on the computation of determinants for square matrices with a certain structure.

Let $A = (a_{r,s})_{1 \le r,s \le n}$ be a square $n \times n$ matrix. It is well-known that the

determinant of A can be computed as

$$|A| = \sum_{\sigma \in \mathcal{S}_n} \operatorname{sgn}(\sigma) \prod_{h=1}^{n} a_{h,\sigma(h)}, \tag{6.4}$$

where $\operatorname{sgn}(\sigma)$ is the parity of σ. If A is a square $n \times n$ matrix of the form

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ 1 & a_{2,2} & \cdots & a_{2,n-1} & a_{2,n} \\ 0 & 1 & \cdots & a_{3,n-1} & a_{3,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{n,n} \end{pmatrix}, \tag{6.5}$$

then we can improve the equation given in (6.4) for the determinant of A computing the additions of the products that do not consider any element under the diagonal with ones. Therefore, we just need to consider permutations $\sigma \in \mathcal{S}_n$ such that $\sigma(h) \geq h-1$ for all $h \in \{1, \ldots, n\}$. Moreover, we can avoid multiplying by one by considering only the products of elements $a_{i,j}$ with $j \geq i$. In order to write this more formally, we introduce the following definitions.

Definition 61. *Let $\hat{\mathcal{S}}_n = \{\sigma \in \mathcal{S}_n \,|\, \sigma(h) \geq h-1$ for all $h \in \{1, \ldots, n\}\}$, and $J_\sigma = \{h_1, \ldots, h_r\} = \{h \in \{1, \ldots, n\} \mid \sigma(h) \geq h\}$. Note that J_σ is not empty, since $\sigma(1) \geq 1$ for any $\sigma \in \hat{\mathcal{S}}_n$. We consider the elements in J_σ ordered, i.e., $h_1 < h_2 < \cdots < h_r$.*

Proposition 62. *Let A be a matrix as in (6.5). Then, the determinant of A is given by*

$$|A| = \sum_{\sigma \in \hat{\mathcal{S}}_n} \operatorname{sgn}(\sigma) \prod_{h \in J_\sigma} a_{h,\sigma(h)},$$

where $\hat{\mathcal{S}}_n$ and J_σ are as in Definition 61.

Proof. Straightforward from (6.4) and Definition 61. □

Example 63. *Consider the matrix A as in (6.5) with $n = 3$, that is,*

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 1 & a_{2,2} & a_{2,3} \\ 0 & 1 & a_{3,3} \end{pmatrix}.$$

In this case, we have that $\mathcal{S}_3 = \{Id, (1,2), (1,3), (2,3), (1,2,3), (1,3,2)\}$ and $\hat{\mathcal{S}}_3 = \{Id, (1,2), (2,3), (1,3,2)\}$. Then, $J_{Id} = \{1,2,3\}$, $J_{(1,2)} = \{1,3\}$, $J_{(2,3)} = \{1,2\}$, and $J_{(1,3,2)} = \{1\}$. Therefore,

$$|A| = a_{1,1}a_{2,2}a_{3,3} - a_{1,2}a_{3,3} - a_{1,1}a_{2,3} + a_{1,3}.$$

Now, we give an alternative expression for the computation of the determinant of a matrix A as in (6.5) by using the minors of the diagonal of A.

Definition 64. *Let $A = (a_{r,s})_{1 \le r,s \le n}$ be a square $n \times n$ matrix. The i-th minor of the diagonal of A of order j, denoted by O_j^i, is the determinant of the i-th submatrix of size j in the diagonal of A, that is,*

$$O_j^i = |(a_{r,s})_{i \le r,s \le i+j-1}|.$$

We consider that $O_0^i = 1$ for all $i \ge 1$. Note that $O_1^i = a_{i,i}$ and $O_n^1 = |A|$.

Proposition 65. *Let A be a matrix as in (6.5). Then, the determinant of A is given by*

$$|A| = O_n^1 = \sum_{k=1}^{n} (-1)^{k-1} a_{1,k} O_{n-k}^{k+1},$$

where O_j^i is the i-th minor of the diagonal of A of order j.

Proof. Using the Laplace expansion on the first column, the determinant $|A|$ can be calculated as follows:

$$|A| = a_{1,1}O_{n-1}^2 - |A'_{n-1}|,$$

where

$$A'_{n-i} = \begin{pmatrix} a_{1,i+1} & a_{1,i+2} & \cdots & a_{1,n-1} & a_{1,n} \\ 1 & a_{i+2,i+2} & \cdots & a_{i+2,n-1} & a_{i+2,n} \\ 0 & 1 & \cdots & a_{i+3,n-1} & a_{i+3,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{n,n} \end{pmatrix}.$$

We can repeat this process with A'_{n-1} obtaining $|A| = a_{1,1}O^2_{n-1} - a_{1,2}O^3_{n-2} + |A'_{n-2}|$, and so on so forth until we have

$$|A| = \sum_{k=1}^{n} (-1)^{k-1} a_{1,k} O^{k+1}_{n-k}.$$

□

Corollary 66. *Let A be a matrix as in (6.5). Then,*

$$O^i_j = \sum_{k=i}^{i+j-1} (-1)^{i-k} a_{i,k} O^{k+1}_{i+j-1-k}, \tag{6.6}$$

where O^i_j is the i-th minor of the diagonal of A of order j, for all $1 \leq i \leq n$ and $i \leq j \leq n$.

From now on, we consider block-matrices, that is, matrices whose entries are submatrices instead of scalars. Indeed, we first define the reduced associated matrix G^{RA} of a generator matrix G in standard form, which is a block-matrix.

Definition 67. *Let G be the generator matrix of a $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$ in standard form, that is, as in (2.10). The reduced associated matrix G^{RA} of G is the matrix*

$$G^{RA} = \begin{pmatrix} A_{1,2} & A_{1,3} & \cdots & A_{1,s} & A_{1,s+1} \\ \mathrm{Id}_{t_2} & A_{2,3} & \cdots & A_{2,s} & A_{2,s+1} \\ \mathbf{0} & \mathrm{Id}_{t_3} & \cdots & A_{3,s} & A_{3,s+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathrm{Id}_{t_s} & A_{s,s+1} \end{pmatrix}. \tag{6.7}$$

In general, a blockwise determinant of a square block-matrix, computed by performing multiplications and additions of blocks, is not well-defined due to the noncompatibility of the different dimensions of each block and the fact that the product of matrices is noncommutative. However, we propose a notion of determinant, called block-determinant, for any block-matrix of the form as in (6.7), by defining an analogous expression to the one given in Proposition 62. Since the block-submatrices $(A_{r,s+1})_{i\leq r,s\leq i+j-1}$ are also in the form of (6.7), we can also provide a notion of block-minors of the block-diagonal of G^{RA}, analogous to the minors O^i_j described in Definition 64. Then, we give an analogue of Proposition 65 to obtain another expression to compute these block-minors.

Definition 68. *Let A be a block-matrix as in (6.7). The i-th block-minor of the block-diagonal of A of order j, denoted also by O^i_j, is the block-determinant of the i-th submatrix of size j in the block-diagonal of A, that is,*

$$
\begin{aligned}
O^i_j = &|(A_{r,s+1})_{i\leq r,s\leq i+j-1}| = \\
&= \begin{vmatrix}
A_{i,i+1} & A_{i,i+2} & \cdots & A_{i,i+j-1} & A_{i,i+j} \\
\mathrm{Id}_{t_{i+1}} & A_{i+1,i+2} & \cdots & A_{i+1,i+j-1} & A_{i+1,i+j} \\
\mathbf{0} & \mathrm{Id}_{t_{i+2}} & \cdots & A_{i+2,i+j-1} & A_{i+2,i+j} \\
\vdots & \mathbf{0} & \ddots & \vdots & \vdots \\
\mathbf{0} & \dots & \mathbf{0} & \mathrm{Id}_{t_{i+j-1}} & A_{i+j-1,i+j}
\end{vmatrix} \\
&= \sum_{\sigma\in\hat{\mathcal{S}}_j} \mathrm{sgn}(\sigma) \prod_{h\in J_\sigma} A_{i+h-1,i+\sigma(h)}, \qquad (6.8)
\end{aligned}
$$

where $\hat{\mathcal{S}}_j$ and J_σ are as in Definition 61.

The following results are used to show that the products in (6.8) are well-defined.

Remark 69. *Let $\sigma\in\hat{\mathcal{S}}_j$ and $h\in\{1,\dots,j\}$. Then,*

1. *$\sigma(h)=h-1$ if $h\notin J_\sigma$,*

2. $\sigma^{-1}(h) \leq h + 1$.

Lemma 70. *Let* $\sigma \in \hat{\mathcal{S}}_j$ *and* $J_\sigma = \{h_1, \ldots, h_r\}$. *Then, for any* $k \in \{1, \ldots, r-1\}$, $h_{k+1} = \sigma(h_k) + 1$.

Proof. First, we see that $\sigma(h_k)+1 \in J_\sigma$. Assume $\sigma(h_k)+1 \notin J_\sigma$. By Remark 69-1, we have that $\sigma(\sigma(h_k)+1) = (\sigma(h_k)+1) - 1 = \sigma(h_k)$. Therefore, since σ is a one-to-one map, $\sigma(h_k) + 1 = h_k$, that is, $\sigma(h_k) = h_k - 1$ which is not possible because $h_k \in J_\sigma$. Therefore, $\sigma(h_k) + 1 \in J_\sigma$, and the result follows if we prove that $\{h_k + 1, \ldots, \sigma(h_k)\} \cap J_\sigma = \emptyset$.

Consider $i \in \{1, \ldots, h_k - 1\}$. By Remark 69-2, $\sigma^{-1}(i) \leq i + 1 \leq h_k$. If $\sigma^{-1}(i) = h_k$, then $\sigma(h_k) = i \leq h_k - 1$ which is not possible since $h_k \in J_\sigma$. Therefore, $\sigma^{-1}(i) \leq h_k - 1$. Since there are $h_k - 1$ different values of both $\sigma^{-1}(i)$ and $i \in \{1, \ldots, h_k - 1\}$, we have that $\sigma^{-1}(h_k) \notin \{1, \ldots, h_k - 1\}$. By Remark 69-2, $\sigma^{-1}(h_k) \leq h_k + 1$. Thus, there are only two possible values remaining for $\sigma^{-1}(h_k)$, say h_k or $h_k + 1$.

First, consider the case $\sigma(h_k) = h_k$. Since $h_k \in J_\sigma$, we have seen that $\sigma(h_k)+1 \in J_\sigma$ and hence $h_k+1 \in J_\sigma$. Then, since h_k and h_{k+1} are consecutive elements in J_σ, necessarily, $h_{k+1} = h_k + 1 = \sigma(h_k) + 1$ and the result holds. Finally, consider the case $\sigma(h_k + 1) = h_k$. In this case, $h_k + 1 \notin J_\sigma$. If $h_k + 1 = \sigma(h_k)$, then clearly $\{h_k + 1, \ldots, \sigma(h_k)\} \cap J_\sigma = \emptyset$ and we are done. If $h_k+1 \neq \sigma(h_k)$, by the same argument as before, $\sigma^{-1}(h_k+1) \notin \{1, \ldots, h_k-1\}$. By Remark 69-2, $\sigma^{-1}(h_k + 1) \leq h_k + 2$. Moreover, we have that $\sigma^{-1}(h_k + 1)$ cannot be h_k+1 or h_k, so $\sigma^{-1}(h_k+1) = h_k+2$. This implies that $h_k+2 \notin J_\sigma$. This argument can be applied recursively, obtaining $\{h_k+1, \ldots, \sigma(h_k)\} \cap J_\sigma = \emptyset$. □

Using Lemma 70, we see that all products in (6.8) are of the form

$$A_{i+h_k-1,i+\sigma(h_k)} A_{i+\sigma(h_k),i+\sigma(h_{k+1})},$$

for any $k \in \{1, \ldots, r-1\}$. Thus, O^i_j is a well-defined matrix of size $t_i \times z$, where z is the amount of columns that the matrices $A_{*,i+j}$ have, in this case t_{i+j}. We consider that $O^i_0 = Id$ for all $i \geq 1$. Note that $O^i_1 = A_{i,i+1}$.

Example 71. *Let G^{RA} be a block-matrix as in (6.7). By Definition 61, we have $\hat{\mathcal{S}}_2 = \{Id, (1,2)\}$, $J_{Id} = \{1,2\}$ and $J_{(1,2)} = \{1\}$. Then, the block-minor O_2^{s-1} can be computed as follows:*

$$\begin{aligned} O_2^{s-1} = & \begin{vmatrix} A_{s-1,s} & A_{s-1,s+1} \\ \mathrm{Id}_{t_s} & A_{s,s+1} \end{vmatrix} \\ = & A_{s-1,s}A_{s,s+1} - A_{s-1,s+1}. \end{aligned}$$

Clearly, $A_{s-1,s}A_{s,s+1}$ is a $t_{s-1} \times (n-t)$ matrix, where n is the total amount of columns of G and $t = \sum_{i=0}^{s} t_i$.

Similarly, taking into account the set $\hat{\mathcal{S}}_3$ and the corresponding sets of indices J_σ for $\sigma \in \hat{\mathcal{S}}_3$, given in Example 63, the block-minor O_3^{s-2} can be computed as follows:

$$\begin{aligned} O_3^{s-2} = & \begin{vmatrix} A_{s-2,s-1} & A_{s-2,s} & A_{s-2,s+1} \\ \mathrm{Id}_{t_{s-1}} & A_{s-1,s} & A_{s-1,s+1} \\ \mathbf{0} & \mathrm{Id}_{t_s} & A_{s,s+1} \end{vmatrix} \\ = & A_{s-2,s-1}A_{s-1,s}A_{s,s+1} - A_{s-2,s}A_{s,s+1} \\ & - A_{s-2,s-1}A_{s-1,s+1} + A_{s-2,s+1}. \end{aligned}$$

In order to compute the block-minor O_4^{s-3}, we consider the set of permutations $\hat{\mathcal{S}}_4 = \{Id, (3,4), (2,3), (2,4,3), (1,2),\ (1,2)(3,4), (1,3,2), (1,4,3,2)\}$. The corresponding sets of indices given in Definition 61 are:

$$\begin{array}{llll} J_{Id} = \{1,2,3,4\}, & J_{(3,4)} = \{1,2,3\}, & J_{(2,3)} = \{1,2,4\}, & J_{(2,4,3)} = \{1,2\}, \\ J_{(1,2)} = \{1,3,4\}, & J_{(1,2)(3,4)} = \{1,3\}, & J_{(1,3,2)} = \{1,4\}, & J_{(1,4,3,2)} = \{1\}. \end{array}$$

Then the block-minor O_4^{s-3} is

$$O_4^{s-3} = \begin{vmatrix} A_{s-3,s-2} & A_{s-3,s-1} & A_{s-3,s} & A_{s-3,s+1} \\ \mathrm{Id}_{t_{s-2}} & A_{s-2,s-1} & A_{s-2,s} & A_{s-2,s+1} \\ \mathbf{0} & \mathrm{Id}_{t_{s-1}} & A_{s-1,s} & A_{s-1,s+1} \\ \mathbf{0} & \mathbf{0} & \mathrm{Id}_{t_s} & A_{s,s+1} \end{vmatrix}$$

$$
\begin{aligned}
=&A_{s-3,s-2}A_{s-2,s-1}A_{s-1,s}A_{s,s+1}\\
&- A_{s-3,s-2}A_{s-2,s-1}A_{s-1,s+1}\\
&- A_{s-3,s-2}A_{s-2,s}A_{s,s+1} + A_{s-3,s-2}A_{s-2,s+1}\\
&- A_{s-3,s-1}A_{s-1,s}A_{s,s+1} + A_{s-3,s-1}A_{s-1,s+1}\\
&+ A_{s-3,s}A_{s,s+1} - A_{s-3,s+1}.
\end{aligned}
$$

Proposition 72. *Let A be a block-matrix as in (6.7). Then, the block-determinant of A is given by*

$$|A| = O_s^1 = \sum_{k=1}^{s}(-1)^{k-1}A_{1,k+1}O_{s-k}^{k+1},$$

where O_j^i is the i-th block-minor of the block-diagonal of A of order j.

Proof. It is easy to prove this statement by following an analogous argument to that used in the proof of Proposition 65. □

Corollary 73. *Let A be a block-matrix as in (6.7). Then,*

$$O_j^i = \sum_{k=i}^{i+j-1}(-1)^{i-k}A_{i,k+1}O_{i+j-1-k}^{k+1}, \tag{6.9}$$

where O_j^i is the i-th block-minor of the block-diagonal of A of order j, for all $1 \leq i \leq s$ and $i \leq j \leq s$.

Now, we give the result that allows us to compute a parity-check matrix using the block-minors of the reduced associated matrix G^{RA} of G.

Theorem 74. *Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of type $(n; t_1, \ldots, t_s)$ with a generator matrix G as in (2.10), and G^{RA} its reduced generator matrix. Then,*

the transpose of a parity-check matrix H of $\mathcal{C}$ is as follows:

$$\begin{pmatrix} H_{1,1} & pH_{1,2} & \cdots & p^{s-3}H_{1,s-2} & p^{s-2}H_{1,s-1} & p^{s-1}H_{1,s} \\ H_{2,1} & pH_{2,2} & \cdots & p^{s-3}H_{2,s-2} & p^{s-2}H_{2,s-1} & p^{s-1}\mathrm{Id}_{t_2} \\ H_{3,1} & pH_{3,2} & \cdots & p^{s-3}H_{3,s-2} & p^{s-2}\mathrm{Id}_{t_3} & \mathbf{0} \\ H_{4,1} & pH_{4,2} & \cdots & p^{s-3}\mathrm{Id}_{t_4} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \iddots & \vdots & \vdots & \vdots \\ H_{s,1} & p\mathrm{Id}_{t_s} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathrm{Id}_{n-t} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{6.10}$$

where $t = \sum_{i=1}^{s} t_i$,

$$H_{i,j} = (-1)^{s+2-i-j} O^i_{s+2-i-j}, \tag{6.11}$$

for all $1 \leq j \leq s$ *and* $1 \leq i \leq s+1-j$, *and* O^i_k *is the* i*-th block-minor of the block-diagonal of* G^{RA} *of order* k.

Proof. Let $\mathcal{C}'$ be the $\mathbb{Z}_{p^s}$-additive code generated by matrix H given in (6.10). First, we prove that $GH^T = (\mathbf{0})$ and hence $\mathcal{C}' \subseteq \mathcal{C}^\perp$. Denote G_s and H_s the matrices G and H, respectively, corresponding to the value s. We prove that $G_s H_s^T = (\mathbf{0})$ by induction on $s \geq 2$. For $s = 2$, we have

$$\begin{aligned} H_{2,1} &= -A_{2,3}, \\ H_{1,1} &= -A_{1,3} - A_{1,2}H_{2,1} = -A_{1,3} + A_{1,2}A_{2,3}, \\ H_{1,2} &= -A_{1,2}. \end{aligned}$$

Clearly,

$$G_2 H_2^T = \begin{pmatrix} \mathrm{Id}_{t_1} & A_{1,2} & A_{1,3} \\ \mathbf{0} & p\mathrm{Id}_{t_2} & pA_{2,3} \end{pmatrix} \cdot \begin{pmatrix} (A_{1,2}A_{2,3} - A_{1,3}) & -pA_{1,2} \\ -A_{2,3} & p\mathrm{Id}_{t_2} \\ \mathrm{Id}_{n-t_1-t_2} & \mathbf{0} \end{pmatrix} = (\mathbf{0}).$$

By induction hypothesis, we assume that $G_k H_k^T = (\mathbf{0})$ for every integer $k \leq s-1$. Let us decompose the matrices G_s and H_s in terms of G_{s-1} and

H_{s-1}:

$$G_s = \begin{pmatrix} & & & A_{1,s+1} \\ & G_{s-1} & & pA_{2,s+1} \\ & & & p^2A_{3,s+1} \\ & & & \vdots \\ \mathbf{0} & \cdots \quad \mathbf{0} & p^{s-1}\mathrm{Id}_{t_s} & p^{s-1}A_{s,s+1} \end{pmatrix},$$

$$H_s^T = \begin{pmatrix} H_{1,1} & & & \\ H_{2,1} & & & \\ H_{3,1} & & pH_{s-1}^T & \\ H_{4,1} & & & \\ \vdots & & & \\ H_{s,1} & & & \\ \mathrm{Id}_{n-t} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix}.$$

We can separate the product $G_sH_s^T$ in two parts. On the one hand, we have that

$$G_s \begin{pmatrix} pH_{s-1}^T \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} pG_{s-1}H_{s-1}^T \\ p^{s-1}\mathrm{Id}_{t_s} \cdot p\mathrm{Id}_{t_s} \end{pmatrix} = (\mathbf{0})$$

by the induction hypothesis and the structure of H_{s-1}^T. On the other hand, we have to prove that

$$G_s \begin{pmatrix} H_{1,1} \\ H_{2,1} \\ H_{3,1} \\ H_{4,1} \\ \vdots \\ H_{s,1} \\ \mathrm{Id}_{n-t} \end{pmatrix} = G_sH' = (\mathbf{0}),$$

where H' is the matrix having the first $n-t$ columns of H_s^T. Note that the i-th block-row of G_s is of the form

$$(G_s)_i = \begin{pmatrix} \mathbf{0} \cdots \mathbf{0} & p^{i-1}\mathrm{Id}_{t_i} & p^{i-1}A_{i,i+1} & \cdots & p^{i-1}A_{i,s+1} \end{pmatrix}$$

for all $1 \leq i \leq s$. Then,

$$\begin{aligned}
&(G_s)_i H' = \\
&= p^{i-1}\mathrm{Id}_{t_i} H_{i,1} + \sum_{j=1}^{s-i} p^{i-1} A_{i,i+j} H_{i+j,1} + p^{i-1} A_{i,s+1} \\
&= p^{i-1}(-1)^{s+1-i} O^i_{s+1-i} \\
&\quad + p^{i-1} \sum_{j=1}^{s-i} A_{i,i+j}(-1)^{s+1-i-j} O^{i+j}_{s+1-i-j} + p^{i-1} A_{i,s+1} \\
&= p^{i-1} \left[(-1)^{s+1-i} O^i_{s+1-i} + \sum_{j=1}^{s+1-i} A_{i,i+j}(-1)^{s+1-i-j} O^{i+j}_{s+1-i-j} \right] \\
&= p^{i-1}(-1)^{s+1-i} \left[O^i_{s+1-i} - \sum_{j=1}^{s+1-i} A_{i,i+j}(-1)^{j-1} O^{i+j}_{s+1-i-j} \right],
\end{aligned}$$

where the second equality is given by (6.11) and the third one by considering that, by definition, $O_0^{s+1} = Id$. Now, using Corollary 73, it is easy to see that

$$O^i_{s+1-i} = \sum_{j=1}^{s+1-i} A_{i,i+j}(-1)^{j-1} O^{i+j}_{s+1-i-j}.$$

Thus $(G_s)_i H' = (\mathbf{0})$ for all $1 \leq i \leq s$ and we conclude that $G_s H_s^T = (\mathbf{0})$.

We have proven that $\mathcal{C}' \subseteq \mathcal{C}^\perp$. For the other inclusion, we consider an arbitrary codeword $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathcal{C}^\perp$ and we have to see that it also belongs to $\mathcal{C}'$. The following argument was adapted from [Wan03, Proposition 1.2]. The first $n-t$ rows of H have the identity matrix in the last $n-t$ columns. Therefore, we can add a linear combination of these $n-t$ first rows to $\mathbf{c}$ to obtain a new codeword of $\mathcal{C}^\perp$ of the form

$$\mathbf{c}^{(1)} = (c_1, c_2, \ldots, c_t, 0, \ldots, 0).$$

Since $\mathbf{c}^{(1)} \in \mathcal{C}^\perp$ is orthogonal to the last t_s rows of G_s, we obtain

$$\begin{aligned}
\mathbf{c}^{(1)}((G_s)_s)^T &= \mathbf{c}^{(1)}(\mathbf{0} \ \cdots \ \mathbf{0} \quad p^{s-1}\mathrm{Id}_{t_s} \quad p^{s-1}A_{s,s+1})^T \\
&= p^{s-1}(c_{t_1+\cdots+t_{s-1}+1}, \ldots, c_t) = \mathbf{0}.
\end{aligned}$$

This means that the components $c_{t_1+\cdots+t_{s-1}+1}, \ldots, c_t$ are all multiples of p. The next t_s rows of H, which are

$$\left(\begin{array}{ccccccc} pH_{1,2} & pH_{2,2} & \ldots & pH_{s-1,2} & p\mathrm{Id}_{t_s} & \mathbf{0} \end{array} \right), \tag{6.12}$$

have zero entries in the last $n-t$ columns and $p\mathrm{Id}_{t_s}$ in the previous t_s columns. Therefore, we can add a linear combination of the rows in (6.12) to obtain a new codeword of $\mathcal{C}^\perp$ of the form

$$\mathbf{c}^{(2)} = (c_1, c_2, \ldots, c_{t_1+\cdots+t_{s-1}}, 0, \ldots, 0).$$

Since $\mathbf{c}^{(2)} \in \mathcal{C}^\perp$ is orthogonal to the second to last t_{s-1} rows of G, we obtain

$$\mathbf{c}^{(2)}((G_s)_{s-1})^T = p^{s-2}(c_{t_1+\cdots+t_{s-2}+1}, \ldots, c_{t_1+\cdots+t_{s-1}}) = \mathbf{0}.$$

This means that the components $c_{t_1+\cdots+t_{s-2}+1}, \ldots, c_{t_1+\cdots+t_{s-1}}$ are multiples of p^2. The same argument can be applied iteratively until we obtain the codeword of $\mathcal{C}^\perp$,

$$\mathbf{c}^{(s)} = (c_1, \ldots, c_{t_1}, 0, \ldots, 0).$$

Since $\mathbf{c}^{(s)} \in \mathcal{C}^\perp$ is orthogonal to the first t_1 rows of G, we obtain

$$\mathbf{c}^{(s)}((G_s)_1)^T = (c_1, \ldots, c_{t_1}) = \mathbf{0}.$$

Hence $\mathbf{c}^{(s)} = \mathbf{0}$ and $\mathbf{c} \in \mathcal{C}'$, since it can be obtained by a linear combination of the rows of H. Consequently, H is a generator and a parity-check matrix for $\mathcal{C}^\perp$ and $\mathcal{C}$, respectively. □

Example 75. *Let $p = 2$ and $s = 2$. Let G be the generator matrix in standard form of a $\mathbb{Z}_4$-additive code $\mathcal{C}$ of type $(n; t_1, t_2)$ and G^{RA} its reduced associated matrix:*

$$G = \left(\begin{array}{ccc} \mathrm{Id}_{t_1} & A_{1,2} & A_{1,3} \\ \mathbf{0} & 2\mathrm{Id}_{t_2} & 2A_{2,3} \end{array} \right), \; G^{RA} = \left(\begin{array}{cc} A_{1,2} & A_{1,3} \\ \mathrm{Id}_{t_2} & A_{2,3} \end{array} \right).$$

Then, by Theorem 74, the transpose of a generator matrix H of $\mathcal{C}^\perp$ can be constructed as follows:

$$H^T = \begin{pmatrix} A_{1,2}A_{2,3} - A_{1,3} & -2A_{1,2} \\ -A_{2,3} & 2\mathrm{Id}_{t_2} \\ \mathrm{Id}_{n-t_1-t_2} & \mathbf{0} \end{pmatrix},$$

since $H_{1,1} = O_2^1 = A_{1,2}A_{2,3} - A_{1,3}$, $H_{2,1} = -O_1^2 = -A_{2,3}$, and $H_{1,2} = -O_1^1 = -A_{1,2}$. Note that this parity-check matrix H of $\mathcal{C}$ generates the same code as the matrix (6.2). Indeed, both matrices are equal if we consider $-A_{2,3}$ instead of $A_{2,3}$. Note that, in both cases, G generates the same code $\mathcal{C}$.

Example 76. *Let $p = 2$ and $s = 3$. Let G be the generator matrix in standard form of a $\mathbb{Z}_8$-additive code $\mathcal{C}$ of type $(n; t_1, t_2, t_3)$ and G^{RA} its reduced associated matrix:*

$$G = \begin{pmatrix} \mathrm{Id}_{t_1} & A_{1,2} & A_{1,3} & A_{1,4} \\ \mathbf{0} & 2\mathrm{Id}_{t_2} & 2A_{2,3} & 2A_{2,4} \\ \mathbf{0} & \mathbf{0} & 4\mathrm{Id}_{t_3} & 4A_{3,4} \end{pmatrix},$$

$$G^{RA} = \begin{pmatrix} A_{1,2} & A_{1,3} & A_{1,4} \\ \mathrm{Id}_{t_2} & A_{2,3} & A_{2,4} \\ \mathbf{0} & \mathrm{Id}_{t_3} & A_{3,4} \end{pmatrix}.$$

Then, the transpose of a generator matrix H of $\mathcal{C}^\perp$ can be constructed as follows:

$$H^T = \begin{pmatrix} -(A_{1,2}A_{2,3}A_{3,4} + A_{1,4} - A_{1,2}A_{2,4} - A_{1,3}A_{3,4}) & 2\,(A_{1,2}A_{2,3} - A_{1,3}) & -4A_{1,2} \\ A_{2,3}A_{3,4} - A_{2,4} & -2A_{2,3} & 4\mathrm{Id}_{t_2} \\ -A_{3,4} & 2\mathrm{Id}_{t_3} & \mathbf{0} \\ \mathrm{Id}_{n-t_1-t_2-t_3} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

by Theorem 74 and Corollary 73. For example, we have that $H_{1,1} = -O_3^1 = A_{1,2}O_2^2 - A_{1,3}O_1^3 + A_{1,4}O_0^4 = A_{1,2}(A_{2,3}A_{3,4} - A_{2,4}) - A_{1,3}A_{3,4} + A_{1,4}$.

The computation of a parity-check matrix by using Theorem 74 requires the reckoning of many minors. The computation of these minors, $O^i_{s+2-i-j}$,

is carried out using Corollary 73, so it requires the computation of j different minors of lower order. In this case, we assume that we compute each one of the blocks of the parity-check matrix $H_{i,j}$ independently. However, now, we show that following an appropriate order in the computation of the different matrices $H_{i,j}$, we are able to obtain an expression to compute $O_{s+2-i-j}$, where all the minors of lower order in (6.9) have already been computed in a previous step. In fact, we can obtain a similar expression, which directly relates $H_{i,j}$ with all others $H_{i,k}$ such that $k \geq j$. This is shown in Theorem 77.

Theorem 77. *Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of type $(n; t_1, \ldots, t_s)$ with a generator matrix G as in (2.10). Then, the blocks of the matrix (6.10) given in Theorem 74, which is the transpose of a parity-check matrix H of $\mathcal{C}$, can be calculated as follows:*

$$H_{i,j} = -\left(A_{i,s-j+2} + \sum_{k=i+1}^{s-j+1} A_{i,k} H_{k,j}\right) \tag{6.13}$$

for all $1 \leq j \leq s$ and $1 \leq i < s-j+1$. Note that $H_{s-j+1,j} = -A_{s-j+1,s-j+2}$.

Proof. We prove this statement by seeing that the matrix computed by using (6.13) is the same as the one in (6.10). To achieve that, we show that $H_{i,j} = \hat{H}_{i,j}$ for all $1 \leq j \leq s$ and $1 \leq i \leq s-j+1$, where $\hat{H}_{i,j}$ is as in (6.11), that is,

$$\begin{aligned} H_{i,j} &= -\left(A_{i,s-j+2} + \sum_{k=i+1}^{s-j+1} A_{i,k} H_{k,j}\right) \\ &= (-1)^{s+2-i-j} O^i_{s+2-i-j} = \hat{H}_{i,j}. \end{aligned} \tag{6.14}$$

We prove this by induction on i for any $j \in \{1, \ldots, s\}$. For the case $i = s+1-j$, we have that

$$H_{s+1-j,j} = -\left(A_{s+1-j,s-j+2}\right) = (-1) O_1^{s+1-j} = \hat{H}_{s+1-j,j}.$$

By induction hypothesis, we assume that (6.14) is true for $i \leq s-j+1$ and

we want to see that it is true for $i-1$, i.e., $H_{i-1,j} = \hat{H}_{i-1,j}$. We have that

$$\begin{aligned}
\hat{H}_{i-1,j} &= (-1)^{s+2-j-(i-1)} O^{i-1}_{s+2-j-(i-1)} \\
&= (-1)^{s+2-j-(i-1)} \sum_{k=i-1}^{s+1-j} A_{i-1,k+1}(-1)^{i-1-k} O^{k+1}_{s+1-j-k} \\
&= (-1)^{s+2-j-(i-1)} \sum_{k=i}^{s+2-j} A_{i-1,k}(-1)^{i-k} O^{k}_{s+2-j-k} \\
&= -\sum_{k=i}^{s+2-j} A_{i-1,k}(-1)^{s+2-j-k} O^{k}_{s+2-j-k} \\
&= -\left(A_{i-1,s+2-j} + \sum_{k=i}^{s+1-j} A_{i-1,k}(-1)^{s+2-j-k} O^{k}_{s+2-j-k} \right) \\
&= -\left(A_{i-1,s+2-j} + \sum_{k=i}^{s+1-j} A_{i-1,k} H_{k,j} \right) \\
&= H_{i-1,j}.
\end{aligned}$$

The first equality is by definition, the second is by Corollary 73, the third is a rearrangement of the indices, the sixth is by the induction hypothesis, and the last one is by definition. □

Recall that $\mathbb{Z}_{p^s}$ is a chain ring, so $\mathbb{Z}_{p^s}$-additive codes are included in the family of linear codes over chain rings. Let $\mathcal{C}$ be a linear code over a finite commutative chain ring R with maximal ideal $\langle\gamma\rangle$ and nilpotency index s. It is well-known that $\mathcal{C}$ is permutation equivalent to a code generated by a matrix in standard form as in (2.10) [NS00], just by replacing p by γ. Therefore, all the results given in this chapter can be applied, exactly in the same way, to linear codes over chain rings since we only use the general properties of rings and the form of the generator matrix in standard form.

6.2 Performance comparison

In this section, we describe two algorithms that implement the computation of a parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes (or, more generally, linear codes over a chain ring), from a generator matrix in standard form. They are based on Theorems 74 and Theorem 77, respectively. First, we show a naive implementation, which is based on computing each submatrix $H_{i,j}$ in (6.10) by using the expression given in (6.11) and Corollary 73. Afterwards, we present an iterative construction that reduces the calculations considerably by using the expression given in (6.13). Then, the performance of these algorithms implemented in MAGMA is compared with the performance if we use the current available function in MAGMA for codes over finite rings in general. A time computation and time complexity analysis are also given.

6.2.1 Algorithms description

The first procedure corresponds to the one presented in Theorem 74, considering that each one of the blocks $H_{i,j}$ in (6.10) is computed independently, by using the expression given in (6.11), that is, $H_{i,j} = (-1)^{s+2-i-j} O^i_{s+2-i-j}$, and Corollary 73 to compute each one of the block-minors $O^i_{s+2-i-j}$ from the computation of different minors of lower order. This implementation is shown in Algorithm 1.

With the result given by Theorem 77, we can easily define a new implementation, which reduces the number of operations compared to Algorithm 1. In particular, for each block-column j, we can compute each $H_{i,j}$ starting from $H_{s-j+1,j} = -A_{s-j+1,s-j+2}$ and using (6.13) to obtain $H_{i,j}$ for $1 \leq i < s - j + 1$, in decreasing order. Since all $H_{k,j}$, for $k \geq i$, have been already determined when $H_{i,j}$ is computed, no additional operations are performed apart from the sums and products of matrices represented in (6.13). This new implementation is shown in Algorithm 2.

Algorithm 1 Parity-check matrix in standard form. Minors construction.

Require: A $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$ of type $(n; t_1, \ldots, t_s)$.

Compute a generator matrix G in standard form of $\mathcal{C}$.
Define $t := t_1 + \cdots + t_s$.
Define a zero matrix H^T with n rows and $n - t$ columns.
Define $numCol := 1$.
for $j := 1, \ldots, s$ **do**
  Define $numRow := 1$.
  for $i := 1, \ldots, s - j + 1$ **do**
    Compute $O^i_{s+2-i-j}$ using the block-matrix G and Corollary 73.
    Define $H_{i,j} := (-1)^{s+2-i-j} O^i_{s+2-i-j}$.
    Insert $p^{j-1} H_{i,j}$ at position $(numRow, numCol)$ in H^T.
    $numRow := numRow + t_i$.
  end for
  if $j = 1$ **then**
    Insert Id_{n-t} at position $(numRow, numCol)$ in H^T.
    $numCol := numCol + n - t$.
  else
    Insert $p^{j-1}\mathrm{Id}_{t_{s-j+2}}$ at position $(numRow, numCol)$ in H^T.
    $numCol := numCol + t_{s-j+2}$.
  end if
end for
return The parity-check matrix H.

6.2.2 Performance comparison

In this subsection, we compare three different methods for computing the parity-check matrix of a $\mathbb{Z}_{p^s}$-additive code. Two of them are the different versions of the method that can be obtained from Theorem 74 and Theorem 77, which are described in Algorithm 1 and Algorithm 2, respectively. The third method consists on using the Magma function `ParityCheckMatrix()`, included in the current official distribution [BCFS19], which computes a parity-check matrix for a linear code defined over any finite ring. First, we make an experimental comparison by using Magma and present the results through some graphs, thereafter we calculate the complexity of the methods introduced in this chapter.

Computation time analysis

In order to compare the performance of Algorithms 1 and 2 and the MAGMA function, we consider a random $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$ of type $(n; \ell, \ldots, \ell)$, that is, with $t_i = \ell$ for any $1 \leq i \leq s$. We study the effect of changing the parameters n, s, and ℓ on the three different methods. The first method (Algorithm 1), which computes each minor independently, is labeled as *Minors*. The second method (Algorithm 2), which computes the minors iteratively, is labeled as *Iterative*. Finally, the third method, which uses the MAGMA function `ParityCheckMatrix()`, is labeled as *Generic*.

Figure 6.1 shows the computation times of all methods for random $\mathbb{Z}_{3^s}$-additive codes of type $(1000; 2, \ldots, 2)$, where s takes values between 2 and 16. Similarly, Figure 6.2 shows the computation times of all methods for random $\mathbb{Z}_{3^4}$-additive codes of type $(1000; \ell, \ldots, \ell)$, where ℓ takes values between 2 and 20. Finally, Figures 6.3 and 6.4 show the computation times of all methods for random $\mathbb{Z}_{3^{10}}$-additive codes of type $(n; 2, \ldots, 2)$, where $n \in \{2^i \cdot 100 \mid 0 \leq i \leq 8\}$.

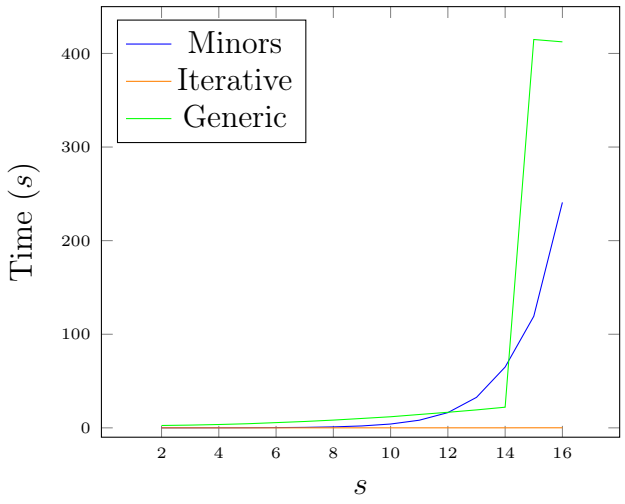

Figure 6.1: Codes of type $(1000; 2, \ldots, 2)$, with $p = 3$ and $2 \leq s \leq 16$.

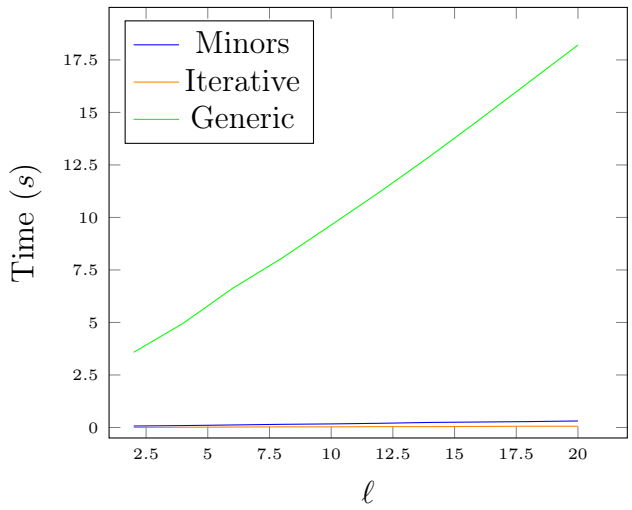

Figure 6.2: Codes of type $(1000; \ell, \ldots, \ell)$, with $p = 3$, $s = 4$ and $2 \leq \ell \leq 20$.

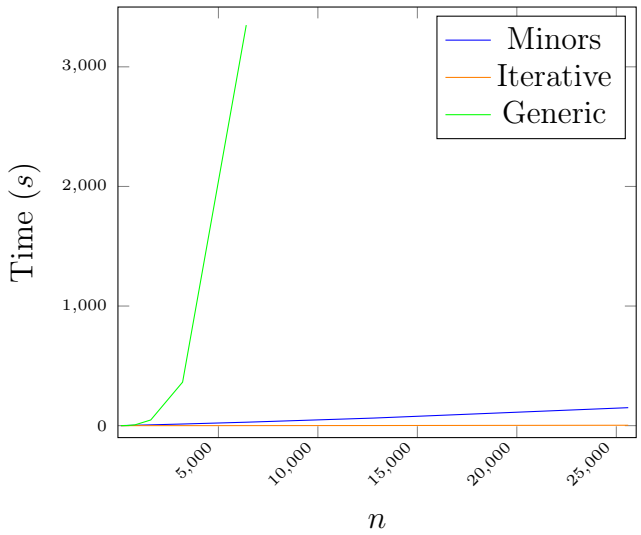

Figure 6.3: Codes of type $(n; 2, \ldots, 2)$, with $p = 3$, $s = 10$ and $100 \leq n \leq 25600$.

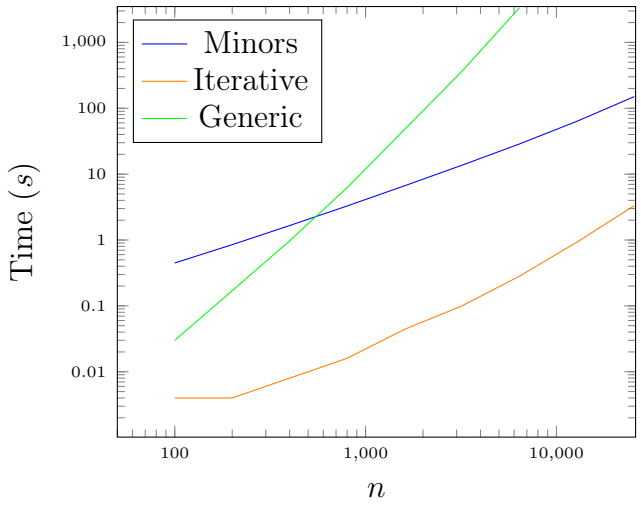

Figure 6.4: Codes of type $(n; 2, \ldots, 2)$, with $p = 3$, $s = 10$ and $100 \leq n \leq 25600$ (logarithmic scale).

Time complexity analysis

Let us consider Algorithms 1 and 2, which are based on the following expressions, respectively:

$$\hat{H}_{i,j} = (-1)^{s+2-i-j} O^i_{s+2-i-j} \quad \text{and} \tag{6.15}$$

$$H_{i,j} = -\left(A_{i,s-j+2} + \sum_{k=i+1}^{s-j+1} A_{i,k} H_{k,j} \right), \tag{6.16}$$

for $1 \leq j \leq s$ and $1 \leq i \leq s - j + 1$. Note that (6.15) and (6.16) coincide with the equations given in Theorem 74 and Theorem 77, respectively. In the first case, we denote the submatrices as $\hat{H}_{i,j}$ instead of $H_{i,j}$ in order to distinguish between both methods.

For simplicity, let us assume that $\mathcal{C}$ is a $\mathbb{Z}_{p^s}$-additive code of type $(n; \ell, \ldots, \ell)$. Then, $t = s\ell$, $H_{i,1}$ is a $\ell \times (n - t)$ matrix for any $1 \leq i \leq s$, and $H_{i,j}$ is a $\ell \times \ell$ matrix for any $2 \leq j \leq s$ and $1 \leq i \leq s-j+1$. We denote by $\hat{T}_{i,j}(s, n, \ell)$ and $T_{i,j}(s, n, \ell)$ the runtime needed to compute $\hat{H}_{i,j}$ and $H_{i,j}$, respectively. We also denote by $S(a, b)$ the computation time of the addition of two $a \times b$ matrices over $\mathbb{Z}_{p^s}$ and $P(a, b, c)$ the computation time of the product of an

$a \times b$ matrix by a $b \times c$ matrix.

With the aim of computing $\hat{H}_{i,j}$, we first estimate the complexity of determining any block-minor O^i_j. Due to the structure of G^{RA}, by Corollary 73, we can compute O^i_j by calculating block-determinants of one dimension less and the same structure. Then, by induction, it easy to show that the runtime needed to compute O^i_j is $(2^{s-i}-1)\left(P(\ell,\ell,n-t)+S(\ell,n-t)\right)$ for $j = s+1-i$ and $(2^{j-1}-1)\left(P(\ell,\ell,\ell)+S(\ell,\ell)\right)$ for $j < s+1-i$. Thus, by using (6.15), we have that

$$\begin{aligned}\hat{T}_{i,1}(s,n,\ell) &= (2^{s-i}-1)\left(P(\ell,\ell,n-t)+S(\ell,n-t)\right),\\ \hat{T}_{i,j}(s,n,\ell) &= (2^{s+1-i-j}-1)\left(P(\ell,\ell,\ell)+S(\ell,\ell)\right) \text{ for } j>1. \end{aligned} \tag{6.17}$$

In order to obtain $H_{i,j}$, we need to compute $H_{i',j}$ for all $i \le i' \le s-j+1$. Thus, for each $1 \le j \le s$, we start with $H_{s-j+1,j} = -A_{s-j+1,s-j+2}$ and then compute the sequence of matrices $H_{s-j,j}, H_{s-j-1,j}, \ldots, H_{1,j}$ by using (6.16). In this case, we have that

$$\begin{aligned}T_{i,1}(s,n,\ell) &= (s-i)P(\ell,\ell,n-t)+(S(\ell,n-t)),\\ T_{i,j}(s,n,\ell) &= (s-j-i+1)(P(\ell,\ell,\ell)+S(\ell,\ell)) \text{ for } j>1. \end{aligned} \tag{6.18}$$

Therefore, the total runtime of computing the parity-check matrix of $\mathcal{C}$ by using Algorithms 1 and 2 is given by the following results:

Proposition 78. *Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of type $(n;\ell,\ldots,\ell)$, and $t = s\ell$. The total runtime of computing the parity-check matrix of $\mathcal{C}$ by using Algorithm 1 is*

$$\begin{aligned}\hat{T}(s,n,\ell) = (2^s-1-s)\left(P(\ell,\ell,n-t)+S(\ell,n-t)\right)+\\ +\left(2^s-1-\frac{s^2}{2}-\frac{s}{2}\right)\left(P(\ell,\ell,\ell)+S(\ell,\ell)\right).\end{aligned}$$

Proof. We have $\hat{T}(s,n,\ell) = \sum_{i=1}^{s}\hat{T}_{i,1}(s,n,\ell)+\sum_{j=2}^{s}\sum_{i=1}^{s-j+1}\hat{T}_{i,j}(s,n,\ell)$. Recall that $\sum_{k=0}^{n}2^k = 2^{n+1}-1$. By using (6.17), since $\sum_{i=1}^{s}(2^{s-i}-1) = 2^s-1-s$ and $\sum_{j=2}^{s}\sum_{i=1}^{s-j+1}(2^{s+1-i-j}-1) = \sum_{j=2}^{s}(2^{s+1-j}-s+j-2) = 2^s-1-\frac{s^2}{2}-\frac{s}{2}$,

the result follows. □

Proposition 79. *Let $\mathcal{C}$ be a $\mathbb{Z}_{p^s}$-additive code of type $(n;\ell,\dots,\ell)$, and $t = s\ell$. The total runime of computing the parity-check matrix of $\mathcal{C}$ by using Algorithm 2 is*

$$\begin{aligned} T(s,n,\ell) = &\frac{s(s-1)}{2}(P(\ell,\ell,n-t)+S(\ell,n-t)) \\ &+\frac{1}{6}(s^3-3s^2+2s)(P(\ell,\ell,\ell)+S(\ell,\ell)). \end{aligned}$$

Proof. We have $T(s,n,\ell) = \sum_{i=1}^{s} T_{i,1}(s,n,\ell) + \sum_{j=2}^{s}\sum_{i=1}^{s-j+1} T_{i,j}(s,n,\ell)$. We also have that $\sum_{i=1}^{s}(s-i) = \sum_{j=0}^{s-1} j = s(s-1)/2$ and

$$\begin{aligned} &\sum_{j=2}^{s}\sum_{i=1}^{s-j+1}(s-j-i+1) = \sum_{j=2}^{s}(s-j)(s-j+1)/2 \\ &=\frac{1}{2}(\sum_{j=2}^{s}s^2+\sum_{j=2}^{s}j^2-\sum_{j=2}^{s}j(2s+1)) \\ &=\frac{1}{2}(s^3-s+(2s^3+3s^2+s-6)/6-(2s^3+3s^2-3s-2)/2 \\ &=\frac{1}{6}(s^3-3s^2+2s). \end{aligned}$$

Finally, by (6.18), the result follows. □

Regarding the asymptotic complexity of the algorithms, since $S(a,b)$ is $O(ab)$ and $P(a,b,c)$ is $O(abc)$, we obtain $S(\ell,n-t)+P(\ell,\ell,n-t) = O((n-t)\ell^2)$ and $S(\ell,\ell)+P(\ell,\ell,\ell) = O(\ell^3)$. Therefore,

$$\begin{aligned} \hat{T}(s,n,\ell) &= O(2^s\ell^2\,(n+s\ell)) \\ T(s,n,\ell) &= O(s^2\ell^2 n). \end{aligned}$$

If we only consider the variable n, we obtain that both algorithms are $O(n)$. Otherwise, if we take into account the variable s, we can see that while the first algorithm is exponential, the second one has square polynomial complexity, which fits the data shown in Figures 6.1, 6.2, 6.3, and 6.4.

Algorithm 2 Parity-check matrix in standard form. Iterative construction.

Require: A $\mathbb{Z}_{p^s}$-additive code $\mathcal{C}$ of type $(n; t_1, \ldots, t_s)$.

Compute a generator matrix G in standard form of $\mathcal{C}$.
Define $t := t_1 + \cdots + t_s$.
Define a zero matrix H^T with n rows and $n - t$ columns.
Define $numCol := 1$
for $j := 1, \ldots, s$ **do**
  Define $numRow := t_1 + \cdots + t_{s-j+1} + 1$.
  if $j = 1$ **then**
    Insert Id_{n-t} at position $(numRow, numCol)$ in H^T.
  else
    Insert $p^{j-1}\mathrm{Id}_{t_{s-j+2}}$ at position $(numRow, numCol)$ in H^T.
  end if
  $numRow := numRow - t_{s-j+1}$
  Define $H_{s-j+1,j} := -A_{s-j+1,s-j+2}$.
  Insert $p^{j-1}H_{s-j+1,j}$ at position $(numRow, numCol)$ in H^T.
  for $i := s - j, \ldots, 1$ by -1 **do**
    $numRow := numRow - t_i$.
    Compute $H_{i,j}$ using (6.13) and previously computed matrices $H_{k,j}$, for $k := i + 1, \ldots, s - j + 1$.
    Insert $p^{j-1}H_{i,j}$ at position $(numRow, numCol)$ in H^T.
  end for
  if $j = 1$ **then**
    $numCol := numCol + n - t$.
  else
    $numCol := numCol + t_{s-j+2}$.
  end if
end for
return The parity-check matrix H.

Chapter 7

Implementation of a MAGMA package

In this chapter, we describe all the software produced during the preparation of this thesis. In particular, we have contributed to the development of a package [FTV23] for the MAGMA Computational Algebra System [BCFS19] that provides new functionality to deal with linear codes over the ring $\mathbb{Z}_{p^s}$, with p prime. Moreover, we have also contributed with some functions to another MAGMA package [BFG+22a], which deals with $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, that is, linear codes with some coordinates over $\mathbb{Z}_2$ and some over $\mathbb{Z}_4$.

The functions contained in this chapter serve two main objectives concerning the main topic of the thesis: on the one hand, they have helped in the research leading to the results presented in previous chapters and, on the other, they include the implementation of the results themselves, which may help in future research regarding these codes. Additionally, we have developed core functions that address most of the essential concepts of $\mathbb{Z}_{p^s}$-additive and $\mathbb{Z}_{p^s}$-linear codes. Regarding the package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, particular attention was given to the computation of the minimum (Lee) weight of a code.

In Section 7.1, we give a brief introduction to the MAGMA system, the package for $\mathbb{Z}_{p^s}$-additive codes and the package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. In Section 7.2, we describe the different functions that have been implemented

in the package for $\mathbb{Z}_{p^s}$-additive codes, which correspond to the results given in the previous chapters. In Section 7.3, we give a brief introduction to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, explain the adaptation of the Brouwer-Zimmermann methods for computing the minimum Lee weight of these codes and describe the functions that have been implemented in the package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, regarding the computation of the minimum Lee weight.

7.1 MAGMA package implementation

MAGMA is a computer algebra system designed to work with structures and concepts from several areas of mathematics such as algebra, number theory, algebraic geometry and combinatorics. MAGMA provides all the necessary features to conduct research on coding theory, since it provides an extensive support for fields, rings, vectors spaces, modules, finite groups and all the related algebraic structures. In particular, it contains packages to deal with linear codes defined over finite fields and finite rings.

Additionally, the Combinatoric, Coding and Security Group (CCSG) of the Universitat Autònoma de Barcelona has developed different packages that extend the functionality of MAGMA by providing features that are specially designed for some families of codes over rings. This is the case of the package for linear codes over the ring $\mathbb{Z}_4$, that is, $\mathbb{Z}_4$-additive codes, which is already included in the official distribution of MAGMA [BPPV16]. This package includes functions for an efficient computation of the rank and dimension of the kernel for any linear code over $\mathbb{Z}_4$, functions for computing the permutation automorphism group of $\mathbb{Z}_4$-additive Hadamard codes, methods for computing the minimum weight of a linear code over $\mathbb{Z}_4$, and several decoding algorithms for linear codes over $\mathbb{Z}_4$. Moreover, a package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes was also published by the CCSG group (version 4.0) [BFG+17], providing a specific structure for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes and generalizing most of the features previously defined for codes over $\mathbb{Z}_4$.

One of the main objectives included in the project leading to this thesis was to contribute in the development of a new MAGMA package for linear

codes over $\mathbb{Z}_{p^s}$, which at the same time would provide support for research in this topic. From the very beginning of the project, we started generalizing the most basic features already defined for codes over $\mathbb{Z}_4$, such as several functions related to the Gray map (including Carlet's generalized Gray map), functions for obtaining a generator matrix in standard form of a linear code over $\mathbb{Z}_{p^s}$, and functions that construct different families of codes (Hadamard, simplex, MacDonald). In the process of achieving the results presented in this thesis, we were simultaneously developing MAGMA functions to implement them. That is, the results in Chapter 3 allowed us to implement functions that give an information set and a systematic encoding for any linear code over $\mathbb{Z}_{p^s}$, as well as their corresponding information space. The results in Chapters 4 and 5 allowed us to introduce functions for obtaining r-PD-sets for $\mathbb{Z}_{p^s}$-linear GH codes, and the permutation decoding method was generalized to any $\mathbb{Z}_{p^s}$-linear code in order to benefit from these new r-PD-sets. Finally, the most efficient algorithm for obtaining a parity-check matrix of any $\mathbb{Z}_{p^s}$-additive code, given in Chapter 6, was also implemented. In particular, a function that computes more efficiently the dual of any $\mathbb{Z}_{p^s}$-additive code is included. All these functions are described in more detail in Section 7.2, along with examples and some relevant performance comparisons between some methods, and are included in the 1.0 version of the new Magma package for linear codes over $\mathbb{Z}_{p^s}$ [FTV23].

Another important focus was on the more efficient computation of the minimum homogeneous weight for any linear code over $\mathbb{Z}_{p^s}$. MAGMA already provides a function for computing the minimum Hamming weight of a linear code over a ring, but not for the homogeneous weight. Initially, we began to develop these methods for $\mathbb{Z}_{p^s}$-linear codes, but in order to complete the 5.0 version of the package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes [BFG+22a], our focus shifted to improving the minimum Lee weight computation for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. Since this is not the main topic of the thesis, we have decided not to include this part of the research among the rest of the results. However, we present the work made on the topic of the minimum weight in the form of its implementation in the package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes in Section 7.3. In fact, the generalization from $\mathbb{Z}_4$-additive to $\mathbb{Z}_{p^s}$-additive codes is easier than

from $\mathbb{Z}_4$-additive to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. Therefore, the knowledge gained during the implementation in the mixed alphabet can be easily carried over to the case of $\mathbb{Z}_{p^s}$-additive codes.

7.2 Functions for linear codes over $\mathbb{Z}_{p^s}$

In this section we detail the functions, related to the results in this thesis, that are implemented in the MAGMA package *"Linear codes over the integer residue ring $\mathbb{Z}_{p^s}$. A MAGMA package"* [FTV23]. The first version of this package and a manual describing all functions are available in a GitHub repository (`https://github.com/merce-github/ZpAdditiveCodes`) and in the CCSG web site (`https://ccsg.uab.cat`). The content of this section corresponds exactly to the sections of the manual that cover these functions, so some of the definitions given here are repeated from previous chapters. The package can be installed in MAGMA by downloading the source code contained in the release v1.0 and by following the instructions in the README file. The examples given in the following sections are available in the `examples` folder of the GitHub repository and can be exactly reproduced. Moreover, every function included in the package have been tested, and the files containing these tests are available in the `test` folder.

In Section 7.2.1, the implementation of the results from Chapter 3 is described, providing functions to compute the information space of a code, an information set and giving two different encodings, one of which is the systematic encoding given by Theorem 19. Moreover, a function is also provided to determine if a set is an information set. The source code of these functions can be found in the file `ZpAdditiveCodes_Encoding.m`, in the `src` folder of the GitHub repository. In Section 7.2.2, the implementation of the results from Chapters 4 and 5 is described, providing functions to perform the permutation decoding method for $\mathbb{Z}_{p^s}$-linear codes and to obtain an r-PD-set for a $\mathbb{Z}_{p^s}$-linear GH code. A function that checks whether a set of permutations is an r-PD-set is also given. The source code of these functions can be found in the file `ZpAdditiveCodes_Decoding.m`, in the `src`

folder of the GitHub repository. In Section 7.2.3, the implementation of the results from Chapter 6 is described, providing a function to compute the dual code and a parity-check matrix with minimum number of rows for any $\mathbb{Z}_{p^s}$-additive code. The source code of these functions can be found in the file `ZpAdditiveCodes_Core.m`, in the `src` folder of the GitHub repository.

7.2.1 Systematic encoding

Let C be a linear code over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$. Let $C_I = \{v_I : v \in C\}$ and v_I is the vector v restricted to the subset $I \subseteq \{1, \ldots, n\}$ of coordinate positions. In MAGMA, the information space for C is $V = \mathbb{Z}_{p^s}^{t_1} \times p\mathbb{Z}_{p^s}^{t_2} \times \cdots \times p^{s-1}\mathbb{Z}_{p^s}^{t_s}$, that is, a $\mathbb{Z}_{p^s}$-submodule of $\mathbb{Z}_{p^s}^{t_1+\cdots+t_s}$ whose first t_1 coordinates are of order p^s, the next t_2 coordinates of order p^{s-1}, and so on, until the last t_s coordinates of order p. Since $p^{j-1}\mathbb{Z}_{p^s}$ is isomorphic to $\mathbb{Z}_{p^{s-j+1}}$ for all $j \in \{1, \ldots, s\}$, V is isomorphic to $\overline{V} = \mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$. Indeed, $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s) \in V$ if and only if $(\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_s) \in \overline{V}$ with $\bar{\mathbf{b}}_i = \mathbf{b}_i/p^{i-1}$. Let G be a generator matrix of C in standard form, that is, as shown in (2.10). We can define an encoding map from V to C as follows:

$$\begin{aligned} f : V &\longrightarrow C \\ (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s) &\mapsto \iota(\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_s)G, \end{aligned} \tag{7.1}$$

where ι is the map from $\mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$ to $\mathbb{Z}_{p^s}^{t_1+t_2+\cdots+t_s}$ defined as $\iota(\mathbf{u}) = (\iota_1(\mathbf{u}_1), \iota_2(\mathbf{u}_2), \ldots, \iota_s(\mathbf{u}_s))$, where $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s) \in \mathbb{Z}_{p^s}^{t_1} \times \mathbb{Z}_{p^{s-1}}^{t_2} \times \cdots \times \mathbb{Z}_p^{t_s}$, $\mathbf{u}_k \in \mathbb{Z}_{p^{s-k+1}}^{t_k}$, and $\iota_k(a) = a$ is the identity map from $\mathbb{Z}_{p^{s-k+1}}$ to $\mathbb{Z}_{p^s}$, for $1 \leq k \leq s$.

Note that since G is in standard form, we have that $J = \{1, \ldots, t_1 + \cdots + t_s\} \subseteq \{1, \ldots, n\}$ is an information set for C, and $J_p = \Phi(J) \subseteq \{1, \ldots, np^{s-1}\}$, as defined in (4.5), is an information set for $C_p = \Phi_s(C)$. Let $\Phi_s|_{J_p}$ be the projection of the image of Φ_s onto the coordinates from the set J_p. Note that if Φ_s is applied to V, then J_p is seen as a subset of $\{1, \ldots, (t_1 + \cdots + t_s)p^{s-1}\}$. Then, we denote by f_p the encoding map for C_p such that diagram (7.2) commutes. This encoding f_p is defined over the space of information vectors

for C_p, which is the k-dimensional vector space over $\mathbb{F}_p$, denoted by $V_p = \mathbb{F}_p^k$, where $k = st_1 + (s-1)t_2 + \ldots, t_s$. Note that neither the encoding f is necessarily systematic with respect to J, nor f_p is with respect to $J_p = \Phi(J)$. Recall that a map f (or f_p) is systematic if $f(v)_J = v$ for all $v \in V$ (resp. $f_p(v_p)_{J_p} = v$ for all $v_p \in V_p$).

$$\begin{array}{ccc} V & \xrightarrow{f} & C \\ \downarrow{\scriptstyle \Phi_s|_{J_p}} & & \downarrow{\scriptstyle \Phi_s} \\ V_p & \xrightarrow{f_p} & C_p \end{array} \tag{7.2}$$

Let s, t be two integers such that $s \geq t \geq 1$. We define the function $\psi_{s,t} : \mathbb{Z}_{p^s} \longrightarrow \mathbb{Z}_{p^t}$ as follows. Let $[u_0, \ldots, u_{s-1}]_p$ be the p-ary expansion of $u \in \mathbb{Z}_{p^s}$, that is, $u = \sum_{i=0}^{s-1} u_i p^i$. Then, $\psi_{s,t}(u) = \sum_{j=0}^{t-1} u_{s-t+j} p^j \in \mathbb{Z}_{p^t}$, that is, the element of $\mathbb{Z}_{p^t}$ such that its p-ary expansion is $[u_{s-t}, \ldots, u_{s-1}]_p$. Note that if $t = s$, then $\psi_{s,s}(u) = u$. We denote by $\Psi_{s,t}$ its extension coordinate-wise. Now, let us define the function $\sigma : \overline{V} \longrightarrow \iota(\overline{V})$ as $\sigma(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_s) = (\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_s)$, where the vectors $\mathbf{b}'_j \in \mathbb{Z}_{p^{s-j+1}}^{t_j}$, for $j \in \{1, \ldots, s\}$, are given by

$$\begin{aligned}
&\mathbf{b}'_1 = \iota_1(\mathbf{u}_1), \\
&\mathbf{b}'_2 = \iota_2(\mathbf{u}_2 - \Psi_{s,s-1}(\mathbf{b}'_1 A_{0,1})), \\
&\mathbf{b}'_3 = \iota_3(\mathbf{u}_3 - \Psi_{s,s-2}(\mathbf{b}'_1 A_{0,2} + p\mathbf{b}'_2 A_{1,2})), \\
&\vdots \\
&\mathbf{b}'_j = \iota_j(\mathbf{u}_j - \Psi_{s,s-j+1}(\mathbf{b}'_1 A_{0,j-1} + p\mathbf{b}'_2 A_{1,j-1} + \cdots + p^{j-2}\mathbf{b}'_{j-1} A_{j-2,j-1})), \\
&\vdots \\
&\mathbf{b}'_s = \iota_s(\mathbf{u}_s - \Psi_{s,1}(\mathbf{b}'_1 A_{0,s-1} + p\mathbf{b}'_2 A_{1,s-1} + \cdots + p^{s-2}\mathbf{b}'_{s-1} A_{s-2,s-1})),
\end{aligned}$$

and $A_{i,j}$ are the submatrices of matrix G in standard form (2.10). Now, by

using the map σ, we can define another encoding map from V to C as follows:

$$\begin{aligned} f : V &\longrightarrow C \\ (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s) &\mapsto \sigma(\bar{\mathbf{b}_1}, \bar{\mathbf{b}_2}, \ldots, \bar{\mathbf{b}_s})G. \end{aligned} \tag{7.3}$$

Let f_p be the map such that diagram (7.2) commutes, using the encoding f defined in 7.3. Then, f_p is systematic with respect to $J_p = \Phi(J)$, as it is proven in Theorem 19.

`ZpInformationSpace(C)`

`IsSystematicEncoding` BOOLELT *Default*: `true`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, return $V = \mathbb{Z}_{p^s}^{t_1} \times p\mathbb{Z}_{p^s}^{t_2} \times \cdots \times p^{s-1}\mathbb{Z}_p^{t_s}$, that is, the space of information vectors for C. Note that V is a $\mathbb{Z}_{p^s}$-submodule of $\mathbb{Z}_{p^s}^{t_1+\cdots+t_s}$ whose first t_1 coordinates of V are of order p^s, the next t_2 coordinates of order p^{s-1}, and so on, until the last t_s coordinates of order p. The function also returns the space of information vectors for the corresponding code $C_p = \Phi_s(C)$ over $\mathbb{F}_p$, where Φ_s is Carlet's generalized Gray map; that is, the k-dimensional vector space over $\mathbb{F}_p$, $V_p = \mathbb{F}_p^k$, where $k = st_1 + (s-1)t_2 + \cdots + t_s$. Finally, for the encoding process, it returns two isomorphisms f and f_p from these spaces of information vectors, V and V_p, onto C and C_p, respectively.

The map f is given as a bijective map from V to C. Nevertheless, f_p is given as an injective map from V_p to $\mathbb{F}_p^{np^{s-1}}$, where np^{s-1} is the length of C_p, having the inverse only defined for the elements in $C_p \subseteq \mathbb{F}_p^{np^{s-1}}$. These two maps are related to each other in the sense that $\Phi_s \circ f = f_p \circ \Phi_s|_{J_p}$, where $\Phi_s|_{J_p}$ denotes the projection to the set of k coordinates $J_p = \Phi(J) \subseteq \{1, \ldots, (t_1 + \cdots + t_s)p^{s-1}\}$ as defined in (4.5), for the set $J = \{1, \ldots, t_1 + \cdots + t_s\}$. That is, diagram (7.2) commutes.

The parameter `IsSystematicEncoding` specifies whether the map f_p corresponds to a systematic encoding for the code C_p or not. It is set to `true` by default. In this case, it returns a systematic encoding f_p with respect to the information set I_p given by function `ZpInformationSet(C)`. Indeed, f_p is such that diagram (7.2) commutes for the encoding f given in

(7.3). Otherwise, it returns an encoding f_p, which may not be systematic. In particular, f_p is such that diagram (7.2) commutes for the encoding f which corresponds to multiplying by the generator matrix given by function `ZpMinRowsGeneratorMatrix(C)` as shown in (7.1).

If C is a linear code over $\mathbb{Z}_4$ of type $(n; t_1, t_2)$, then `InformationSpace(C)` can also be used. The second output parameter coincides in both functions. However, `InformationSpace(C)` instead of returning $\mathbb{Z}_4^{t_1} \times p\mathbb{Z}_2^{t_2}$, it returns $p\mathbb{Z}_2^{t_2} \times \mathbb{Z}_4^{t_1}$, and both isomorphisms f and f_p from the spaces of information vectors, V and V_p, onto C and C_p, are also different.

Example 80. *The function is applied to a linear code C over $\mathbb{Z}_{27}$ of type $(5; 1, 1, 1)$. Two information vectors $(17, 21, 18)$ and $\Phi_3(17, 21, 18) = (1, 0, 0, 2, 0, 2)$ for C and $C_p = \Phi_3(C)$, respectively, are encoded by using the given encodings f and f_p. It is also checked that $\Phi_3 \circ f = f_p \circ \Phi_3|_{J_p}$, where $J_p = \Phi_3(\{1, 2, 3\}) = \{1, 2, 4, 10, 13, 19\}$.*

```
> C := LinearCode<Integers(27), 5 | [[1,2,5,8,6],
>                                   [0,3,6,12,21],
>                                   [0,0,9,9,18]]>;
> V, Vp, f, fp := ZpInformationSpace(C);

> (#V eq #C) and (#Vp eq #C);
true
> Set([f(i) : i in V]) eq Set(C);
true
> Set([fp(i) : i in Vp]) eq Set(CarletGrayMapImage(C));
true

> i := V![17,21,18];
> c := f(i);
> c;
(17 22 25 25  0)
> c in C;
true

> ip := Vp![1,0,0,2,0,2];
> cp := fp(ip);
```

```
> cp;
(1 0 2 0 2 1 2 1 0 2 0 1 0 1 2 1 2 0 2 0 1 1 2 0 0 1 2 2 0
 1 1 2 0 0 1 2 0 0 0 0 0 0 0 0 0 0)
> cp in CarletGrayMapImage(C);
true

> mapGrayC := CarletGrayMap(C);
> mapGrayV := CarletGrayMap(3, ZpType(C));
> Set(Vp) eq {mapGrayV(v) : v in V};
true
> ip eq mapGrayV(i);
true
> cp eq mapGrayC(c);
true
> [mapGrayC(f(v)) : v in V] eq [fp(mapGrayV(v)) : v in V];
true
```

`ZpInformationSet(C)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, return an information set $I \subseteq \{1, \ldots, n\}$ for C. Moreover, it also returns an information set I_p for the corresponding code $C_p = \Phi_s(C)$ over $\mathbb{F}_p$, where Φ_s is Carlet's generalized Gray map. These information sets I and I_p are returned as a sequence of $t_1 + t_2 + \cdots + t_s$ and $st_1 + (s-1)t_2 + \cdots + t_s$ coordinate positions, respectively. The information set I_p coincides with $\Phi(I)$ as defined in (4.5), and the encoding map f_p given by function `ZpInformationSpace(C)` is systematic with respect to this information set I_p.

An information set I for C is an ordered set of $t_1 + t_2 + \cdots + t_s$ coordinate positions such that $|C_I| = |C|$, where $C_I = \{v_I : v \in C\}$ and v_I is the vector v restricted to the I coordinates. An information set I_p for C_p is an ordered set of $st_1 + (s-1)t_2 + \cdots + t_s$ coordinate positions such that $|(C_p)_{I_p}| = |C_p| = |C|$.

If C is a linear code over $\mathbb{Z}_4$, then function `InformationSet(C)` can also be used even though both output parameters may be different.

`IsZpInformationSet(C, I)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and a sequence $I \subseteq \{1, \ldots, n\}$ or $I \subseteq \{1, \ldots, np^{s-1}\}$, 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 $I \subseteq \{1, \ldots, np^{s-1}\}$ is an information set for the corresponding code $C_p = \Phi_s(C)$ over $\mathbb{F}_p$, where Φ_s is Carlet's generalized Gray map.

An information set I for C is an ordered set of $t_1 + t_2 + \cdots + t_s$ coordinate positions such that $|C_I| = |C|$, where $C_I = \{v_I : v \in C\}$ and v_I is the vector v restricted to the I coordinates. An information set I_p for C_p is an ordered set of $st_1 + (s-1)t_2 + \cdots + t_s$ coordinate positions such that $|(C_p)_{I_p}| = |C_p| = |C|$.

If C is over $\mathbb{Z}_4$, function `IsZpInformationSet(C, I)` coincides with function `IsInformationSet(C, I)`, which works only for linear codes over $\mathbb{Z}_4$, but the former may perform less efficiently when $I \subseteq \{1, \ldots, 2n\}$.

Example 81. *The functions are applied to the same linear code over $\mathbb{Z}_{27}$ of type $(5; 1, 1, 1)$ as in Example 80 It is checked that the given sets are information sets for C and $C_p = \Phi_s(C)$, and that the encoding f_p given by function `ZpInformationSpace(C)` is systematic with respect to the information set I_p given by function `ZpInformationSet(C)`.*

```
> C := LinearCode<Integers(27), 5 | [[1,2,5,8,6],
>                                   [0,3,6,12,21],
>                                   [0,0,9,9,18]]>;
> V, Vp, f, fp := ZpInformationSpace(C);

> I, Ip := ZpInformationSet(C);
> I;
[ 1, 2, 3 ]
> Ip;
[ 1, 2, 4, 10, 13, 19 ]

> #PunctureCode(C, {1..5} diff Set(I)) eq #C;
true
> Cp := CarletGrayMapImage(C);
```

```
> #{Eltseq(cp)[Ip] : cp in Cp} eq #Cp;
true

> IsZpInformationSet(C, I);
true false
> IsZpInformationSet(C, Ip);
false true
> IsZpInformationSet(C, [1,2,3,4,5,6]);
false false

> ip := Vp![1,0,0,2,0,2];
> cp := fp(ip);
> Vp![cp[i] : i in Ip] eq ip;
true
> #[ip : ip in Vp | Vp![fp(ip)[i] : i in Ip] eq ip ] eq #Vp;
true
```

`Encoding(C, v)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and an element v from the space $V_p = \mathbb{F}_p^k$, where $k = st_1 + (s-1)t_2 + \cdots + t_s$, of information vectors for $C_p = \Phi_s(C)$, where Φ_s is Carlet's generalized Gray map (or an element v from the space $V = \mathbb{Z}_{p^s}^{t_1} \times p\mathbb{Z}_{p^s}^{t_2} \times \cdots \times p^{s-1}\mathbb{Z}_p^{t_s}$ of information vectors for C, given as a vector of length $t_1 + \cdots + t_s$ over $\mathbb{Z}_{p^s}$), return the codewords $c \in C$ and $c_p = \Phi_s(c) \in C_p$ corresponding to an encoding of $(\Phi_s|_{J_p})^{-1}(v) \in V$ and $v \in V_p$, respectively (or $v \in V$ and $(\Phi_s|_{J_p})(v) \in V_p$ if v is given from V), where $\Phi_s|_{J_p}$ is the projection of the image of Φ_s onto the coordinates from $J_p = \Phi(\{1, \ldots, t_1 + \cdots + t_s\})$.

This encoding for C, denoted by f, corresponds to multiplying an element in V by the generator matrix given by function `ZpMinRowsGeneratorMatrix(C)`, and the encoding for C_p corresponds to the map f_p that makes diagram (7.2) commute for the encoding f. Note that $f((\Phi_s|_{J_p})^{-1}(v)) = c$ and $f_p(v) = c_p$ (or $f(v) = c$ and $f_p((\Phi_s|_{J_p})(v)) = c_p$ if v is given from V). The encodings f and f_p coincide with the ones provided by function `ZpInformationSpace(C : IsSystematicEncoding := false)`.

`Encoding(C, L)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and a sequence L of elements from $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$, return a sequence of codewords from C, and also the corresponding sequence of codewords from $C_p = \Phi_s(C)$, given by an injective map, which corresponds to an encoding of the elements of L. The encodings for C and C_p coincide with the ones provided by function `ZpInformationSpace(C : IsSystematicEncoding := false)`.

Depending on the elements of L, the function automatically selects the appropriate encoding over $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$, by using function `Encoding(C, v)`. If it detects that L contains more than one information vector, then it computes the encoding for each one of them and returns a sequence of codewords. If it is necessary, zeros are added at the end of the sequence L to complete the last information vector before encoding.

`Encoding(C, M)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and a matrix M over $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$, returns a sequence of codewords from C, and also the corresponding sequence of codewords from $C_p = \Phi_s(C)$, given by an injective map, which corresponds to an encoding of the rows of M. The encodings for C and C_p coincide with the ones provided by function `ZpInformationSpace(C : IsSystematicEncoding := false)`.

The matrix can contain either information vectors over $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$. Depending on the length of the rows of M and its entries, the function automatically selects the appropriate encoding over $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$, by using function `Encoding(C, v)`.

Example 82. *The functions are applied to the same linear code over $\mathbb{Z}_{27}$ of type $(5; 1, 1, 1)$ as in Example 80. Some information vectors for C and $C_p = \Phi_3(C)$ are encoded by using the above functions. It is also checked that the enccoding for C corresponds to multiplying by the generator matrix given by function `ZpMinRowsGeneratorMatrix(C)`, as shown in (7.1).*

```
> C := LinearCode<Integers(27), 5 | [[1,2,5,8,6],
>                                   [0,3,6,12,21],
```

```
>                                        [0,0,9,9,18]]>;
> V, Vp, f, fp := ZpInformationSpace(C :
                          IsSystematicEncoding := false);
> mapGrayC := CarletGrayMap(C);
> mapGrayV := CarletGrayMap(3, ZpType(C));

> i := V![21,12,18];
> ip := Vp![2,2,0,1,2,2];
> ip eq mapGrayV(i);
true
> Encoding(C, ip) eq Encoding(C, i);
true
> c, cp := Encoding(C, i);
> (c eq f(i)) and (cp eq fp(ip));
true
> cp eq mapGrayC(c);
true

> ibar := [i[1], i[2] div 3, i[3] div 9];
> c eq Vector(ibar) * ZpMinRowsGeneratorMatrix(C);
true

> L := Eltseq(i) cat Eltseq(V![13,3,9]);
> Encoding(C, L);
[
    (21  0 12 18  3),
    (13  2 26 17  9)
]
[
    (2 2 2 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 2 2 2 0 0 0
     2 2 2 2 2 2 2 2 2 0 0 0 1 1 1 2 2 2),
    (1 2 0 2 0 1 0 1 2 0 2 1 0 2 1 0 2 1 2 1 0 1 0 2 0 2 1
     1 0 2 0 2 1 2 1 0 1 1 1 1 1 1 1 1 1)
]
> Encoding(C,L)[1] eq Encoding(C, i);
true

> M := Matrix(Integers(27), 3, L);
> Encoding(C, M) eq Encoding(C, L);
true
```

`SystematicEncoding(C, v)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and an element v from the space $V_p = \mathbb{F}_p^k$, where $k = st_1 + (s-1)t_2 + \cdots + t_s$, of information vectors for $C_p = \Phi_s(C)$, where Φ_s is Carlet's generalized Gray map (or an element v from the space $V = \mathbb{Z}_{p^s}^{t_1} \times p\mathbb{Z}_{p^s}^{t_2} \times \cdots \times p^{s-1}\mathbb{Z}_p^{t_s}$ of information vectors for C, given as a vector of length $t_1 + \cdots + t_s$ over $\mathbb{Z}_{p^s}$), return the codewords $c \in C$ and $c_p = \Phi_s(c) \in C_p$ corresponding to an encoding of $(\Phi_s|_{J_p})^{-1}(v) \in V$ and $v \in V_p$, respectively (or $v \in V$ and $(\Phi_s|_{J_p})(v) \in V_p$ if v is given from V), where $\Phi_s|_{J_p}$ is the projection of the image of Φ_s onto the coordinates from $J_p = \Phi(\{1, \ldots, t_1 + \cdots + t_s\})$. Unlike function `Encoding(C, v)`, in this case, the given encoding for C_p is systematic with respect to the information set I_p given by function `ZpInformationSet(C)`.

This encoding for C_p, denoted by f_p, corresponds to the systematic encoding with respect to the information set I_p, as given by the function `ZpInformationSet(C)`, and the encoding for C corresponds to the map f that makes diagram (7.2) commute for the encoding f_p. Note that $f((\Phi_s|_{J_p})^{-1}(v)) = c$ and $f_p(v) = c_p$ (or $f(v) = c$ and $f_p((\Phi_s|_{J_p})(v)) = c_p$ if v is given from V). Moreover, since f_p is systematic, $(c_p)_{I_p} = v$ (or $(c_p)_{I_p} = (\Phi_s|_{J_p})(v)$ if v is given from V). The encodings f and f_p coincide with the ones provided by function `ZpInformationSpace(C)`.

`SystematicEncoding(C, L)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and a sequence L of elements from $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$, return a sequence of codewords from C, and also the corresponding sequence of codewords from $C_p = \Phi_s(C)$. The encodings for C and C_p coincide with the ones provided by function `ZpInformationSpace(C)`. Unlike function `Encoding(C, L)`, in this case, the given encoding for C_p is systematic with respect to the information set I_p given by function `ZpInformationSet(C)`.

Depending on the elements of L, the function automatically selects the appropriate encoding: either over $\mathbb{Z}_{p^s}$ or over $\mathbb{F}_p$, by using the function

`SystematicEncoding(C, v)`. If it detects that L contains more than one information vector, then it computes the encoding for each one of them and returns a sequence of codewords. If it is necessary, zeros are added at the end of the sequence L to complete the last information vector before encoding.

`SystematicEncoding(C, M)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and a matrix M over $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$, returns a sequence of codewords from C, and also the corresponding sequence of codewords from $C_p = \Phi_s(C)$. The encodings for C and C_p coincide with the ones provided by the function `ZpInformationSpace(C)`. Unlike function `Encoding(C, M)`, in this case, the given encoding for C_p is systematic with respect to the information set I_p given by the function `ZpInformationSet(C)`.

The matrix can contain either information vectors over $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$. Depending on the length of the rows of M and its entries, the function automatically selects the appropriate encoding over $\mathbb{Z}_{p^s}$ or $\mathbb{F}_p$, by using function `SystematicEncoding(C, v)`.

Example 83. *The functions are applied to the same linear code over $\mathbb{Z}_{27}$ of type $(5; 1, 1, 1)$ as in Example 80. Some information vectors for C and $C_p = \Phi_3(C)$ are encoded by using the above functions. It is also checked that the enccoding for C_p is systematic with respect to the information set given by function `ZpInformationSet(C)`.*

```
> C := LinearCode<Integers(27), 5 | [[1,2,5,8,6],
>                                   [0,3,6,12,21],
>                                   [0,0,9,9,18]]>;
> V, Vp, f, fp := ZpInformationSpace(C);

> I, Ip := ZpInformationSet(C);
> I;
[ 1, 2, 3 ]
> Ip;
[ 1, 2, 4, 10, 13, 19 ]
```

```
> mapGrayC := CarletGrayMap(C);
> mapGrayV := CarletGrayMap(3, ZpType(C));
> [fp(vp) : vp in Vp] eq [mapGrayC(f(vp @@ mapGrayV)) :
                                                    vp in Vp ];
true
> [vp : vp in Vp] eq [ Vector([fp(vp)[i] : i in Ip ]) :
                                                    vp in Vp ];
true

> i := V![21,12,18];
> ip := Vp![2,2,0,1,2,2];
> ip eq mapGrayV(i);
true
> SystematicEncoding(C, ip) eq SystematicEncoding(C, i);
true
> c, cp := SystematicEncoding(C, i);
> (c eq f(i)) and (cp eq fp(ip));
true
> cp eq mapGrayC(c);
true
> ip eq Vp![cp[j] : j in Ip];
true

> L := Eltseq(i) cat Eltseq(V![13,3,9]);
> SystematicEncoding(C, L);
[
    (21 12 18 21 24),
    (13  5 14 11 21)
]
[
    (2 2 2 0 0 0 1 1 1 1 1 1 2 2 2 0 0 0 2 2 2 2 2 2 2 2 2 2 2
         2 0 0 0 1 1 1 2 2 2 1 1 1 0 0 0),
    (1 2 0 2 0 1 0 1 2 0 2 1 1 0 2 2 1 0 1 0 2 2 1 0 0 2 1 1 0
         2 1 0 2 1 0 2 2 2 2 0 0 0 1 1 1)
]
> SystematicEncoding(C, L)[1] eq SystematicEncoding(C, i);
true

> M := Matrix(Integers(27), 3, L);
> SystematicEncoding(C, M) eq SystematicEncoding(C, L);
true
```

7.2.2 Permutation Decoding

Permutation decoding is a technique, introduced by Prange and developed by MacWilliams, that involves finding a subset of the permutation automorphism group of a code in order to assist in decoding. This method can also be used for any nonlinear code, as long as it is systematic. Since all codes over $\mathbb{F}_p$ that are Carlet's generalized Gray map image of a linear code over $\mathbb{Z}_{p^s}$ are systematic [TV22a], we can use this method in order to decode.

Let C be a linear code over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$ and $C_p = \Phi_s(C)$ be the corresponding code over $\mathbb{F}_p$, where Φ_s is Carlet's generalized Gray map, given in Section 2.2. Let t be the error-correcting capability of C_p and $r \in \{1, \ldots, t\}$. A subset $S_p \subseteq \mathrm{PAut}(C_p)$ is an r-PD-set for C_p with respect to an information set $I_p \subseteq \{1, \ldots, np^{s-1}\}$ if every r-set of coordinate positions in $\{1, \ldots, np^{s-1}\}$ is moved out of I_p by at least one element of S_p.

Let $\Phi : \mathrm{Sym}(n) \to \mathrm{Sym}(np^{s-1})$ be the map defined as

$$\Phi(\tau)(i) = p^{s-1}\tau((i + \chi(i))/p^{s-1}) - \chi(i),$$

where $\chi(i) = p^{s-1} - (i \bmod p^{s-1})$, for all $\tau \in \mathrm{Sym}(n)$ and $i \in \{1, \ldots, np^{s-1}\}$. Given a subset $S \subseteq \mathrm{Sym}(n)$, we define

$$\Phi(S) = \{\Phi(\tau) : \tau \in S\} \subseteq \mathrm{Sym}(np^{s-1}). \tag{7.4}$$

It is easy to see that if $S \subseteq \mathrm{PAut}(C) \subseteq \mathrm{Sym}(n)$, then $\Phi(S) \subseteq \mathrm{PAut}(\Phi(C)) \subseteq \mathrm{Sym}(np^{s-1})$.

If every r-set of coordinate positions in $\{1, \ldots, n\}$ can be moved out of an information set $I \subseteq \{1, \ldots, n\}$ for C by an element of $S \subseteq \mathrm{PAut}(C)$, then $S_p = \Phi(S)$ is an r-PD-set for $C_p = \Phi_s(C)$ with respect to the information set $I_p = \Phi(I)$, where $\Phi(I)$ is defined as in (4.5) and $\Phi(S)$ as in (7.4). The converse is also true.

`IsZpPermutationDecodeSet(C, I, S, r)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, a sequence $I \subseteq \{1, \ldots, np^{s-1}\}$, a sequence S of elements in the symmetric group of permutations on the set $\{1, \ldots, np^{s-1}\}$, $\mathrm{Sym}(np^{s-1})$, and an integer $r \geq 1$, return `true` if and only if S is an r-PD-set for $C_p = \Phi_s(C)$, where Φ_s is Carlet's generalized 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 r-PD-set for $C_p = \Phi_s(C)$ with respect to the information set $\Phi(I)$, where $\Phi(I)$ is defined as in (4.5) and $\Phi(S)$ as in (7.4).

Depending on the length of the code C, its type, and the integer r, this function could take some time to compute whether S or $\Phi(S)$ is an r-PD-set for C_p with respect to I or $\Phi(I)$, respectively. Specifically, if the function returns `true`, it is necessary to check $\sum_{i=1}^{r} \binom{|I|}{i} \cdot \binom{N-|I|}{r-i}$ r-sets, where $N = n$ and $|I| = t_1 + \cdots + t_s$ when I is given as an information set for C, or $N = np^{s-1}$ and $|I| = st_1 + (s-1)t_2 + \cdots t_1$ when I is given as an information set for $C_p = \Phi_s(C)$.

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 returns `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 r-PD-set. If it is set to level 2, the percentage of the computation process performed is also printed.

If C is over $\mathbb{Z}_4$, function `IsZpPermutationDecodeSet(C, I, S, r)` coincides with function `IsPermutationDecodeSet(C, I, S, r)`, which only works for linear codes over $\mathbb{Z}_4$, but the former may perform less efficiently when $I \subseteq \{1, \ldots, 2n\}$ because it calls function `IsZpInformation Set(C, I)` instead of `IsInformationSet(C, I)`.

Example 84. *The Hadamard code C over $\mathbb{Z}_8$ of type $(8; 2, 0, 0)$ is defined, and two sets of permutations $S \subseteq \mathrm{Sym}(8)$ and $S_p \subseteq \mathrm{Sym}(32)$ are considered.*

It is checked that $\Phi(S)$ and S_p are both 3-PD-sets for $C_p = \Phi_3(C)$, with respect to the information set I_p given by function `ZpInformationSet(C)`. *The verbose flag* `IsPDsetFlag` *is set to level 2 to show the whole process of checking.*

```
> C := ZpHadamardCode(2, [2,0,0]);

> p1 := Sym(8)!(1,3,5,7)(2,4,6,8);
> p2 := Sym(8)!(1,5)(2,6)(3,7)(4,8);
> p3 := Sym(8)!(1,7,5,3)(2,8,6,4);
> S := [Sym(8)!1, p1, p2, p3];

> p1 := Sym(32)!(1,9,17,25)(2,10,18,26)(3,11,19,27)(4,12,20,28)
                (5,13,21,29)(6,14,22,30)(7,15,23,31)
                (8,16,24,32);
> p2 := Sym(32)!(1,17)(2,18)(3,19)(4,20)(5,21)(6,22)(7,23)
                (8,24)(9,25)(10,26)(11,27)(12,28)(13,29)
                (14,30)(15,31)(16,32);
> p3 := Sym(32)!(1,25,17,9)(2,26,18,10)(3,27,19,11)(4,28,20,12)
                (5,29,21,13)(6,30,22,14)(7,31,23,15)
                (8,32,24,16);
> Sp := [Sym(32)!1, p1, p2, p3];

> I, Ip := ZpInformationSet(C);

> SetVerbose("IsPDSetFlag", 2);

> IsZpPermutationDecodeSet(C, I, S, 3);
Checking whether I is an information set...
Checking whether S is in the permutation automorphism group...
Checking whether S is an r-PD-set...
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
```

```
Took 0.010 seconds (CPU time)
true

> IsZpPermutationDecodeSet(C, Ip, Sp, 3);
Checking whether I is an information set...
Checking whether S is in the permutation automorphism group...
Checking whether S is an r-PD-set...
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
Took 0.040 seconds (CPU time)
true
```

`ZpPermutationDecode(C, Ip, S, r, u)`

Given

- a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$,
- an information set $I_p \subseteq \{1, \ldots, np^{s-1}\}$ for $C_p = \Phi_s(C)$ as a sequence of coordinate positions,
- a sequence S such that either S or $\Phi(S)$ is an r-PD-set for C_p with respect to the information set I_p,
- an integer $r \in \{1, \ldots, t\}$, where t is the error-correcting capability of C_p, and
- a vector u which can be defined from the ambient space $U = \mathbb{Z}_{p^s}^n$ or from the ambient space $U_p = \mathbb{F}_p^{np^{s-1}}$,

the function attempts to decode $u \in U_p$ (or $\Phi_s(u) \in U_p$ if $u \in U$) with respect to the code C_p (or C if $u \in U$), assuming a systematic encoding with respect to the information set I_p. If the decoding algorithm succeeds in computing a codeword $u' \in C_p$ as the decoded version of $u \in U_p$ (or

$\Phi_s(u) \in U_p$ if $u \in U$), then the function returns `true`, the preimage of u' by Carlet's generalized Gray map Φ_s and finally u'. If the decoding algorithm does not succeed in decoding u, then the function returns `false`, the zero codeword in C and the zero codeword in C_p.

The permutation decoding algorithm consists in moving all errors in a received vector $u = c+e$, where $u \in U_p$, $c \in C_p$ and $e \in U_p$ is an 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_p for C_p, by using a permutation in $S \subseteq \mathrm{PAut}(C_p)$ (or $\Phi(S) \subseteq \mathrm{PAut}(C_p)$ if $S \subseteq \mathrm{PAut}(C)$). If $S \subseteq \mathrm{PAut}(C) \subseteq \mathrm{Sym}(n)$, then $\Phi(S) \subseteq \mathrm{PAut}(C_p) \subseteq \mathrm{Sym}(np^{s-1})$ is computed by using the map Φ defined in (7.4). The function does not check whether I_p is an information set for C_p, whether S or $\Phi(S)$ is an r-PD-set for C_p with respect to I_p, or whether $r \leq t$.

If C is over $\mathbb{Z}_4$, function `ZpPermutationDecode(C, Ip, S, r, u)` coincides with function `PermutationDecode(C, I, S, r, u)`, which works only for linear codes over $\mathbb{Z}_4$. However, the former only accepts an information set $I_p \subseteq \{1, \ldots, 2n\}$ for C_2, while the latter also accepts an information set $I \subseteq \{1, \ldots, n\}$ for C as long as the sequence $S \subseteq \mathrm{PAut}(C)$.

`ZpPermutationDecode(C, Ip, S, r, Q)`

Given

- a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$,
- an information set $I_p \subseteq \{1, \ldots, np^{s-1}\}$ for $C_p = \Phi_s(C)$ as a sequence of coordinate positions,
- a sequence S such that either S or $\Phi(S)$ is an r-PD-set for C_p with respect to the information set I_p,
- an integer $r \in \{1, \ldots, t\}$, where t is the error-correcting capability of C_p, and
- a sequence Q of vectors which can be defined from the ambient space $U = \mathbb{Z}_{p^s}^n$ or from the ambient space $U_p = \mathbb{F}_p^{np^{s-1}}$,

the function attempts to decode all vectors $u \in Q$ if $Q \subseteq U_p$ (or $\Phi_s(u)$ if $Q \subseteq U$) with respect to the code C_p (or C if $Q \subseteq U$), assuming a systematic encoding with respect to the information set I_p. The function returns three values: a sequence of booleans representing whether the decoding process have been successful for each $u \in Q$, a sequence of codewords of C and a sequence of codewords of C_p. For each $u \in Q$, if the decoding algorithm succeeds in computing a codeword $u' \in C_p$ as the decoded version of $u \in U_p$ (or $\Phi_s(u) \in U_p$ if $u \in U$), then it returns the preimage of u' by Carlet's generalized Gray map Φ_s in the second sequence, and u' in the third sequence. If the decoding algorithm does not succeed in decoding u, then the function returns the zero codeword in C and the zero codeword in C_p in the corresponding positions of the second and third sequences, respectively.

The permutation decoding algorithm consists in moving all errors in a received vector $u = c+e$, where $u \in U_p$, $c \in C_p$ and $e \in U_p$ is an 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_p for C_p, by using a permutation in $S \subseteq \mathrm{PAut}(C_p)$ (or $\Phi(S) \subseteq \mathrm{PAut}(C_p)$ if $S \subseteq \mathrm{PAut}(C)$). If $S \subseteq \mathrm{PAut}(C) \subseteq \mathrm{Sym}(n)$, then $\Phi(S) \subseteq \mathrm{PAut}(C_p) \subseteq \mathrm{Sym}(np^{s-1})$ is computed by using the map Φ defined in (7.4). The function does not check whether I_p is an information set for C_p or not, nor whether S or $\Phi(S)$ is an r-PD-set for C_p with respect to I_p.

If C is over $\mathbb{Z}_4$, function `ZpPermutationDecode(C, Ip, S, r, Q)` coincides with function `PermutationDecode(C, I, S, r, Q)`, which works only for linear codes over $\mathbb{Z}_4$. However, the former only accepts an information set $I_p \subseteq \{1, \ldots, 2n\}$ for C_2, while the latter also accepts an information set $I \subseteq \{1, \ldots, n\}$ for C as long as the sequence $S \subseteq \mathrm{PAut}(C)$.

Example 85. *First, the generalized Hadamard code C over $\mathbb{Z}_9$ of type $(9; 2, 0)$ is constructed. Then, it is checked that the set S_p, formed by 4 permutations in $\mathrm{Sym}(9)$, is a 3-PD-set for $C_p = \Phi_2(C)$, with respect to the information set I_p given by function `ZpInformationSet(C)`.*

A codeword $c \in C$ is considered and then perturbed by an error vector e of weight 3 (considering the homogeneous weight). The resulting vector $u = c+e$ is mapped to a vector $u_p = \Phi_2(u)$ over $\mathbb{F}_3$, which is not a codeword of C_p. Both vectors u and u_p are decoded using function `ZpPermutationDecode`*, which is successful in obtaining the correct codewords c and $c_p = \Phi_2(c)$, respectively.*

```
> C := ZpHadamardCode(3, [2,0]);
> n := Length(C);
> np := n*3;

> U := RSpace(Integers(9), n);
> Up := VectorSpace(GF(3), np);
> grayMap := CarletGrayMap(UniverseCode(Integers(9), n));

> I, Ip := ZpInformationSet(C);

> p1 := Sym(np)!(1,22,16,10,4,25,19,13,7)
                (2,23,17,11,5,26,20,14,8)
                (3,24,18,12,6,27,21,15,9);
> p2 := Sym(np)!(1,16,4,19,7,22,10,25,13)
                (2,17,5,20,8,23,11,26,14)
                (3,18,6,21,9,24,12,27,15);
> p3 := Sym(np)!(1,10,19)(2,11,20)(3,12,21)(4,13,22)(5,14,23)
>               (6,15,24)(7,16,25)(8,17,26)(9,18,27);
> Sp := [Sym(np)!1, p1, p2, p3];

> IsZpPermutationDecodeSet(C, Ip, Sp, 3);
true

> c := C![0,1,2,3,4,5,6,7,8];
> e := U![0,1,0,0,0,0,0,0,0];
> u := c+e;
> cp := grayMap(c);
> up := grayMap(u);
> (u in U) and (up in Up);
true

> isDecoded, uDecoded, upDecoded := ZpPermutationDecode(C, Ip,
                                                        Sp, 3, u);
> isDecoded;
```

```
true
> uDecoded eq c;
true
> upDecoded eq cp;
true

> isDecoded, uDecoded, upDecoded := ZpPermutationDecode(C, Ip,
                                                        Sp, 3, up);
> isDecoded;
true
> uDecoded eq c;
true
> upDecoded eq cp;
true
```

`PDSetHadamardCodeZps(p, [t_1, t_2, ..., t_s])`

`AlgMethod` MONSTGELT *Default*: `"Deterministic"`

Given a prime p and a sequence of nonnegative integers $[t_1, \ldots, t_s]$ with $t_1 > 0$ and $s > 1$, the generalized Hadamard code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, given by function `ZpHadamardCode(p, [t_1, ..., t_s])`, is considered. The function returns an information set $I = \{i_1, \ldots, i_{t_1+\cdots+ts}\} \subseteq \{1, \ldots, n\}$ for C together with a subset S of the permutation automorphism group of C such that $\Phi(S)$ is an r-PD-set for $C_p = \Phi_s(C)$ with respect to $\Phi(I)$, where Φ_s is Carlet's generalized Gray map as defined in (2.13), $\Phi(I)$ is as in (4.5) and $\Phi(S)$ as in (7.4). The function also returns the information set $\Phi(I)$ and the r-PD-set $\Phi(S)$.

Note that for $p = 2$ and $[t_1, \ldots, t_s] = [1, 0, \ldots, 0, t_s]$ or $[t_1, \ldots, t_s] = [1, 0, \ldots, 0, 1, t_s]$, we have that C_p is linear [FVV19], so it is possible to find an r-PD-set of size $r+1$ for C_p, for any $r \leq \lfloor 2^m/(m+1) \rfloor$, by using function `PDSetHadamardCode(m)` with $m = t_s+s$ if $[t_1, \ldots, t_s] = [1, 0, \ldots, 0, t_s]$ and $m = t_s + s + 2$ if $[t_1, \ldots, t_s] = [1, 0, \ldots, 0, 1, t_s]$.

The information sets I and $\Phi(I)$ are returned as sequences of $t_1 + \cdots + t_s$ and $s(t_1 - 1) + (s - 1)t_2 + \cdots + t_s$ integers, giving the coordinate positions that correspond to the information sets for C and C_p, respectively.

The sets S and $\Phi(S)$ are also returned as sequences of elements in the symmetric groups $\mathrm{Sym}(n)$ and $\mathrm{Sym}(np^{s-1})$ of permutations on the set $\{1,\ldots,n\}$ and $\{1,\ldots,np^{s-1}\}$, respectively.

A deterministic algorithm is used by default. In this case, when $t_1 \geq 2$, the function first computes r as the maximum of $g_p^{t_1,\ldots,t_s}$ and $\tilde{f}_p^{t_1,\ldots,t_s}$, where $g_p^{t_1,\ldots,t_s}$ is given by the general construction described in Chapter 5 and

$$\tilde{f}_p^{t_1,\ldots,t_s} = \max\{f_p^{t_1,0,\ldots,0}, f_p^{1,t_2,0,\ldots,0}, \ldots, f_p^{1,0,\ldots,0,t_s}\} \leq f_p^{t_1,\ldots,t_s},$$

where $f_p^{t_1,\ldots,t_s}$ is the theoretical upper bound for the value of r such that there exists an r-PD-set of size $r+1$ for a Hadamard code over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, given in Corollary 38. Let i be the first index $i \geq 1$ such that the maximum $\tilde{f}_p^{t_1,\ldots,t_s}$ is achieved. Then, if $\tilde{f}_p^{t_1,\ldots,t_s} > g_p^{t_1,\ldots,t_s}$, it constructs an r-PD-set of size $r+1$ for the generalized Hadamard code over $\mathbb{Z}_{p^s}$ of type $(n; t_1, 0, \ldots, 0)$ if $i = 1$ or $(n; 1, 0, \ldots, 0, t_i, 0, \ldots, 0)$ otherwise, which is transformed into an r-PD-set for C, by using the recursive construction defined in Chapter 4. The value of r remains unchanged after the recursive construction, so $r = \tilde{f}_p^{t_1,\ldots,t_s}$. On the other hand, if $g_p^{t_1,\ldots,t_s} \geq \tilde{f}_p^{t_1,\ldots,t_s}$, the general construction is applied and an r-PD-set of size $r+1$ with $r = g_p^{t_1,\ldots,t_s}$ is obtained. When $t_1 = 1$ and there exists an index $i \geq 2$ such that $t_i \neq 0$ and $t_2 = \cdots = t_{i-1} = 0$, it first constructs an r-PD-set of size $r+1$ for the generalized Hadamard code over $\mathbb{Z}_{p^s}$ of type $(n; 1 + t_i, t_{i+1}, \ldots, t_s)$, which satisfies that $1 + t_i \geq 2$, by following the same process as described above when $t_1 \geq 2$. Then, the obtained r-PD-set is transformed into an r-PD-set for C as described in Chapters 4 and 5.

If the parameter `AlgMethod` is assigned the value `"Nondeterministic"`, the function tries to improve the previous results by finding an r-PD-set of size $r+1$ such that $\tilde{f}_p^{t_1,\ldots,t_s} \leq r \leq f_p^{t_1,\ldots,t_s}$. In this case, the function starts from the maximum value of $r = f_p^{t_1,\ldots,t_s}$ and attempts to find an r-PD-set within a time limit of 5.0 seconds of "user time". This is performed 5 times, each time starting from an empty set and trying to construct the

r-PD set incrementally, by adding elements randomly. If such an r-PD-set is not found, the value of r decreases by one and the same process takes place with the new value of r. The value of r keeps decreasing until an r-PD-set is found or $r = \tilde{f}_p^{t_1,\ldots,t_s}$.

The verbose flag `PDsetHadamardFlag` is set to level 0 by default. If it is set to level 1, some information about the process of constructing the r-PD-set of size $r+1$ is shown. Moreover, the value of the theoretical upper bound $f_p^{t_1,\ldots,t_s}$ given in Corollary 38 is also shown.

If $p = 2$ and $s = 2$, then function `PDSetHadamardCodeZ4(`t_1`,` $2t_1 + t_2 - 1$`)` can also be used. Both function only coincide when $t_2 = 0$. When $t_2 > 0$, the output parameters I and $\Phi(I)$ coincide as sets and `PDSetHadamardCodeZps(2,` $[t_1, t_2]$`)` may give a larger r-PD-set.

Example 86. *First, the generalized Hadamard code C over $\mathbb{Z}_{27}$ of type $(729; 3, 0, 0)$ is constructed. Then, by using the function* ***`PDSetHadamard-CodeZps(p, [t1,t2,...,ts])`****, we obtain an information set I for C and a set $S \subseteq \mathrm{PAut}(C)$ such that $I_p = \Phi(I)$ is an information set for $C_p = \Phi_3(C)$ and $S_p = \Phi(S)$ is an r-PD-set of size $r+1$ for C_p with $r = 242$. In this case, the maximum value of r is obtained since $r \leq f_3^{3,0,0} = 242$.*

A codeword $c \in C$ is considered and the corresponding $c_p = \Phi_3(c) \in C_p$ is perturbed by an error vector e_p of Hamming weight 242. The resulting vector $u_p = c_p + e_p$ is decoded using function ***`ZpPermutationDecode`****, which is successful in obtaining the correct codewords c and c_p, respectively.*

```
> p := 3;
> type := [3,0,0];
> s := #type;
> C := ZpHadamardCode(p, type);
> n := Length(C);
> np := n*p^(s-1);

> U := RSpace(Integers(p^s), n);
> Up := VectorSpace(GF(p), np);
> grayMap := CarletGrayMap(UniverseCode(Integers(p^s), n));
```

```
> I, S, Ip, Sp := PDSetHadamardCodeZps(p, type);
> r := #Sp-1; r;
242
> t1 := type[1];
> r eq Floor((p^(s*(t1-1))-t1)/t1);
true
> c := C![6^^n];
> cp := grayMap(c);
> ep := Up![1^^r,0^^(np-r)];
> up := cp + ep;

> isDecoded, uDecoded, upDecoded := ZpPermutationDecode(
                                      C, Ip, Sp, r, up);
> isDecoded;
true
> uDecoded eq c;
true
> upDecoded eq cp;
true
```

If we consider a generalized Hadamard code over $\mathbb{Z}_{p^s}$ that is not of type $(n; t_1, 0, \ldots, 0)$ or $(n; 1, 0, \ldots, 0, t_i, 0, \ldots, 0)$ for $i \in \{2, \ldots, s\}$, then the deterministic method may not be able to give an r-PD-set of size $r+1$ for the maximum value of $r \leq f_p^{t_1,\ldots,t_s}$. However, by selecting the `Nondeterministic` *method, the function tries to increase the value of r. The flag* `PDsetHadamardFlag` *is set to level 1 to see the process of trying to obtain the r-PD-set.*

```
> p := 3;
> type := [1,0,1,1];
> s := #type;
> C := ZpHadamardCode(p, type);
> n := Length(C);
> np := n*p^(s-1);

> SetVerbose("PDSetHadamardFlag", 1);
> I, S, Ip, Sp := PDSetHadamardCodeZps(p, type);
The upper bound for an r-PD-set of size r+1 is 8.
A 5-PD-set of size 6 has been obtained by applying the general
construction for a Hadamard code of type [ 2, 1 ] and then
adapted for a Hadamard code oftype [ 1, 0, 1, 1 ].
```

```
> r := #Sp-1; r;
5
> IsZpPermutationDecodeSet(C, I, S, r);
true

> I, S, Ip, Sp := PDSetHadamardCodeZps(p, type :
                AlgMethod := "Nondeterministic");
The upper bound for an r-PD-set of size r+1 is 8.
Trying to find an 8-PD-set of size 9 randomly...
Trying to find an 7-PD-set of size 8 randomly...
A 7-PD-set has been found in iteration 2 of 5!!
> r := #Sp-1; r;
7
> IsZpPermutationDecodeSet(C, I, S, r);
true
```

When $p = 2$ and $s = 2$, function `PDSetHadamardCodeZps(p, [t1,0])` *coincides with function* `PDSetHadamardCodeZ4(t1, 2*t1-1)`.

```
> I, S, Ibin, Sbin := PDSetHadamardCodeZps(2, [3,0]);
The upper bound for an r-PD-set of size r+1 is 4.
A 4-PD-set of size 5 has been obtained.
> I2, S2, I2bin, S2bin := PDSetHadamardCodeZ4(3, 5);

> I eq I2; I;
true
[ 1, 2, 5 ]
> S eq S2; S;
true
[
  Id($),
  (1, 12, 13, 8, 9, 4, 5, 16)(2, 15, 14, 11, 10, 7, 6, 3),
  (1, 13, 11, 7)(2, 8, 12, 14)(3, 15, 9, 5)(4, 6, 10, 16),
  (1, 8, 5, 10)(2, 9, 16, 13)(3, 14, 7, 4)(6, 15, 12, 11),
  (1, 11)(2, 12)(3, 9)(4, 10)(5, 15)(6, 16)(7, 13)(8, 14)
]
> Ibin eq I2bin; Ibin;
true
[ 1, 2, 3, 4, 9, 10 ]
> Sbin eq S2bin; Sbin;
true
```

```
[
  Id($),
  (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),
  (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),
  (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),
  (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)
]
```

7.2.3 Parity-check matrix and dual code

Let $\mathcal{C}$ by a linear code over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, with a generator matrix G in standard form, that is,

$$G = \begin{pmatrix} \mathrm{Id}_{t_1} & A_{0,1} & A_{0,2} & A_{0,3} & \cdots & \cdots & A_{0,s} \\ \mathbf{0} & p\mathrm{Id}_{t_2} & pA_{1,2} & pA_{1,3} & \cdots & \cdots & pA_{1,s} \\ \mathbf{0} & \mathbf{0} & p^2\mathrm{Id}_{t_3} & p^2A_{2,3} & \cdots & \cdots & p^2A_{2,s} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & p^{s-1}\mathrm{Id}_{t_s} & p^{s-1}A_{s-1,s} \end{pmatrix}. \tag{7.5}$$

Then, the following matrix H is a parity-check matrix for $\mathcal{C}$.

$$H = \begin{pmatrix} H_{1,1} & pH_{1,2} & \cdots & p^{s-3}H_{1,s-2} & p^{s-2}H_{1,s-1} & p^{s-1}H_{1,s} \\ H_{2,1} & pH_{2,2} & \cdots & p^{s-3}H_{2,s-2} & p^{s-2}H_{2,s-1} & p^{s-1}\mathrm{Id}_{t_2} \\ H_{3,1} & pH_{3,2} & \cdots & p^{s-3}H_{3,s-2} & p^{s-2}\mathrm{Id}_{t_3} & \mathbf{0} \\ H_{4,1} & pH_{4,2} & \cdots & p^{s-3}\mathrm{Id}_{t_4} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \iddots & \vdots & \vdots & \vdots \\ H_{s,1} & p\mathrm{Id}_{t_s} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathrm{Id}_{n-t} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{7.6}$$

where the submatrices $H_{i,j}$ can be constructed as

$$H_{i,j} = -\left(A_{i,s-j+2} + \sum_{k=i+1}^{s-j+1} A_{i,k} H_{k,j} \right) \tag{7.7}$$

as given in Theorem 77, for all $1 \leq j \leq s$ and $1 \leq i < s - j + 1$. Note that $H_{s-j+1,j} = -A_{s-j+1,s-j+2}$.

In order to compute a parity-check matrix of a linear code over $\mathbb{Z}_{p^s}$, we use Algorithm 2, given in Chapter 6. Example 87 shows that the function `ZpDual(C)`, defined below, which uses this method, is faster than using the MAGMA function `Dual(C)`, which computes the dual code of any linear code over a ring. Similarly, Example 88 shows that the function `ZpMinRowsParityCheckMatrix(C)`, which also uses Algorithm 2, is faster than using the MAGMA function `ParityCheckMatrix(C)`, which computes the parity-check matrix of any linear code over a ring.

`ZpDual(C)`

Given a linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, return the dual code D of C. The dual code consists of all codewords in the $\mathbb{Z}_{p^s}$-space $V = \mathbb{Z}_{p^s}^n$ which are orthogonal to all codewords of C. In particular, the dual code D is of type $(n; n - t_1 - t_2 - \cdots - t_s, t_s, \ldots, t_2)$.

This function creates the generator matrix of D following Algorithm 2. This construction improves the computation time with respect to the generic function `Dual(C)` for codes over finite rings.

If C is over $\mathbb{Z}_4$, function `ZpDual(C)` coincides with function `DualZ4(C)`, but the former may perform less efficiently in general.

`ZpMinRowsParityCheckMatrix(C)`

A parity-check matrix for the linear code C over $\mathbb{Z}_{p^s}$ of type $(n; t_1, \ldots, t_s)$, with the minimum number of rows, that is, with $n - t_1$ rows. It also returns the sequence $[n - t_1 - t_2 - \cdots - t_s, t_s, \ldots, t_2]$ and a permutation transforming $C^\perp$ into a permutation-equivalent code with generator matrix in standard form.

This function should be faster for most codes over $\mathbb{Z}_{p^s}$ than the general function `ParityCheckMatrix(C)` for codes over finite rings. Another parity-check matrix for the code C can be obtained as the generator matrix of the dual of C with the minimum number of rows, that is, as `ZpMinRowsGeneratorMatrix(ZpDual(C))`.

If C is a linear code over $\mathbb{Z}_4$ of type $(n; t_1, t_2)$, then `MinRowsParityCheckMatrix(C)` can also be used to obtain a parity-check matrix with minimum number of rows. However, only the matrix is returned, which may not coincide with the one given by `ZpMinRowsGeneratorMatrix(C)`.

Example 87. *For a random linear code C over $\mathbb{Z}_{3^{10}}$, the dual code is computed by using function `Dual(C)` and the new function `ZpDual(C)`, which is only applicable for linear codes over $\mathbb{Z}_{p^s}$ with p prime, and is usually faster than the former one. It is also checked the relation between a generator matrix of C and of $C^{\perp}$, and how to obtain the type of the dual code.*

```
> C := RandomZpAdditiveCode(3, 1000, [2^^10]);

> time D := Dual(C);
Time: 12.875
> time Dp := ZpDual(C);
Time: 0.234
> D eq Dp;
true

> G := GeneratorMatrix(C);
> H := GeneratorMatrix(D);
> IsZero(G * Transpose(H));
true

> ZpTypeDual(C) eq ZpType(ZpDual(C));
true
```

Example 88. *Again, for a random linear code C over $\mathbb{Z}_{3^{10}}$, the relation between a generator matrix of C and of $C^{\perp}$, both having minimum number of rows, is checked. Different ways of computing a parity-check matrix for C are also shown.*

```
> C := RandomZpAdditiveCode(3, 1000, [2^^10]);

> G := ZpMinRowsGeneratorMatrix(C);
> H := ZpMinRowsParityCheckMatrix(C);
> IsZero(G * Transpose(H));
true
> LinearCode(G) eq C;
true
> LinearCode(H) eq ZpDual(C);
true

> Nrows(G) eq &+ZpType(C);
true
> Nrows(H) eq &+ZpTypeDual(C);
true

> time H := ZpMinRowsParityCheckMatrix(C);
Time: 0.016
> time H2 := ZpMinRowsGeneratorMatrix(ZpDual(C));
Time: 6.297
> time H3 := ParityCheckMatrix(C);
Time: 31.422
> H eq H2;
false
> H2 eq H3;
false
> H eq H3;
false
> LinearCode(H2) eq ZpDual(C);
true
> LinearCode(H3) eq ZpDual(C);
true
```

Example 89. *A linear Hadamard code over $\mathbb{Z}_4$ is constructed, and the outputs of the functions* `ZpDual(C)`, `ZpMinRowsGeneratorMatrix(C)` *and* `ZpMinRowsParityCheckMatrix(C)` *are compared with the ones obtained by using the functions* `DualZ4(C)`, `MinRowsGeneratorMatrix(C)` *and* `MinRowsParityCheckMatrix(C)`, *respectively, which work only for linear codes over $\mathbb{Z}_4$.*

```
> C := ZpHadamardCode(2, [3,1]);

> G_Zp, type := ZpMinRowsGeneratorMatrix(C);
> G_Z4, t2, t1 := MinRowsGeneratorMatrix(C);
> G_Zp eq G_Z4;
false
> LinearCode(G_Zp) eq LinearCode(G_Z4);
true
> type eq [t1, t2];
true

> H_Zp := ZpMinRowsParityCheckMatrix(C);
> H_Z4 := MinRowsParityCheckMatrix(C);
> H_Zp eq H_Z4;
false
> LinearCode(H_Zp) eq LinearCode(H_Z4);
true

> ZpDual(C) eq DualZ4(C);
true
```

7.3 Functions for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes

In this section, we present the developed functions regarding $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, which provide different methods to compute the minimum Lee weight of a $\mathbb{Z}_2\mathbb{Z}_4$-additive code. The most commonly used method is the Brouwer-Zimmermann method, which has already been introduced in Section 2.5 for linear codes over finite fields. In [Whi06], this method is generalized to linear codes over the ring $\mathbb{Z}_4$. Adapting this generalization to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes we obtain a new method to compute the minimum Lee weight of these codes.

In Section 7.3.1, we give a brief introduction to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. In Section 7.3.2, we describe how the Brouwer-Zimmermann method can be adapted to compute the minimum Lee weight of a $\mathbb{Z}_2\mathbb{Z}_4$-additive code. In

Section 7.3.3, we give some details about the implementation of the Brouwer-Zimmermann method which has been included in the version 5.0 of the package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, describing all the developed functions and showing some examples.

7.3.1 A brief introduction to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes

A $\mathbb{Z}_2\mathbb{Z}_4$-*additive code* is a subgroup of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$, that is, a linear code with some coordinates over $\mathbb{Z}_2$ and some over $\mathbb{Z}_4$. They have been extensively studied by the members of CCSG, as it can be seen from the references given in [BFP+22b]. The same authors have also developed a MAGMA package [BFG+22a] in order to provide support for these codes. A $\mathbb{Z}_2\mathbb{Z}_4$-additive code is a subgroup of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$, that is, it has α coordinates over $\mathbb{Z}_2$ and β coordinates over $\mathbb{Z}_4$. They are simultaneously a generalization of binary codes (when $\beta = 0$) and of quaternary codes (when $\alpha = 0$). The image of a $\mathbb{Z}_2\mathbb{Z}_4$-additive code by the extended Gray map $\Phi : \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta \longrightarrow \mathbb{Z}_2^{\alpha+2\beta}$ given by

$$\Phi(x, y) = (x, \phi(y_1), \ldots, \phi(y_\beta)) \quad \forall x \in \mathbb{Z}_2^\alpha,\ \forall y = (y_1, \ldots, y_\beta) \in \mathbb{Z}_4^\beta; \tag{7.8}$$

where ϕ is the usual Gray map, is a binary code called $\mathbb{Z}_2\mathbb{Z}_4$-*linear*. Similarly to the quaternary case, $\mathbb{Z}_2\mathbb{Z}_4$-linear codes are not necessarily linear as binary codes.

Any $\mathbb{Z}_2\mathbb{Z}_4$-additive code C is permutation-equivalent to a $\mathbb{Z}_2\mathbb{Z}_4$-additive code C_{SF} which is in standard form, that is, with a generator matrix in the form

$$\left(\begin{array}{cc|ccc} I_\kappa & T_b & 2T_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2T_1 & 2I_{\gamma-\kappa} & \mathbf{0} \\ \hline \mathbf{0} & S_b & S_q & R & I_\delta \end{array}\right), \tag{7.9}$$

where T_b, S_b are matrices over $\mathbb{Z}_2$; T_1, T_2, R are matrices over $\mathbb{Z}_4$ with all entries in $\{0, 1\} \subset \mathbb{Z}_4$; and S_q is a matrix over $\mathbb{Z}_4$.

The *Lee weight* of a codeword $\mathbf{v}$, denoted by $\mathrm{wt}^{\mathrm{L}}(\mathbf{v})$ is the Hamming weight of the binary part of $\mathbf{v}$ (that is, the first α coordinates) plus the Lee

weight of the quaternary part of $\mathbf{v}$ (that is, the rest of the coordinates) (see [BFP$^+$10]). Moreover, it corresponds to the Hamming weight of $\Phi(\mathbf{v})$, where Φ is the Gray map defined in (7.8). For $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, the Lee weight and Lee distance distributions coincide (in particular, minimum Lee weight and minimum Lee distance coincide).

7.3.2 Brouwer-Zimmermann method for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes

In Section 2.5, we presented a brief introduction to the computation of the minimum weight for linear codes over finite fields. We focused on Brouwer's algorithm and its subsequent improvement proposed by Zimmermann. In [Whi06] the Brouwer-Zimmermann algorithm is generalized to linear codes over the ring $\mathbb{Z}_4$. In this section, we generalize this method to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes.

In Brouwer's method, a sequence of systematic generator matrices with disjoint information sets is constructed. This can be done by diagonalizing the columns which have not been used yet in any information set and then appending the already used part, with the appropriate linear combinations of rows. However, for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, when considering a generator matrix of the code C restricted to the remaining columns, it may happen that the type of the generated code is not equal to the type of C. In these cases Brouwer's method stops, since some of the already used coordinates should be included in the information set.

In Zimmermann's variation, we can keep constructing generator matrices, reusing information coordinates positions from previous matrices. While this is trivial for linear codes over finite fields, it represents a bigger issue for linear codes over finite rings, especially when they are defined over mixed alphabets. The main difficulty in this process is to create a generator matrix in standard form while requiring a certain subset of columns to be in the information set. To illustrate the mechanism, consider a $\mathbb{Z}_2\mathbb{Z}_4$-additive code C of type $(\alpha, \beta; \gamma, \delta; \kappa)$. Suppose G_1 is a generator matrix of C in standard form, as in (7.9). The information set consists of the first κ and last δ

coordinates, that is, $I_{free} = \{1, \ldots, \kappa, n-\delta+1, \ldots, n\}$. The next generator matrix should, ideally, have an information set formed by columns in the remaining positions. Let H_1 be the matrix formed by these columns, that is,

$$H_1 = \left(\begin{array}{c|cc} T_b & 2T_2 & \mathbf{0} \\ \mathbf{0} & 2T_1 & 2I_{\gamma-\kappa} \\ S_b & S_q & R \end{array}\right).$$

Let C_1 be the code generated by matrix H_1 of type $(\alpha-\kappa, \beta-\delta; \gamma', \delta'; \kappa')$. We know for sure that $\delta' \leq \delta$, since the order of a row can not increase when we restrict the matrix G to some columns. However, γ' can be greater than γ since some rows of order 4 become of order 2 in H_1. Similarly, κ' can be greater than κ. This does not mean that we can obtain κ' new information coordinates, since the last δ rows (those originally of order 4) can not be used as generators of order 2, once the rest of columns are appended. The actual number of new binary information coordinates is given by the type of the subcode generated by the rows of order 2 of H_1. Let κ'' be that number, which satisfies $\kappa'' \leq \kappa'$ and $\kappa'' \leq \kappa$. Then, we can obtain

$$H_1' = Q_1 H_1 P_1 = \left(\begin{array}{cc|cc} I_{\kappa''} & K_1 & 2K_2 & \mathbf{0} \\ \mathbf{0} & A_1 & 2A_2 & \mathbf{0} \\ \mathbf{0} & D_1 & D_2 & I_{\delta'} \end{array}\right),$$

where Q_1 is a row transformation matrix and P_1 is a column permutation matrix. Let

$$\tilde{P}_1 = \left(\begin{array}{ccc} I_\kappa & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & P_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_\delta \end{array}\right).$$

Appending the rest of the matrix, we obtain

$$G_1' = Q_1 G_1 \tilde{P}_1 = \left(\begin{array}{ccc|ccc} K_0 & I_{\kappa''} & K_1 & 2K_2 & \mathbf{0} & 2K_3 \\ A_0 & \mathbf{0} & A_1 & 2A_2 & \mathbf{0} & A_3 \\ D_0 & \mathbf{0} & D_1 & D_2 & I_{\delta'} & D_3 \end{array}\right).$$

Note that the subcode generated by the submatrix

$$A = \left(\begin{array}{ccc|ccc} A_0 & \mathbf{0} & A_1 & 2A_2 & \mathbf{0} & A_3 \end{array} \right)$$

is of type $(\alpha, \beta; \gamma - \kappa'', \delta - \delta'; \kappa - \kappa'')$. We can transform this matrix into standard form, obtaining an identity submatrix of size $\kappa - \kappa''$ in A_0 and an identity submatrix of size $\delta - \delta'$ in A_3. Then, the rest of the matrix is diagonalized using these identity submatrices. Let M_1 be the transformation matrix associated to these row combinations. Then,

$$G_2 = M_1 G_1' = M_1 Q_1 G_1 = \left(\begin{array}{cccc|cccc} K_0' & \mathbf{0} & I_{\kappa''} & K_1' & 2K_2' & \mathbf{0} & \mathbf{0} & 2K_3' \\ A_{0,1} & I_{\kappa-\kappa''} & \mathbf{0} & A_{1,1} & 2A_{2,1} & \mathbf{0} & \mathbf{0} & 2A_{3,1} \\ A_{0,2} & \mathbf{0} & \mathbf{0} & A_{1,2} & 2A_{2,2} & \mathbf{0} & \mathbf{0} & 2A_{3,2} \\ A_{0,3} & \mathbf{0} & \mathbf{0} & A_{1,3} & A_{2,3} & \mathbf{0} & I_{\delta-\delta'} & A_{3,3} \\ D_0' & \mathbf{0} & \mathbf{0} & D_1' & D_2' & I_{\delta'} & \mathbf{0} & D_3' \end{array} \right),$$

where we have assumed a column permutation to simplify the notation; but there is no need to swap any column. The process continues with the restricted matrix

$$H_2 = \left(\begin{array}{c|c} K_1' & 2K_2' \\ A_{1,1} & 2A_{2,1} \\ A_{1,2} & 2A_{2,2} \\ A_{1,3} & A_{2,3} \\ D_1' & D_2' \end{array} \right).$$

Note that the part of G_2 not included in H_2 corresponds to coordinate positions that have already been used as information coordinates. The procedure ends when a new generator matrix is not able to provide any new information coordinate (either binary or quaternary) or when all columns have been included in an information set. This process gives a sequence of generator matrices $\{G_1, G_2, \ldots, G_h\}$ with minimum number of rows and pairwise disjoint information sets $\{I_1, I_2, \ldots, I_h\}$.

Brouwer's algorithm updates the lower bound assuming pairwise disjoint

information sets. However, when there is an overlap between different information sets, a correction should be made to the lower bound in order to account for the actual contribution of the new information set. Moreover, in the case of quaternary or $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, a distinction should be made between the free part of the information vector (the one associated to rows of order 4) and the torsion part (the one associated to rows of order 2). Only the free part of the information vector contributes to the lower bound of the minimum weight. This is addressed in [Whi06] for quaternary codes and here we generalize it to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes.

Each information set I_i is the disjoint union of the free and torsion parts, $I_i = I_i^{free} \cup I_i^{tor}$. Then, the *relative information set* I_i^{rel} is defined as the set of free information coordinates that have not been used in any information set I_j^{free}, for $j < i$, that is,

$$I_i^{rel} = I_i^{free} \setminus \left(\bigcup_{j<i} I_j^{free} \right) \tag{7.10}$$

and the *rank deficit* of each information set is defined as

$$k_i^{def} = |I^{tor}| + 2|I_i^{free} \setminus I_i^{rel}|. \tag{7.11}$$

Note that

$$k_1^{def} = |I^{tor}| = \gamma - \kappa$$

and, for any $i > 1$, if H_{i-1} generates a subcode of type $(\alpha, \beta; \gamma - \kappa'', \delta - \delta'; \kappa - \kappa'')$, then

$$k_i^{def} = \gamma - \kappa + (\kappa - \kappa'') + 2(\delta - \delta'). \tag{7.12}$$

Lemma 90 and Theorem 91 give a lower bound for the Lee weight of all codewords that have not been enumerated yet. They are completely equivalent to Lemma 5.6 and Theorem 5.7 in [Whi06], which are stated for linear codes over $\mathbb{Z}_4$.

Lemma 90. *Let G_i be a generator matrix with minimum number of rows and*

rank deficit k^{def}. *If* G *generates a* $\mathbb{Z}_2\mathbb{Z}_4$*-additive code* C *of type* $(\alpha, \beta; \gamma, \delta; \kappa)$ *and*

$$S = \{\mathbf{m}G : \mathbf{m} \in \mathbb{Z}_2^{\gamma} \oplus \mathbb{Z}_4^{\delta},\ \mathrm{wt}^{\mathrm{L}}(\mathbf{m}) \leq r\},$$

then for all $\mathbf{v} \in C \setminus S$

$$\mathrm{wt}^{\mathrm{L}}(\mathbf{v}) \geq r + 1 - k^{def}. \tag{7.13}$$

Theorem 91. *Let* $G_1, G_2, \ldots, G_h$ *be a sequence of generator matrices with minimum number of rows of a* $\mathbb{Z}_2\mathbb{Z}_4$*-additive code* C *of type* $(\alpha, \beta; \gamma, \delta; \kappa)$. *Let* $k_1^{def}, k_2^{def}, \ldots, k_h^{def}$ *be the corresponding rank deficits. Then, for any* r, *if*

$$S_i = \{\mathbf{m}G_i : \mathbf{m} \in \mathbb{Z}_2^{\gamma} \oplus \mathbb{Z}_4^{\delta},\ \mathrm{wt}^{\mathrm{L}}(\mathbf{m}) \leq r\},$$

for each matrix G_i, *then all* $\mathbf{v} \in C \setminus \bigcup_{i=1}^{h} S_i$ *satisfy*

$$\mathrm{wt}^{\mathrm{L}}(\mathbf{v}) \geq \sum_{i=1}^{h} \max(0, r + 1 - k_i^{def}). \tag{7.14}$$

Note that the rank deficits determine the contribution of each generator matrix to the lower bound. The terms $(\kappa - \kappa'')$ and $(\delta - \delta'')$ in (7.11) are specific to each generator matrix, but the term $\gamma - \kappa$ is common for all the matrices in the sequence. This means that, even when there is no overlap of free information coordinates, the torsion part makes a significant penalty in the efficiency of the algorithm.

7.3.3 Implementation in Magma

In this section, we give present some details regarding the implementation of the Brouwer-Zimmermann method in the package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes [BFG+22a]. Moreover, we also compare it with other methods that have been implemented. Then, the description of each developed function is given, along with some examples and use-cases.

Five different methods have been implemented in the package to compute the minimum Lee weight of a $\mathbb{Z}_2\mathbb{Z}_4$-additive code. Two of them were already available in the version 4.0 [BFG+17]: the first one is by using brute

force, which is equivalent to compute the whole Lee weight distribution of the code. The second one is by using the representation of the code as the union of cosets of the kernel, and the known Brouwer-Zimmermann's algorithm applied to binary linear codes given by the Gray map image of cosets of the kernel [VZP15]. For the version 5.0 [BFG+22a], three additional methods have been included in the package: the third method uses Brouwer's algorithm adapted to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, and the fourth one uses Zimmermann's variation (also known as Brouwer-Zimmermann method) adapted to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. The fifth method considers the code as a quaternary code and uses the minimum weight function already implemented in MAGMA. Each method may perform better in different circumstances, which are usually related to the parameters of the code and the minimum distance itself. For small codes, the brute force method can be faster than any other method, but as the number of codewords grows it becomes unfeasible.

Brouwer's and Brouwer-Zimmermann algorithms are enumeration methods. They rely on different generator matrices of the code in standard form, each one with respect to a different set of information coordinate positions, as explained in Section 7.3.2. In each step of the enumeration process, all subsets of r rows of the generator matrices are used. After each step, lower and upper bounds of the minimum weight are obtained. When the upper bound is equal or smaller than the lower bound, the result is achieved, without necessarily enumerating all codewords. The difference between these two methods is that Brouwer's original algorithm only uses generator matrices with no overlap between information coordinate positions, while Zimmerman's variation allows for an overlap between them in order to produce more generator matrices and benefit from all coordinate positions. The reader is referred to [BBF+06, Whi06] for more information on these algorithms.

Both Brouwer's and Brouwer-Zimmermann methods perform well for codes with a low minimum distance. Zimmermann's variation improves greatly when Brouwer's method leaves many coordinate positions unused. For example, this occurs when there are many more quaternary coordinates than binary coordinates. In this case, Brouwer's method produces different

information sets with disjoint binary coordinates positions only until the binary coordinates have run out. On the other hand, Zimmermann's variation reuses some binary coordinates along with new quaternary coordinates in order to use all coordinate positions. Using all coordinate positions helps to increase the lower bound faster, but at the same time, the method is slowed down because of the rank deficits of the generator matrices (the number of reused information coordinates associated to the matrix). There are cases where many of these matrices have large rank deficits. The method automatically discards the matrices that do not contribute to a faster increase of the lower bound, but there may be matrices that do contribute and do not offset the increase in the number of enumerated codewords in each step. In these cases, Brouwer's method may perform better.

The method based on the cosets of the kernel is not affected as much by a high minimum distance and performs well in general. However, the main weakness of the method is that it becomes slower when the number of cosets grows. This is controlled exponentially by the parameter δ, since the codewords of order two are always included in the kernel. This method performs particularly well compared to the Brouwer-Zimmermann method when $\gamma > \kappa$. In this case, there are two types of information coordinate positions: free and torsion positions. Using the Brouwer-Zimmermann method, only the free part (δ positions) contributes to increase the lower bound, while the torsion part ($\gamma - \kappa$ positions) is treated as an overlap, and hence it induces a rank deficit. On the other hand, the method based on the kernel works directly on binary linear codes, which do not suffer from this torsion problem.

Finally, the fifth method considers the code as a quaternary code, by multiplying the first α coordinates by 2 and by duplicating the last β coordinates. The resulting code is a quaternary code of length $\alpha+2\beta$ and type $2^\gamma 4^\delta$ with the same weight distribution. Note that this method does not consider the first κ positions as information coordinate positions, hence it does not benefit from a large value of κ.

The functions defined in the version 5.0 of the package [BFG+22a] include a selector function in order to choose the most suitable method. In general, this is not easy to know without computing the minimum distance

itself. Assuming a random $\mathbb{Z}_2\mathbb{Z}_4$-additive code, a probabilistic argument is used along with the concept of work factors in order to make a reasonable decision. Surely, there are codes for which the selected method may not be optimal. In any case, the user can also decide which method to use by setting the parameter `Method` to `"Distribution"`, `"KernelCosets"`, `"Brouwer"`, `"Zimmermann"`, or `"Quaternary"`.

The functions in this section take into account the following attributes: `MinimumLeeWeight`, `MinimumLeeWeightWord`, `LeeWeightDistribution`, `MinimumLeeWeightLowerBound`, `MinimumLeeWeightUpperBound` associated to a $\mathbb{Z}_2\mathbb{Z}_4$-additive code. The verbose flag `IgnoreWeightAttributes` is set to level 0 by default. In this case, the functions check whether these attributes are already assigned and, if they are, return them directly. If the flag is set to level 1, these attributes are ignored. This can be useful to perform tests and comparisons between the different available methods.

`Z2Z4MinimumLeeWeight(C : parameters)`
`Z2Z4MinimumLeeDistance(C : parameters)`

`Method` MONSTGELT *Default*: "Auto"

Given a $\mathbb{Z}_2\mathbb{Z}_4$-additive code C, return the minimum Lee weight of the codewords belonging to the code C, which is also the minimum Lee distance between any two codewords. Depending on the parameters of the code C, some methods to obtain the minimum Lee weight may be faster than others. For example, sometimes, a brute force calculation of the entire Lee weight distribution can be a faster way for small codes. When the parameter `Method` is set to the default `"Auto"`, then the method is internally chosen. The user can specify which method they want to use, setting the parameter `Method` to `"Distribution"`, `"KernelCosets"`, `"Brouwer"`, `"Zimmermann"`, or `"Quaternary"`.

`MinimumWord(C)`

`Method` MONSTGELT *Default*: "Auto"

Given a $\mathbb{Z}_2\mathbb{Z}_4$-additive code C, return one codeword of the code C having minimum Lee weight. Depending on the parameters of the code C, some

methods to obtain one codeword of minimum Lee weight may be faster than others. For example, sometimes, a brute force calculation of the entire Lee weight distribution can be a faster way for small codes. When the parameter `Method` is set to the default `"Auto"`, then the method is internally chosen. The user can specify which method they want to use, setting the parameter `Method` to `"Distribution"`, `"KernelCosets"`, `"Brouwer"`, `"Zimmermann"`, or `"Quaternary"`.

`MinimumWords(C : parameters)`

`NumWords`	RngIntElt	*Default*: -
`Method`	MonStgElt	*Default*: "Auto"

Given a $\mathbb{Z}_2\mathbb{Z}_4$-additive code C, return the set of all codewords of C having minimum Lee weight. If `NumWords` is set to a non-negative integer, then the algorithm will terminate after at least that total of codewords have been found.

Depending on the parameters of the code C, some methods to collect the codewords of minimum Lee weight may be faster than others. For example, sometimes, a brute force calculation of the entire Lee weight distribution can be a faster way for small codes. When the parameter `Method` is set to the default `"Auto"`, then the method is internally chosen. The user can specify which method they want to use, setting the parameter `Method` to `"Distribution"`, `"KernelCosets"`, `"Brouwer"`, `"Zimmermann"`, or `"Quaternary"`.

Example 92. *We compute the minimum Lee weight of four $\mathbb{Z}_2\mathbb{Z}_4$-additive codes by using different methods. We show that if the code is small, a brute force calculation can be faster than the other methods. The Brouwer-Zimmermann method performs well for codes with a low minimum distance, such as extended perfect codes. When this method uses matrices with very large rank deficits, as with Hadamard codes, Brouwer's original method may perform better. The method based on the kernel is not as affected by a high minimum distance and works well in general, but it becomes slower when the number of cosets grows.*

The verbose flag `IgnoreWeightAttributes` *is set to level 1 in order to compare the computing time between the different available methods. At the end of the example, it is set back to level 0.*

```
> SetVerbose("IgnoreWeightAttributes", 1);

> C := Z2Z4HadamardCode(3, 11);
> #C;
4096
> time Z2Z4MinimumLeeWeight(C : Method := "Distribution");
1024
Time: 1.400
> time Z2Z4MinimumLeeWeight(C : Method := "KernelCosets");
1024
Time: 1.700
> time Z2Z4MinimumLeeWeight(C : Method := "Brouwer");
1024
Time: 39.250
> time Z2Z4MinimumLeeWeight(C : Method := "Zimmermann");
1024
Time: 398.790
> time Z2Z4MinimumLeeWeight(C : Method := "Quaternary");
1024
Time: 216.820

> C := Z2Z4ExtendedPerfectCode(2, 5);
> #C;
67108864
> time Z2Z4MinimumLeeWeight(C : Method := "Distribution");
4
Time: 781.040
> time Z2Z4MinimumLeeWeight(C : Method := "KernelCosets");
4
Time: 0.090
> time Z2Z4MinimumLeeWeight(C : Method := "Brouwer");
4
Time: 0.060
> time Z2Z4MinimumLeeWeight(C : Method := "Zimmermann");
4
Time: 0.050
> time Z2Z4MinimumLeeWeight(C : Method := "Quaternary");
```

```
4
Time: 0.000

> C := Z2Z4ExtendedPerfectCode(2, 6);
> #C;
144115188075855872
> time Z2Z4MinimumLeeWeight(C : Method := "KernelCosets");
4
Time: 656.480
> time Z2Z4MinimumLeeWeight(C : Method := "Brouwer");
4
Time: 0.610
> time Z2Z4MinimumLeeWeight(C : Method := "Zimmermann");
4
Time: 0.500
> time Z2Z4MinimumLeeWeight(C : Method := "Quaternary");
4
Time: 2.440

> C := Z2Z4ReedMullerCode(2, 2, 6);
> #C;
4194304
> time Z2Z4MinimumLeeWeight(C : Method := "Distribution");
16
Time: 74.100
> time Z2Z4MinimumLeeWeight(C : Method := "KernelCosets");
16
Time: 0.090
> time Z2Z4MinimumLeeWeight(C : Method := "Brouwer");
16
Time: 13.900
> time Z2Z4MinimumLeeWeight(C : Method := "Zimmermann");
16
Time: 14.940
> time Z2Z4MinimumLeeWeight(C : Method := "Quaternary");
16
Time: 0.040
```

Then, we check some relations given by the functions related to the minimum Lee weight.

```
> C := Z2Z4HadamardCode(2, 5);
> Z2Z4MinimumLeeWeight(C) eq LeeWeight(MinimumWord(C),
                                                C‘Alpha);
true
> MinimumWord(C) in MinimumWords(C);
true
> LeeWeightDistribution(C);
[ <0, 1>, <16, 62>, <32, 1> ]
> MinimumWords(C) eq MinimumWords(C : NumWords := 62);
true

> SetVerbose("IgnoreWeightAttributes", 0);
```

Chapter 8

Conclusions

In [HKC+94], it was proven that some families of nonlinear binary codes with more codewords than any linear binary code can be seen as the image under the Gray map of linear codes defined over $\mathbb{Z}_4$. Following this result, the research on linear codes over rings gained much attention, especially codes over $\mathbb{Z}_4$, which are also called quaternary codes. In this thesis we deal with $\mathbb{Z}_{p^s}$-additive codes, which are a generalization of quaternary codes, and their corresponding image under a generalized Gray map: $\mathbb{Z}_{p^s}$-linear codes. In particular, we focus on an alternative permutation decoding method that can be applied to nonlinear codes, as long a systematic encoding is known. We give a systematic encoding for any $\mathbb{Z}_{p^s}$-linear code, enabling the permutation decoding method, and then we give constructions of r-PD-sets for $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes. We also give an efficient construction of a parity-check matrix for any $\mathbb{Z}_{p^s}$-linear code and describe the MAGMA package developed during the course of this thesis to handle $\mathbb{Z}_{p^s}$-linear codes.

8.1 Summary

In Chapter 3, we prove that $\mathbb{Z}_{p^s}$-linear codes with p prime and $s \geq 2$, which are not necessarily linear over $\mathbb{Z}_p$, are systematic by giving a systematic encoding for a specific information set. This is a generalization of a previous result for $\mathbb{Z}_4$-linear codes given in [BBFV15]. The generalization is not

straightforward, firstly, because the generalized Gray map Φ_s is not a bijection for $p^s \neq 4$; and secondly, because a family of new functions $\psi_{s,t}$ from $\mathbb{Z}_{p^s}$ to $\mathbb{Z}_{p^t}$ with $t \leq s$ needs to be defined. Moreover, we also show that this systematic encoding for $\mathbb{Z}_{p^s}$-linear codes allows us to use the permutation decoding method described in [BBFV15].

In Chapter 4, we determine the permutation automorphism group of $\mathbb{Z}_{p^s}$-additive GH codes, $\mathrm{PAut}(\mathcal{H}^{t_1,\ldots,t_s})$, and we give a representation of the elements as matrices of the general linear group over $\mathbb{Z}_{p^s}$ of dimension $t_1+\cdots+t_s$. Then, explicit constructions of r-PD-sets of size $r+1$ for $\mathbb{Z}_{p^s}$-linear GH codes of types $(n; t_1, 0, \ldots, 0)$ and $(n; 1, 0, \ldots, 0, t_i, 0, \ldots, 0)$, with $t_1 \geq 2$ and $t_i \geq 1$, respectively, are given. For these cases, the value of r is upper-bounded by $f_p^{t_1,0,\ldots,0}$ or $f_p^{1,0,\ldots,0,t_i,0,\ldots,0}$ depending on the type. In general, for $\mathbb{Z}_{p^s}$-linear GH codes of any type $(n; t_1, \ldots, t_s)$, we also present some recursive constructions of r-PD-sets of size $r+1$, but only up to $r \leq \tilde{f}_p^{t_1,\ldots,t_s} \leq f_p^{t_1,\ldots,t_s}$.

In Chapter 5, new explicit constructions of r-PD-sets of size $r+1$ for $\mathbb{Z}_{p^s}$-linear GH codes, for values of r larger than $\tilde{f}_p^{t_1,\ldots,t_s}$ and closer to the upper bound $f_p^{t_1,\ldots,t_s}$, are given. That is, these new constructions provide a larger r than the recursive constructions given in Chapter 4.

In Chapter 6, two different methods to compute a parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes are introduced. Even though they are very similar methods, and their performance are comparable under some conditions, we have showed that they perform very differently when the parameters of the code change. We have also established experimentally that both are better than the current algorithm included in Magma for any linear code over a finite ring. These methods may also be used to compute a parity-check matrix for codes over chain rings in general.

In Chapter 7, we present the functions developed for a Magma package for $\mathbb{Z}_{p^s}$-additive codes, which implement the results found in the previous chapters. Namely, there are functions that return the systematic encoding found in Chapter 3 and its corresponding information space, others that return the r-PD-sets given by the constructions presented in Chapters 4 and 5, and others that are able to construct efficiently the parity-check matrix of a $\mathbb{Z}_{p^s}$-additive code, as explained in Chapter 6. This package also generalizes

some of the functions for codes over $\mathbb{Z}_4$, which are already included in the standard MAGMA distribution [BCFS19]. It has been developed mainly by the author of this thesis and his advisor. The first version of this new package and a manual describing all functions are available in a GitHub repository (`https://github.com/merce-github/ZpAdditiveCodes`) and in the CCSG website (`https://ccsg.uab.cat`). In Chapter 7 we also describe the computation of the minimum homogeneous weight for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, using an adapted version of the Brouwer-Zimmermann algorithm, as part of a different MAGMA package for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes.

8.2 Further research

In the following lines, we give some possible directions of further research which could extend the results given in this thesis.

- In our results, we have considered the generalized Gray map given by Carlet [Car98]. By considering other Gray maps, like the ones defined in [Kro07, SWK19], we could obtain different (and probably, nonequivalent) $\mathbb{Z}_{p^s}$-linear codes. Similarly, a systematic encoding for these other families of codes could also be established as further research.

- In [BBFV15], the given systematic encoding is also presented for $\mathbb{Z}_2\mathbb{Z}_4$-linear codes, which are a generalization of $\mathbb{Z}_4$-linear codes. Specifically, $\mathbb{Z}_2\mathbb{Z}_4$-linear codes are the Gray map images of $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, which are subgroups of $\mathbb{Z}_2^{\alpha_1} \times \mathbb{Z}_4^{\alpha_2}$, and have been studied extensively in [BFP$^+$10, BFP$^+$22b]. The family of $\mathbb{Z}_p\mathbb{Z}_{p^2} \dots \mathbb{Z}_{p^s}$-additive codes, with p prime, can be defined in a similar way to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. The results presented in Chapter 3 can also be easily applied to $\mathbb{Z}_p\mathbb{Z}_{p^2} \dots \mathbb{Z}_{p^s}$-linear codes, that is, codes over $\mathbb{Z}_p$ (not necessarily linear) which are the Gray map images of $\mathbb{Z}_p\mathbb{Z}_{p^2} \dots \mathbb{Z}_{p^s}$-additive codes.

- For some infinite families of $\mathbb{Z}_{p^s}$-linear GH codes, of type $(t_1, \dots, t_s)$, the constructions given in Chapters 4 and 5 allow us to construct r-PD-sets with r up to the upper bound, that is, for all $r \leq f_p^{t_1,\dots,t_s}$.

A natural direction of further research on this topic is to achieve the theoretical upper bound for all cases, or to prove that there are cases where it is impossible, resulting in a lower upper bound which depends on the type of the code.

- In Chapter 4, for any $\mathbb{Z}_{p^s}$-additive GH code $\mathcal{H}^{t_1,\dots,t_s}$, we have studied $\mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})$, which is a subgroup of the monomial automorphism group, denoted by $\mathrm{MAut}(\mathcal{H}^{t_1,\dots,t_s})$. Then, the given r-PD-sets are elements of $\Phi(\mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s}))$. Note that $\Phi(\mathrm{PAut}(\mathcal{H}^{t_1,\dots,t_s})) \subseteq \mathrm{PAut}(H^{t_1,\dots,t_s})$. The study of both of these groups, $\mathrm{MAut}(\mathcal{H}^{t_1,\dots,t_s})$ and $\mathrm{PAut}(H^{t_1,\dots,t_s})$, remains as an open problem for $p \geq 3$ or $s \geq 3$. For $\mathbb{Z}_4$-linear Hadamard codes, these groups are studied in [KV15]. The description of the group $\mathrm{PAut}(H^{t_1,\dots,t_s})$ may allow us to find r-PD-sets of size $r+1$ for $r > f_p^{t_1,\dots,t_s}$ or r-PD-sets of larger size, up to the error-correcting capability, improving the results obtained in this thesis.

- The results in Chapters 4 and 5 regarding the permutation automorphism group and the construction of r-PD-sets for $\mathbb{Z}_{p^s}$-linear GH codes could be generalized to $\mathbb{Z}_p\mathbb{Z}_{p^2}\cdots\mathbb{Z}_{p^s}$*-linear GH codes*. Such codes are GH codes that can be obtained from the generalized Gray map image of subgroups over mixed alphabets $\mathbb{Z}_p^{\alpha_1} \times \mathbb{Z}_{p^2}^{\alpha_2} \times \cdots \times \mathbb{Z}_{p^s}^{\alpha_s}$. The permutation decoding method given in [BBFV15] is also apt for these codes, since they are systematic. More generally, $\mathbb{Z}_p\mathbb{Z}_{p^2}\cdots\mathbb{Z}_{p^s}$-linear codes have been studied, for example in [AS15, LSW$^+$24, SWK19]. The results given in Chapter 3 can be extended to $\mathbb{Z}_p\mathbb{Z}_{p^2}\cdots\mathbb{Z}_{p^s}$-linear codes, in order to obtain a systematic encoding for these codes, enabling the permutation decoding method. This gives a motivation to construct r-PD-sets for $\mathbb{Z}_p\mathbb{Z}_{p^2}\cdots\mathbb{Z}_{p^s}$-linear GH codes, which have been recently studied in [BFVV24, BFV22a, BFV22b, BFV23], showing that they are not necessarily equivalent to the $\mathbb{Z}_{p^s}$-linear GH codes considered in this thesis.

- Similarly, the results in Chapters 4 and 5 could be generalized to other families of $\mathbb{Z}_{p^s}$-linear codes [FVV21, SHQ$^+$21, SSA18]. For example,

the families of simplex and MacDonald codes, which have a high error-correcting capability, may also present a large enough permutation automorphism group in order to find r-PD-sets.

- A natural generalization of the results given in Chapter 6 would be to adapt the given algorithms to compute a parity-check matrix for codes over mixed alphabets like $\mathbb{Z}_p\mathbb{Z}_{p^2}$-additive codes, $\mathbb{Z}_p\mathbb{Z}_{p^s}$-additive codes or even the more generic $\mathbb{Z}_p\mathbb{Z}_{p^2}\ldots\mathbb{Z}_{p^s}$-additive codes.

- The generator matrix of a $\mathbb{Z}_{p^s}$-additive code in standard form has a very similar structure to the partial generator matrix of convolutional codes (also called expanded partial generator matrix). Due to this similarity, the methods presented in Chapter 6 can be adapted to compute a parity-check matrix for these codes as long as it exists.

- The MAGMA package described in Chapter 7 is still on its first version. The functionality that this package provides for $\mathbb{Z}_{p^s}$-linear and $\mathbb{Z}_{p^s}$-additive codes is still not on par with that of the packages created for other structures. For example, the computation of the homogeneous minimum weight through the Brouwer-Zimmerman algorithm has not yet been implemented.

- Most of the functions included in the MAGMA package described in Chapter 7 can be generalized without much trouble to mixed-alphabet codes such as $\mathbb{Z}_p\mathbb{Z}_{p^2}\ldots\mathbb{Z}_{p^s}$-additive and $\mathbb{Z}_p\mathbb{Z}_{p^2}\ldots\mathbb{Z}_{p^s}$-linear codes.

Bibliography

[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(6), pp. 2271–2278, 2013.

[AS15] I. Aydogdu and I. Siap, "On $\mathbb{Z}_{p^r}\mathbb{Z}_{p^s}$-additive codes," *Linear and Multilinear Algebra*, vol. 63(10), pp. 2089–2102, 2015

[BV18] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for binary linear and $\mathbb{Z}_4$-linear Hadamard codes," *Des. Codes Cryptogr.*, vol. 86(3), pp. 569–586, 2018.

[BV19] R. D. Barrolleta and M. Villanueva, "Partial permutation decoding for several families of linear and $\mathbb{Z}_4$-linear Codes," *IEEE Trans. Inf. Theory*, vol. 65(1), pp. 131–141, 2019.

[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(2), pp. 269–277, 2015.

[BS13] J. J. Bernal, J. J. Simón, "Partial permutation decoding for Abelian codes," *IEEE Trans. Inf. Theory*, vol. 59(8), pp. 5152–5170, 2013.

[BS23] J. J. Bernal, J. J. Simón, "New advances in permutation decoding on first-order Reed-Muller codes," *Finite Fields Their Appl.*, vol. 88, 102182, 2023.

[BBF+06] A. Betten, M. Braun, H. Fripertinger, A. Kerber, A. Kohnert, A. Wassermann, "Error-Correcting Linear Codes: Classification by Isometry and Applications," *Algorithms and Computation in Mathematics*, vol. 18, Springer, 2006.

[BFV22a] D. K. Bhunia, C. Fernández-Córdoba, and M. Villanueva, "On the linearity and classification of $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes," *Des. Codes and Cryptogr.*, vol. 90(4), pp. 1037–1058, 2022.

[BFV22b] D. K. Bhunia, C. Fernández-Córdoba, and M. Villanueva, "On the constructions of $\mathbb{Z}_p\mathbb{Z}_{p^2}$-linear generalized Hadamard codes," *Finite Fields Their Appl.*, vol. 83, 102093, 2022.

[BFV23] D. K. Bhunia, C. Fernández-Córdoba, and M. Villanueva, "Linearity and classification of $\mathbb{Z}_p\mathbb{Z}_{p^2}$-linear generalized Hadamard codes," *Finite Fields Their Appl.*, vol. 86, 102140, 2023.

[BFVV24] D. K. Bhunia, C. Fernández-Córdoba, C. Vela, and M. Villanueva, "On the equivalence of $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes," *Des. Codes and Cryptogr.*, 2024. DOI 10.1007/s10623-023-01325-2

[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(2), pp. 167–179, 2010.

[BFG+17] J. Borges, C. Fernández-Córdoba, B. Gastón, J. Pujol, J. Rifà, and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-additive codes," version 4.0, Universitat Autònoma de Barcelona, 2017. `https://ccsg.uab.cat`.

[BFG+22a] J. Borges, C. Fernández-Córdoba, B. Gastón, J. Pujol, J. Rifà, and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-additive codes," version 5.0, Universitat Autònoma de Barcelona, 2022. `https://ccsg.uab.cat`.

[BFP+22b] J. Borges, C. Fernández-Córdoba, J. Pujol, J. Rifà, and M. Villanueva, *$\mathbb{Z}_2\mathbb{Z}_4$-linear codes*, Springer Cham, Switzerland, 2022.

[BCFS19] W. Bosma, J.J. Cannon, C. Fieker, and A. Steel (eds.), Handbook of Magma functions, Edition 2.24-10, 2019. `http://magma.maths.usyd.edu.au/magma/`.

[CS95] A. R. Calderbank and J. A. Sloane, "Modular and p-adic cyclic codes," *Des. Codes and Cryptogr.*, vol. 6, pp. 21–35, 1995.

[Car98] C. Carlet, "$\mathbb{Z}_{2^k}$-linear codes," *IEEE Trans. Inf. Theory*, vol. 44(4), pp. 1543–1547, 1998.

[CD21] B. J. Chathely and R. P. Deore, "Construction of binary Hadamard codes and their s-PD sets," *Cryptography and Communications*, vol. 13, pp. 425–438, 2021.

[CH97] I. Constantinescu and W. Heise, "A metric for codes over residue class rings," *Problemy Peredachi Informatsii*, vol 33(3), pp. 22–28, 1997.

[DF11] S. T. Dougherty and C. Fernández-Córdoba, "Codes over $\mathbb{Z}_{2^k}$, Gray map and self-dual codes," *Advances in Mathematics of Communications*, vol. 5(4), pp. 571–588, 2011.

[DRV16] S. T. Dougherty, J. Rifà, and M. Villanueva, "Ranks and kernels of codes from generalized Hadamard matrices," *IEEE Trans. Inf. Theory*, vol. 62(2), pp. 687–694, 2016.

[FTV23] C. Fernández-Córdoba, A. Torres-Martín, and M. Villanueva, "Linear codes over the integer residue ring $\mathbb{Z}_{p^s}$. A MAGMA package", version 1.0, Universitat Autònoma de Barcelona, 2023. `https://ccsg.uab.cat`

[FTVV24a] C. Fernández-Córdoba, A. Torres-Martín, C. Vela, and M. Villanueva, "Parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes: efficient computation," submitted to *2024 IEEE International Symposium on Information Theory (ISIT)*, 7-12 July, 2024.

[FTVV24b] C. Fernández-Córdoba, A. Torres-Martín, C. Vela, and M. Villanueva, "Computing efficiently a parity-check matrix for $\mathbb{Z}_{p^s}$-additive codes," submitted to *IEEE Trans. Inf. Theory*, 2024.

[FVV19] C. Fernández-Córdoba, C. Vela, and M. Villanueva, "On $\mathbb{Z}_{2^s}$-linear Hadamard codes: kernel and partial classification," *Des. Codes and Cryptogr.*, vol. 87(2-3), pp. 417–435, 2019.

[FVV20b] C. Fernández-Córdoba, C. Vela, and M. Villanueva, "On $\mathbb{Z}_8$-linear Hadamard codes: rank and classification," *IEEE Trans. Inf. Theory*, vol. 66(2), pp. 970–982, 2020.

[FVV21] C. Fernández-Córdoba, C. Vela, and M. Villanueva, "Nonlinearity and kernel of $\mathbb{Z}_{2^s}$-linear simplex and MacDonald codes," *IEEE Trans. Inf. Theory*, vol. 68(11), pp.7174–7183, 2022.

[FKM12] W. Fish, J. D. Key, and E. Mwambeme, "Partial permutation decoding for simplex codes," *Adv. Math. Commun.*, vol. 6(4), pp. 505–516, 2012.

[Gra06] M. Grassl, "Searching for linear codes with large minimum distance", Discovering Mathematics with MAGMA, pp. 287–313, 2006.

[GS99] M. Greferath and S. E. Schmidt, "Gray isometries for finite chain rings and a nonlinear ternary $(36, 3^{12}, 15)$ code," *IEEE Trans. Inf. Theory*, vol. 45(7), pp. 2522–2524, 1999.

[GBL05] M. K. Gupta, M. C. Bhandari, and A. K. Lal, "On some linear codes over $\mathbb{Z}_{2^s}$," *Des. Codes and Cryptogr.*, vol. 36(3), pp. 227–244, 2005.

[Ham50] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, 1950.

[HKC+94] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. S. A. Sloane, and P. Solé, "The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals and related codes," *IEEE Trans. Inf. Theory*, vol. 40(2), pp. 301–319, 1994.

[Han06] J. Han, "General linear group over a ring of integers of modulo k," *Kyungpook Mathematical Journal*, vol. 46, pp. 255–260, 2006.

[HP03] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*. Cambridge: Cambridge University Press, 2003.

[Jun79] D. Jungnickel, "On difference matrices, resolvable transversal designs and generalized Hadamard matrices," *Math. Zeitschrift*, 167(1), pp. 49–60, 1979.

[Ker72] A. M. Kerdock, "A class of low-rate nonlinear binary codes," *Information and Control*, vol. 20, pp. 182–187.

[Kro01] D. S. Krotov, "$\mathbb{Z}_4$-linear Hadamard and extended perfect codes," *Electron. Notes Discrete Math.*, vol. 6, pp. 107–112, 2001.

[Kro07] D. S. Krotov, "On $\mathbb{Z}_{2^k}$-dual binary codes," *IEEE Trans. Inf. Theory*, vol. 53(4), pp. 1532–1537, 2007.

[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(2), pp. 887–894, 2015.

[LSW+24] X. Li, M. Shi, S. Wang, H. Lu, and Y. Zheng, "Rank and pairs of rank and dimension of kernel of $\mathbb{Z}_p\mathbb{Z}_{p^2}$-linear codes," *IEEE Trans. Inf. Theory*, vol. 70(5), pp. 3202–3212, 2024.

[Mac64] F. J. MacWilliams, "Permutation decoding of systematic 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*. 16, Elsevier, 1977.

[Nec91] A. A. Nechaev, "The Kerdock code in a cyclic form," *Discrete Math. Appl.*, vol. 1, pp. 365–384, 1991.

[NS00] G. H. Norton and A. Sălăgean, "On the structure of linear and cyclic codes over a finite chain ring," *Applicable Algebra in Engineering, Communications and Computing*, vol. 10, pp. 489–506, 2000.

[PPV14] J. Pernas, J. Pujol, and M. Villanueva, "Characterization of the automorphism group of quaternary linear Hadamard codes," *Des. Codes Cryptogr.*, vol. 70(1-2), pp. 105–115, 2014.

[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(1), pp. 316–319, 2006.

[Pra62] E. Prange, "The use of information sets in decoding cyclic codes," *IRE Trans. Inform. Theory*, vol. 8(5), pp. 5–9, 1962.

[Pre68] F. P. Preparata, "A class of optimum nonlinear double-error-correcting codes," *Information and Control*, vol. 13, pp. 378–400, 1968.

[RTV24] J. Rifà, A. Torres-Martín, and M. Villanueva, "Improving explicit constructions of PD-sets for $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes," to appear in *IEEE Trans. Inf. Theory*, 2024. DOI: 10.1109/TIT.2024.3448230.

[Sha48] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, 379–423, 1948.

[SHQ+21] M. Shi, T. Honold, P. Solé, Y. Qiu, R. Wu, and Z. Sepasdar, "The geometry of two-weight codes over $\mathbb{Z}_{p^m}$," *IEEE Trans. Inf. Theory*, vol. 67(12), pp. 7769–7781, 2021.

[SSA18] M. Shi, Z. Sepasdar, A. Alahmadi, and P. Solé, "On two-weight $\mathbb{Z}_{2^k}$-codes," *Des. Codes and Cryptogr.*, vol. 86(6), pp. 1201–1209, 2018.

[SWK19] M. Shi, R. Wu, and D. S. Krotov, "On $\mathbb{Z}_p\mathbb{Z}_{p^k}$-additive codes and their duality," *IEEE Trans. Inf. Theory*, vol. 65(6), pp. 3841–3847, 2019.

[TV03] H. Tapia-Recillas and G. Vega, "On $\mathbb{Z}_{2^k}$-linear and quaternary codes," *SIAM J. Discrete Math.*, vol. 17(1), pp. 103–113, 2003.

[TV20] A. Torres-Martín and M. Villanueva, "Systematic encoding for $\mathbb{Z}_{2^s}$-linear codes," in Proc. of *2020 Algebraic and Combinatorial Coding Theory (ACCT)*, pp. 140–144, 2020.

[TV22a] A. Torres-Martín and M. Villanueva, "Systematic encoding and permutation decoding for $\mathbb{Z}_{p^s}$-linear codes," *IEEE Trans. Inf. Theory*, vol. 68(7), pp. 4435–4443, 2022.

[TV22b] A. Torres-Martín and M. Villanueva, "Partial permutation decoding for $\mathbb{Z}_8$-linear Hadamard codes," in Proc. of *2022 IEEE Information Theory Workshop (ITW)*, 1-2 November 2022 - Virtual, 2022.

[TV24a] A. Torres-Martín and M. Villanueva, "Partial permutation decoding and PD-sets for $\mathbb{Z}_{p^s}$-linear generalized Hadamard codes," *Finite Fields Their Appl.*, vol. 93, 102316, 2024.

[Var97] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Trans. Inf. Theory*, vol. 43(6), pp. 1757–1766, 1997.

[VZP15] M. Villanueva, F. Zeng, and J. Pujol, "Efficient representation of binary nonlinear codes: constructions and minimum distance computation," *Des. Codes Cryptogr.*, vol. 76(1), pp. 3–21, 2015.

[Wan03] Z.-X. Wan, *Lectures on Finite Fields and Galois Rings*, World Scientific, Singapore, 2003.

[Whi06] G. White, *Enumeration-Based Algorithms in Linear Coding Theory*, PhD Thesis, University of Sydney, 2006.

[Zim96] K.-H. Zimmermann, "Integral Hecke Modules, Integral Generalized Reed-Muller Codes, and Linear codes," *Technical Report 3-96*, Technische Universitat Hamburg-Harburg, 1996.

Adrián Torres Martín
Cerdanyola del Vallès, September 2024